IBM Cloud Private 3.2.0

Índice

Documentação do IBM® Cloud Private v3.2.0

Bem-vindo à documentação do IBM Cloud Private, na qual é possível localizar informações sobre como instalar, manter e usar o IBM Cloud Private.

Informações Iniciais

Visão Geral

Requisitos do Sistema

Notas sobre a Liberação

Recursos de acessibilidade para o IBM Cloud Private

Plataforma Cloud Foundry do IBM Cloud Private

IBM Multicloud Manager

Tarefas Comuns

Instalando o IBM Cloud Private

Gerenciando cargas de trabalho

Gerenciando gráficos e aplicativos

Gerenciando imagens

Gerenciando sua plataforma

Resolução de Problemas e Suporte

Resolução de problemas

Canal do IBM Cloud Technology no Slack

Comunidade técnica

Visualize as perguntas no Stack Overflow com a tag ibm-cloud-private

Recursos de Acessibilidade para

IBM Cloud Private

Os recursos de acessibilidade ajudam usuários com algum tipo de deficiência, tal como mobilidade restrita ou visão limitada, a usarem conteúdo de tecnologia de informações com êxito.

Visão Geral

IBM Cloud Private inclui os principais recursos de acessibilidade a seguir:

- Operações somente por teclado
- Operações do leitor de tela
- Interface da linha de comandos (CLI) para gerenciar o cluster do IBM Cloud Private

O IBM Cloud Private usa o W3C Standard mais recente, WAI-ARIA 1.0, para assegurar a conformidade com a Seção 508 de Padrões para Eletroeletrônicos e Tecnologia da Informação e Web Content Accessibility Guidelines (WCAG) 2.0. Para aproveitar os recursos de acessibilidade, use a liberação mais recente

de seu leitor de tela e o navegador da web mais recente que é suportado pelo IBM Cloud Private.

A documentação online do produto IBM Cloud Private no IBM Knowledge Center é ativada para acessibilidade. Os recursos de acessibilidade do IBM Knowledge Center estão descritos na seção Acessibilidade das notas sobre a liberação do IBM Knowledge Center. Para obter informações de acessibilidade geral, consulte Acessibilidade na IBM.

Hyperlinks

Todos os links externos, que são links para o conteúdo que é hospedado fora do IBM Knowledge Center, são abertos em uma nova janela. Esses links externos também são sinalizados com um ícone de link externo ().

Navegação pelo teclado

IBM Cloud Private usa chaves de navegação padrão.

O IBM Cloud Private usa os atalhos de teclado a seguir.

Ação	Atalho para o Internet Explorer	Atalho para o Firefox
Mover para		

o quadro Visualização de Conteúdo|Alt+C e, em seguida, pressione Enter e Shift+F6|Shift+Alt+C e Shift+F6|

Informações de interface

Use a versão mais recente de um leitor de tela com o IBM Cloud Private.

As interfaces com o usuário do IBM Cloud Private não possuem conteúdo que é atualizado de 2 a 55 vezes por segundo.

A interface com o usuário da web do IBM Cloud Private depende das folhas de estilo em cascata para renderizar o conteúdo corretamente e para fornecer uma experiência utilizável. O aplicativo fornece uma maneira equivalente para os usuários com baixa visão para usar as configurações de exibição do sistema, incluindo o modo de alto contraste. É possível controlar o tamanho de fonte usando as configurações do dispositivo ou do navegador da web.

É possível acessar o IBM Cloud Private nos navegadores suportados a seguir:

Tabela 2. Navegadores suportados

Plataforma	Navegadores Suportados
Windows™	• Edge - versão mais recente • Mozilla Firefox - versão mais recente para Windows • Google Chrome-versão mais recente para o Windows
Linux®	• Mozilla Firefox-versão mais recente para o Linux • Google Chrome-versão mais recente para o Linux
MacOS	• Mozilla Firefox - versão mais recente para Mac • Google Chrome - versão mais recente para Mac • Safari - versão mais recente

Para acessar a console de gerenciamento, abra um navegador da web e navegue para a URL a seguir:

`https://<Cluster Master Host>:<Cluster Master API Port>`

Em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal. O nome do usuário e a senha são definidos no arquivo config.yaml.

O console de gerenciamento não depende de folhas de estilo em cascata para renderizar o conteúdo corretamente e para fornecer uma experiência utilizável. No entanto, a documentação do produto, que está disponível no IBM Knowledge Center, depende de folhas de estilo em cascata. O IBM Cloud Private fornece uma maneira equivalente para usuários com pouca visão usarem configurações de exibição do sistema, incluindo o modo de alto contraste. É possível controlar o tamanho da fonte usando as configurações do dispositivo ou navegador. Observe que a documentação do produto contém caminhos de arquivos, variáveis de ambiente, comandos e outros conteúdos que podem ser pronunciados errados por leitores de tela padrão. Para descrições mais precisas, defina suas configurações do leitor de tela para ler todas as pontuações.

Software do fornecedor

IBM Cloud Private inclui certos produtos de software de fornecedor que não estão cobertos no contrato de licença da IBM. A IBM não faz nenhuma representação sobre os recursos de acessibilidade desses produtos. Entre em contato com o fornecedor para obter as informações de acessibilidade sobre seus produtos.

Informações Relacionadas à Acessibilidade

Além do IBM help desk padrão e de websites de suporte, a IBM possui um serviço telefônico TTY para uso por clientes surdos ou com deficiência auditiva para acessar serviços de vendas e suporte:

Serviço TTY
800-IBM-3383 (800-426-3383)
(na América do Norte)

Para obter mais informações sobre o compromisso que a IBM tem com a acessibilidade, consulte IBM Accessibility.

Notas sobre a liberação

- O que Há de Novo
- Problemas e limitações conhecidos
- Estabilizada, descontinuada e removida

O que há de novo na versão 3.2.0

Obtenha uma visão geral rápida do que foi incluído, mudado, melhorado ou descontinuado nessa liberação.

O IBM® Cloud Private Versão 3.2.0 apresenta os novos recursos e aprimoramentos a seguir:

- Instalação, configuração e upgrade
- Segurança e conformidade
- Rede
- Armazenamento
- Monitoramento e criação de log
- Melhorias de desempenho
- IBM Cloud Private console de gerenciamento
- IBM Cloud Private CLI (cloudctl)
- IBM Cloud Private Cloud Foundry e o Cloud Foundry Enterprise Environment
- IBM Multicloud Manager
- Visualização de tecnologia
- APIs
- Mudanças na versão do pacote
- Resolução de Problemas e Suporte

Instalação, configuração e upgrade

Ativando o IBM Multicloud Manager

É possível configurar o IBM Multicloud Manager durante a instalação do IBM® Cloud Private customizando seu arquivo `config.yaml`. Para obter detalhes, consulte IBM Multicloud Manager.

Upgrade de Multi-release

É possível fazer upgrade diretamente para o IBM Cloud Private3.2.0 a partir das versões 3.1.0, 3.1.1 e 3.1.2. Para obter informações adicionais, consulte Fazendo upgrade do .

Implementando o Red Hat OpenShift versão 3.11 no modo de alta

disponibilidade

Agora é possível configurar a alta disponibilidade para um cluster do IBM Cloud Private with OpenShift versão 3.11 implementando o IBM Cloud Private em nós dedicados do OpenShift. No arquivo `config.yaml` durante a instalação, é possível especificar os nós

dedicados do OpenShift para os nós do cluster principal, de proxy e de gerenciamento que implementam o componente do IBM Cloud Private como cargas de trabalho do OpenShift.

Pacote do Key Management Service Hardware Security Module

O gráfico e as imagens do Key Management Service (KMS) Hardware Security Module (HSM) não estão mais incluídos no instalador do IBM Cloud Private. Para usar o KMS, é possível fazer download e instalar o pacote `key-management-hsm-amd64.tar.gz` do *3.2.0 Key Management HSM* a partir do IBM Passport Advantage. Para obter mais informações, consulte Configurando o Key Management Service.

Instalação modularizada

Para reduzir a área de cobertura do IBM Cloud Private em sua plataforma, durante a instalação e posteriormente, é possível desativar serviços de gerenciamento. Após a instalação, se você precisar de quaisquer serviços desativados, será possível ativá-los. Para obter mais informações sobre os componentes que estão disponíveis e os serviços de gerenciamento que estão incluídos com o componente, consulte Ativando e desativando componentes do IBM Cloud Private.

Segurança e conformidade

- Os certificados SSL (Secure Sockets Layer) que são necessários para sua conexão LDAP sobre SSL (LDAPS) agora são configurados automaticamente quando você se conecta a seu diretório. Para obter mais informações, consulte Configurando a conexão LDAP.
- É possível criar uma política de sincronização para designar um horário para as seguintes varreduras: pod do Vulnerability Advisor (VA), imagem do VA, processo do Mutation Advisor. Para obter mais informações, consulte a seção *Política de cronometragem* na Página do Vulnerability Advisor.
- Novas APIs de gerenciamento de usuários foram incluídas. Para obter uma lista completa das APIs de gerenciamento de usuários, consulte APIs de gerenciamento de usuários.
- APIs de verificação de funcionamento e de versão do serviço do IAM são incluídas. Para obter mais informações, consulte APIs de verificação de funcionamento e de versão do serviço.
- O MongoDB é usado no lugar de MariaDB for OpenID Connect (OIDC).
- Vários tópicos da guia de adoção do IAM são atualizados. Para obter mais informações, consulte Guia de segurança.
- É possível mudar as configurações de cache e de procura Logjam e LDAP. Para obter mais informações, consulte IAM para usuários da plataforma IBM Cloud Private e Resolução de problemas de procura de usuários e de grupos de usuários.
- O guia de adoção do Key Management Service é incluído. Para obter mais informações, consulte Guia de adoção do Key Management Service (KMS).
- Como um usuário de gerenciamento de chave, é possível criar um segredo com uma anotação específica que provisiona uma instância do Key Management Service. Para obter mais informações, consulte Provisionando instâncias do KMS.
- O IBM Cloud Private versão 3.2.0 agora suporta o nCipher nShield Connect HSM 12.40.2. Para obter informações adicionais, consulte Configurando o nCipher nShield Connect HSM 12.40.2.

auditoria

O guia de adoção de criação de log de auditoria é incluído. Para obter mais informações, consulte Guia de adoção de criação de log de auditoria.

- Use estatísticas sobre dados de auditoria gerados para ajudar a ajustar políticas de auditoria, alocar espaço em disco e preparar o ELK ou SIEM para manipular registros de auditoria. Para obter mais informações, consulte Estatísticas de dados de criação de log de auditoria.
- É possível integrar seus logs de auditoria do IBM Cloud Private com o Splunk. Para obter mais informações, consulte Integrando o IBM Cloud Private ao Splunk.
- Aprenda como incluir painéis customizados no Kibana para que seja possível analisar os logs de auditoria. Para obter mais informações, consulte Painel do Kibana de criação de log de auditoria.

Certificados

Substituindo, atualizando e restaurando certificados criados pelo instalador

É possível substituir o certificado de autoridade de certificação raiz e atualizar e restaurar os certificados que são criados pelo instalador e usados por serviços de plataforma no ambiente IBM Cloud Private. Para obter informações adicionais, consulte Substituindo certificados, Atualizando certificados e Restaurando certificados.

O certificado de CA raiz agora está armazenado no segredo do `ibmcloud-cluster-ca-cert` no namespace `kube-public`. O certificado pode ser importado em seus armazenamentos confiáveis do cliente para acessar as APIs do IBM Cloud Private Platform. Para obter mais informações, consulte Certificados no IBM Cloud Private.

Mudanças no Gerenciador de certificados

- É possível visualizar os Certificados, Emissores e ClusterIssuers em seu cluster, incluindo informações sobre a idade e expiração dos certificados. Para obter mais informações, consulte Visualizando recursos cert-manager do IBM Cloud Private.
- É possível atualizar manualmente os certificados cert-manager e reiniciar os pods automaticamente usando esses certificados. Para obter mais informações, consulte Atualizando certificados do cert-manager do IBM Cloud Private.
- É possível configurar durações e janelas de renovação de certificados. Para obter mais informações, consulte Customizando certificados do cert-manager do IBM Cloud Private.
- É possível configurar endereços IP, além de servidores DNS em certificados do cert-manager. Para obter mais informações, consulte Criando certificados do cert-manager do IBM Cloud Private.
- Agora é possível configurar o emissor ACME para criar certificados confiáveis a partir de letsencrypt.org. Para obter informações adicionais, consulte Incluindo certificados usando o emissor ACME.

Rede

O VMware NSX-T é atualizado para a versão 2.4.

Configure um refletor de rota Calico se seu cluster estiver em um ambiente com diferentes segmentos da Camada 3 e você não quiser conectividade da Camada 3 nesses segmentos. Para obter mais informações sobre como configurar um refletor de rota durante a instalação do IBM Cloud Private, consulte Implementando o IBM Cloud Private em segmentos isolados da Camada 3. Para obter mais informações sobre como configurar um refletor de rota após a instalação do IBM Cloud Private, consulte Configurando o refletor de rota do Calico após a instalação do IBM Cloud Private.

Memória

- Agora é possível configurar uma classe de armazenamento para o vSphere durante a instalação do IBM Cloud Private. Para obter informações adicionais, consulte Usar o arquivo config.yaml para configuração do vSphere Cloud Provider.
- Agora o GlusterFS e o Minio podem ser configurados em plataformas Linux® x86_64, Linux® on Power® (ppc64le) e IBM® Z.
- O servidor externo do CephFS pode ser integrado ao IBM Cloud Private.

Monitorando e criando logs

IBM Cloud Private de monitoramento

Agora, o gráfico do Helm `ibm-icpmonitoring` fornece controles de acesso baseados em função (RBAC) para acessar os painéis de monitoramento no Grafana. Para obter mais informações, consulte Acesso baseado em função para painéis de monitoramento.

Criação de log do IBM Cloud Private

Procedimentos disponíveis para gerenciar a configuração de criação de log que cobrem o ajuste de escala horizontal e vertical e o gerenciamento de segurança.

- É possível ativar recursos de segurança para serviços de criação de log após a instalação inicial do IBM Cloud Private. Para obter informações adicionais, consulte Ativando a segurança para serviços de criação de log.
- O serviço de criação de log descontinuou a execução sem a segurança ativada. Para obter informações adicionais, consulte Ativando a segurança para serviços de criação de log.

- É possível gerenciar recursos que são alocados para serviços de criação de log. Para obter informações adicionais, consulte Gerenciando a alocação de recurso para serviços de criação de log.
- Agora é possível ativar o monitoramento de funcionamento do Elastic Stack para criação de log. Para obter informações adicionais, consulte Ativando o monitoramento do Elastic.
- É possível customizar suas políticas de retenção de dados para ajudar a manter os tamanhos dos dados sob controle. Para obter mais informações, consulte Modificando a política de retenção de dados para serviços de criação de log.
- É possível instalar mais instâncias do gráfico de Criação de Log com a segurança ativada para evitar acesso não autenticado e para restringir o acesso com base no acesso ao namespace. Instâncias extras ativam a operação segura para muitos cenários, incluindo ocupação variada. Para obter mais informações, consulte Instalando instâncias de criação de log adicionais.
- Se você precisar de mais capacidade, agora é possível escalar horizontalmente o serviço de criação de log para usar nós recém-incluídos após a instalação inicial do IBM Cloud Private. Para obter mais informações, consulte Escalando serviços de criação de log após a instalação do IBM Cloud Private.
- É possível aplicar filtragem adicional ao processo de coleta de log. Para obter mais informações, consulte Atualizando filtros de coleção de serviço de criação de log.

Monitoramento no OpenShift

O OpenShift fornece um componente de monitoramento opcional baseado em Prometheus, mas não fornece os mesmos recursos que o serviço de monitoramento do IBM Cloud Private. Ao instalar o IBM Cloud Private no OpenShift, o serviço de monitoramento do IBM Cloud Private é instalado por padrão. É possível desativar o serviço de monitoramento no OpenShift. Para obter mais informações, consulte a seção *Gerenciando painéis Grafana* na página de monitoramento do IBM Cloud Private.

Se o IBM Multicloud Manager estiver instalado, o monitoramento do IBM Cloud Private deverá ser ativado para federar métricas de seus outros clusters.

Efetuando LogonOpenShift

O OpenShift fornece um serviço de criação de log opcional baseado no Elasticsearch que coleta logs de componentes de sistema e de aplicativo automaticamente. É possível optar por instalar o serviço de criação de log do IBM Cloud Private. Para obter mais informações, consulte IBM Cloud Private log .

Melhorias de desempenho

A sincronização do repositório do Helm automático busca apenas os gráficos do Helm atualizados: quando os repositórios do Helm são sincronizados automaticamente, apenas os gráficos que possuem atualizações são buscados. Isso economiza tempo, já que todos os gráficos não são buscados, não importa se eles possuem atualizações pendentes.

Agora é possível usar o Vulnerability Advisor para varrer registros de imagem externos. Para obter informações adicionais, consulte Varrendo registros externos com o Vulnerability Advisor.

IBM Cloud Private console de gerenciamento

O local da página *Introdução* mudou e contém novas informações. Por exemplo, é possível acessar todas as ferramentas de CLI suportadas e é possível ver o conteúdo *Configurar cliente*, que também está disponível no menu do usuário. Além disso, é possível acessar o terminal da web a partir do cabeçalho.

O *Search* agora está disponível para o IBM Cloud Private e o IBM Multicloud Manager. É possível procurar por recursos do Kubernetes em qualquer cluster e filtrar sua procura pelos campos de recurso. Os resultados da procura são baseados fora de seus objetos de cluster. Por exemplo, é possível procurar por `created` (o quão recentemente o objeto foi criado) ou por `cluster` (o cluster no qual o objeto se encontra).

Também é possível configurar seu próprio logotipo na `Página de login`, em `Sobre o modal` e no `Cabeçalho comum` executando `kubectl edit configmap platform-ui-config -n kube-system`.

Agora é possível atualizar os repositórios do Helm individualmente. Além de atualizar todos os seus repositórios do Helm na página de liberações do Helm com um único clique, também é possível atualizar cada repositório do Helm individualmente. Consulte Gerenciando repositórios do Helm para obter informações adicionais.

IBM Cloud PrivateAgora, o console de gerenciamento suporta brokers de serviço em nível de namespace. Para obter mais informações, consulte Gerenciando recursos do Catalog de serviço.

Os links *Ativar* de serviços na página *Liberações do Helm* são movidos para a página de detalhes da liberação. Os links que estavam na página *Liberações do Helm* que ativam os serviços agora estão disponíveis selecionando o nome da liberação na página *Liberações do Helm*. Pode ser necessário selecionar o nome da implementação para ver o link. Se um link estiver disponível para o serviço, selecione **Ativar** para testar o serviço.

Agora é possível implementar gráficos do Helm em namespaces remotos. Ao alavancar o IBM Multicloud Manager, é possível implementar gráficos do Helm em namespaces que estejam em clusters remotos, assim como namespaces que estejam em seu cluster local. Consulte Implementando gráficos do Helm no Catalog para obter mais informações.

É possível usar continuamente o Catalog no IBM Cloud Private e no IBM Multicloud Manager como um único plano de controle para gerenciar as cargas de trabalho implementadas em clusters locais e remotos.

O nome do cluster com o qual você está trabalhando agora está visível no cabeçalho do painel.

IBM Cloud Private CLI (cloudctl)

Agora é possível usar a CLI do IBM Cloud Private para gerenciar múltiplos clusters com o novo comando `mc`. Consulte Comandos de multicluster (mc) da CLI do IBM Cloud Private para aprender sobre os comandos `cloudctl mc` que podem ser executados para acessar seus clusters do IBM Multicloud Manager.

Execute os novos comandos da CLI `cloudctl iam oauth-client` do IBM Cloud Private para simplificar a integração e o gerenciamento de cargas de trabalho. Consulte Comandos iam da CLI do IBM Cloud Private (iam).

IBM Cloud Private Cloud Foundry e o Cloud Foundry Enterprise Environment

Para obter os detalhes das mudanças no IBM Cloud Private Cloud Foundry e no Cloud Foundry Enterprise Environment, consulte O que há de novo no IBM Cloud Private Cloud Foundry e no Cloud Foundry Enterprise Environment Versão 3.2.0.

IBM Multicloud Manager

Visualize e monitore múltiplos clusters com o IBM Multicloud Manager. É possível assegurar que seus clusters estejam protegidos, operando de forma eficiente e entregando os níveis de serviço que os aplicativos esperam ao configurar o IBM Multicloud Manager com seu cluster do IBM Cloud Private. Consulte a Visão geral da configuração do IBM Multicloud Manager para obter os tópicos de configuração.

O IBM Multicloud Manager agora está integrado ao IBM Cloud Private. A console de gerenciamento do IBM Cloud Private agora exibe o IBM Multicloud Manager sem uma interface separada.

Com a integração, agora é possível configurar o IBM Multicloud Manager durante ou após a instalação do IBM Cloud Private. Além disso, é possível `importar` recursos para gerenciar vários clusters usando a CLI do IBM Cloud Private. Saiba mais sobre as mudanças do IBM Multicloud Manager a partir das descrições a seguir:

- Configure o IBM Multicloud Manager durante a instalação do IBM Cloud Private customizando seu arquivo `config.yaml`. Veja Customizando o cluster com o arquivo config.yaml. Além disso, é possível configurar após a instalação.
- Com o novo comando da CLI `cloudctl mc cluster import` do IBM Cloud Private, é possível importar clusters a partir de diferentes provedores de nuvem do Kubernetes, incluindo o IBM Cloud Private. Depois de configurar seu arquivo e executar o `cloudctl mc cluster import`, o cluster de destino se torna um managed-cluster para o hub-cluster do IBM Multicloud Manager.
- Além disso, é possível gerenciar um cluster independente do OpenShift. Consulte todas as opções para importar e gerenciar clusters em Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager.

Consulte os pacotes opcionais que estão disponíveis em Instalando pacotes do IBM Multicloud Manager, que inclui o Federation-v2.

Também é possível configurar o registro de serviço do IBM Multicloud Manager para seus managed-clusters do IBM Multicloud Manager para descobrir serviços Kubernetes, como serviços Ingress e Istio. Consulte Trabalhando com a descoberta de serviço do IBM Multicloud Manager para obter mais informações.

Visualização de tecnologia

As visualizações de tecnologia a seguir são novas para esta versão. Para conhecer todos os recursos disponíveis no IBM® Cloud Private como código de visualização de tecnologia (TPC), consulte a seção Visualização de tecnologia.

Serviço de funcionamento do sistema

Agora é possível ativar o serviço de funcionamento do sistema para entender o funcionamento de seu sistema IBM Cloud Private. Para obter mais informações, consulte Serviço de funcionamento do sistema IBM Cloud Private.

Controlador de política de mutação

O IBM Cloud Private agora inclui um controlador de política de mutação para cumprir políticas de mutação. Crie uma política de mutação para relatar pods mudados a partir de imagens digitalizadas. Para obter informações adicionais, consulte o Controlador de política de mutação.

Instalando o IBM Cloud Private com o IBM Cloud Kubernetes Service

É possível implementar remotamente o IBM Cloud Paks em um cluster do IBM Cloud Kubernetes Service usando o IBM Multicloud Manager. Para obter mais informações, consulte Instalando o IBM Cloud Private com o IBM Cloud Kubernetes Service.

Instalando o Knative no IBM Cloud Private

Agora é possível instalar um gráfico do Knative em seu cluster do IBM Cloud Private 3.2.0. Para obter mais informações, consulte Instalando o Knative no IBM Cloud Private.

Definindo configurações de failover para seus clusters do IBM Multicloud Manager

Para seus clusters do IBM Multicloud Manager, é possível preparar o Minio e, em seguida, configurar o failover para IBM Multicloud Manager para seus hub-clusters e managed-clusters. Para obter detalhes completos, consulte Definindo configurações de failover para seus clusters do IBM Multicloud Manager.

APIs

A documentação para as APIs do Helm Tiller está agora disponível. Consulte APIs de REST do Helm Tiller para obter mais informações.

Mudanças na versão do pacote

Com a introdução do IBM Cloud Private versão 3.2.0, as versões do pacote a seguir foram mudadas:

Pacote	Versão	Nota
Kubernetes	1.13.5	Atualizado da versão 1.12.4
Docker	18.06.2	Além das outras versões suportadas
Controlador de ingresso NGINX	0.23.0	Atualizado da versão 0.21.0
GlusterFS	4.1.5	Atualizado a partir da versão 4.0.2
Calico	3.5.2	Atualizado da versão 3.3.1
CLI do Helm	2.12.3	Submetido a upgrade a partir da versão 2.9.1
Helm Tiller	2.12.3	
Istio	1.0.2	Submetido a upgrade a partir da versão 1.0.0
Serviço Catalog	0.1.40	Atualizado da versão 0.1.26

Resolução de

Problemas e Suporte

Para depurar seus problemas, é possível ver se o problema relatado foi corrigido na liberação. Para obter a lista, consulte Problemas relatados corrigidos.

Problemas e Limitações Conhecidos

Revise os problemas conhecidos para a versão 3.2.0.

A cota de recurso pode não atualizar

Talvez você ache que a cota de recurso não está atualizando no cluster. Isso é devido a um problema no kube-controller-manager. A solução alternativa é parar o contêiner do líder do kube-controller-manager nos nós principais e permitir que ele seja reiniciado. Se a alta disponibilidade estiver configurada para o cluster, é possível verificar o log do kube-controller-manager para localizar o líder. Apenas o líder kube-controller-manager está funcionando. Os outros controladores esperam ser escolhidos como o novo líder quando o líder atual estiver inativo.

Por exemplo:

```
# docker ps | grep hyperkube | grep controller-manager
97bccea493ea        4c7c25836910
"/hyperkube controll…"   7 days ago          Up 7 days                                  k8s_controller-
manager_k8s-master-9.111.254.104_kube-system_b0fa31e0606015604c409c09a057a55c_2
```

Para parar o líder, execute o comando a seguir com o ID do processo do Docker:

```
docker rm -f 97bccea493ea
```

O contêiner falha ao iniciar devido ao problema do Docker

A instalação falha durante a criação do contêiner devido a um problema do Docker 18.03.1. Se você tiver um subPath na montagem do volume, poderá receber o erro a seguir do serviço kubelet, que falha ao iniciar o contêiner:

```
Error: failed to start container "heketi": Error response from daemon: OCI runtime create failed:
container_linux.go:348: starting container process caused "process_linux.go:402: container init
caused \"rootfs_linux.go:58: mounting \\\"/var/lib/kubelet/pods/7e9cb34c-b2bf-11e8-a9eb-
0050569bdc9f/volume-subpaths/heketi-db-secret/heketi/0\\\" to rootfs
\\\"/var/lib/docker/overlay2/ca0a54812c6f5718559cc401d9b73fb7ebe43b2055a175ee03cdffaffada2585/merged
\\\" at
\\\"/var/lib/docker/overlay2/ca0a54812c6f5718559cc401d9b73fb7ebe43b2055a175ee03cdffaffada2585/merged
/backupdb/heketi.db.gz\\\" caused \\\"no such file or directory\\\"\"": unknown
```

Para obter mais informações, consulte a Documentação do Kubernetes.

Para resolver esse problema, exclua o pod com falha e tente a instalação novamente.

O Key Management Service deve ser implementado em um nó de gerenciamento em uma plataforma Linux®

O Key Management Service é implementado no nó de gerenciamento e é suportado somente na plataforma Linux®. Se não houver nenhum nó de gerenciamento `amd64` no cluster, o Key Management Service não será implementado.

A afinidade do cookie não funciona quando o FIPS está ativado

Quando um Federal Information Processing Standard (FIPS) é ativado, a afinidade de cookie não funciona porque o `nginx.ingress.kubernetes.io/session-cookie-hash` pode ser configurado apenas no `sha1/md5/index`, que não é suportado no modo FIPS.

A IU do Grafana não pode ser aberta após o upgrade da versão de liberação do serviço de

monitoramento

Se o volume persistente estiver ativado para o serviço de monitoramento e a senha do Grafana não estiver configurada durante a instalação do serviço de monitoramento, o painel Grafana não estará acessível após o upgrade para uma versão mais recente. É possível ver a mensagem de erro {"message":"Invalid username or password"} ao tentar acessar o painel Grafana. É possível resolver esse problema antes de fazer upgrade da liberação de monitoramento ou após o upgrade.

Etapas de pré-upgrade

1. Obtenha a senha do Grafana por meio de `monitoring-grafana-secret`:

   ```
   export PASSWORD=$(kubectl get -n kube-system secret/monitoring-grafana-secret -o yaml|grep
   password|awk -F': ' '{print $2}'|base64 -d)
   ```

2. Configure a senha do Grafana durante o upgrade. Se você fizer upgrade usando o painel do IBM Cloud Private, configure a senha Grafana na página de configuração durante o upgrade. Se você fizer upgrade usando a CLI, siga estas etapas:

   1. Obtenha o `values.yaml` da liberação existente:

      ```
      helm get values --tls monitoring >> values.yaml
      ```

   2. Insira a senha do Grafana da Etapa 1 no `values.yaml`:

      ```
      grafana:
      password: PASSWORD
      ```

   3. Execute o comando de upgrade do Helm:

      ```
      helm upgrade --tls monitoring -f values.yaml ibm-icpmonitoring-1.3.0.tar.gz
      ```

Etapas pós-upgrade

1. Obtenha a senha do Grafana por meio de `monitoring-grafana-secret`:

   ```
   export PASSWORD=$(kubectl get -n kube-system secret/monitoring-grafana-secret -o yaml|grep
   password|awk -F': ' '{print $2}'|base64 -d)
   ```

Nota: a opção `-d` para o comando base64 é usada para decodificar a senha codificada. Essa opção pode variar em diferentes sistemas de operação. Por exemplo, no macOS, essa opção é `-D`.

2. Obtenha o nome do pod do Grafana:

```
export GRAFANA_POD=$(kubectl get pod -n kube-system|grep grafana|grep Running|awk '{split($0,
a, " "); print a[1]}')
```

3. Reconfigure a senha para o contêiner do Grafana:

```
kubectl exec -n kube-system $GRAFANA_POD -c grafana -it -- grafana-cli admin reset-admin-
password --homepath "/usr/share/grafana" $PASSWORD
```

Nota: se você resolver esse problema após o upgrade, o problema poderá ocorrer novamente após o retrocesso. Deve-se seguir as mesmas etapas de pós-upgrade para corrigi-lo.

O Tiller 2.7.2 não suporta o upgrade ou a instalação de recursos do Kubernetes 1.9 - 1.10

O Tiller versão 2.9.1 é instalado com o IBM Cloud Private versão 3.2.0. O Tiller 2.9.1 usa o Kubernetes API versão 1.8 e 1.9. Não é possível instalar ou fazer upgrade dos gráficos Helm que usam apenas os recursos do Kubernetes versão 1.10.

Você pode encontrar um erro de upgrade de liberação do Helm. A mensagem de erro será semelhante ao seguinte conteúdo:

```
Erro: FALHA DE UPGRADE: falha ao criar correção: não é possível localizar o campo da api na
estrutura Não estruturada para o campo json "spec"
```

Se você encontra essa mensagem de erro, deve-se excluir a liberação e instalar uma nova versão do gráfico.

Páginas de alerta, de criação de log ou de monitoramento exibem 500 Internal Server Error

Para resolver este problema, conclua as seguintes etapas no nó principal:

1. Crie um alias para o login da api kubectl não seguro executando o seguinte comando:

```
Alias kc = 'kubectl -n kube-system'
```

2. Edite o mapa de configuração para Kibana. Execute o comando a seguir:

```
kc edit cm kibana-nginx-config
```

Inclua as atualizações a seguir:

```
upstream kibana {
server localhost:5602;
}
Change localhost to 127.0.0.1
```

3. Localize e reinicie o pod Kibana executando os seguintes comandos:

```
kc get pod | grep -i kibana

kc delete pod <kibana-POD_ID>
```

4. Edite o mapa de configuração para Grafana executando o seguinte comando:

```
kc edit cm grafana-router-nginx-config
```

Inclua as atualizações a seguir:

```
upstream grafana {
server localhost:3000;
}
Change localhost to 127.0.0.1
```

5. Localize e reinicie o pod Grafana executando os seguintes comandos:

```
kc get pod | grep -i monitoring-grafana

kc delete pod <monitoring-grafana-POD_ID>
```

6. Edite o mapa de configuração para o Alertmanager executando o seguinte comando:

```
kc edit cm alertmanager-router-nginx-config
```

Inclua as atualizações a seguir:

```
upstream alertmanager {
server localhost:9093;
}
Change localhost to 127.0.0.1
```

7. Localize e reinicie o Alertmanager executando os seguintes comandos:

```
kc get pod | grep -i monitoring-prometheus-alertmanager
```

```
kc delete pod <monitoring-prometheus-alertmanager-POD_ID>
```

IPv6 não é suportado

O IBM Cloud Private não pode usar redes IPv6. Comente as configurações no arquivo `/etc/hosts` em cada nó do cluster para remover as configurações de IPv6. Para obter informações adicionais, consulte Configurando seu cluster.

Não é possível efetuar login no console de gerenciamento com um usuário LDAP após a reinicialização do mestre líder

Se não for possível efetuar login na console de gerenciamento após reiniciar o nó principal inicial em um cluster de alta disponibilidade, execute as ações a seguir:

1. Efetue login no console de gerenciamento com as credenciais do administrador de cluster. O nome do usuário é `admin` e a senha é `admin`.
2. Clique em **Menu** > **Gerenciar** > **Identidade e Acesso** .
3. Clique em **Editar** e, em seguida, clique em **Salvar**.

   Nota: Os usuários LDAP podem efetuar login no console de gerenciamento.

Se o problema persistir, o MongoDB e os pods que dependem do `auth-idp` podem não estar em execução. Siga estas instruções para identificar a causa.

1. Verifique se o pod MongoDB está em execução sem erros.
   - Use o comando a seguir para verificar o status do pod. O pod deve mostrar o status como `1/1 Running`. Verifique os logs, se necessário.

     ```
     kubectl -n kube-system get pods | grep -e mongodb
     ```

   - Se o pod não mostrar o status como `1/1 Running`, reinicie o pod excluindo-o.

     ```
     kubectl -n kube-system delete pod -l app=icp-mongodb
     ```

     Aguarde um ou dois minutos para que o pod seja reiniciado. Verifique o status do pod usando o seguinte comando. O status deve mostrar `1/1 Em execução`.

     ```
     kubectl -n kube-system get pods | grep -e mongodb
     ```

2. Depois que o pod do MongoDB estiver em execução, reinicie os pods `auth-idp` excluindo-os.

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

   Aguarde um minuto ou dois para os pods reiniciarem. Verifique o status do pod usando o seguinte comando. O status deve mostrar `4/4 Em execução`.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Limitação de prefixo do Calico nos nós do Linux® on Power® (ppc64le)

Se você instalar o IBM Cloud Private nas LPARs do PowerVM Linux e seus dispositivos Ethernet virtuais usarem o prefixo `ibmveth`, será necessário configurar o adaptador de rede para usar a rede do Calico. Durante a instalação, certifique-se de configurar um valor de parâmetro `calico_ip_autodetection_method` no arquivo `config.yaml`. A configuração é semelhante ao conteúdo a seguir:

```
calico_ip_autodetection_method: interface=<device_name>
```

O parâmetro `<device_name>` é o nome de seu adaptador de rede. Você deve especificar a interface `ibmveth0` em cada nó do cluster, incluindo os nós do trabalhador.

Nota: se você usou o PowerVC para implementar seu nó do cluster, esse problema não afetará você.

A sincronização de repositórios pode não atualizar o conteúdo do gráfico Helm

A sincronização de repositórios leva vários minutos para ser concluída. Enquanto a sincronização está em andamento, pode haver um erro se você tentar exibir o arquivo leia-me. Após a sincronização ser concluída, é possível visualizar o arquivo leia-me e implementar o gráfico.

Alguns recursos não estão disponíveis na nova console de gerenciamento

O IBM Cloud Private 3.2.0 suporta apenas a nova console de gerenciamento. Algumas opções do console anterior ainda não estão disponíveis. Para acessar as opções do console anterior, você deve usar a CLI `kubectl` para as funções.

O console de gerenciamento exibe 502 Bad Gateway Error

O console de gerenciamento exibe um `Erro 502 Bad Gateway` após instalar ou reinicializar o nó principal.

Se você tiver instalado o IBM Cloud Private recentemente, aguarde alguns minutos e recarregue a página.

Se você reinicializou o nó principal, execute as etapas a seguir:

1. Obtenha os endereços IP dos pods `icp-ds`. No nó principal, execute o comando a seguir:

   ```
   kubectl get pods -o wide -n kube-system | grep "icp-ds"
   ```

   A saída se assemelha ao texto a seguir:

   ```
   icp-ds-0                                                  1/1       Running       0          1d
   10.1.231.171   10.10.25.134
   ```

   Nesse exemplo, `10.1.231.171` é o endereço IP do pod.

   Em ambientes de alta disponibilidade (HA), existe um pod `icp-ds` para cada nó principal.

2. No nó principal, execute ping nos pods `icp-ds`. Verifique o endereço IP para cada pod `icp-ds` executando o comando a seguir para cada endereço IP:

   ```
   ping 10.1.231.171
   ```

   Se a saída for semelhante ao texto a seguir, deve-se excluir o pod:

   ```
   connect: Invalid argument
   ```

3. No nó principal, exclua cada pod que não está responsivo executando o seguinte comando:

   ```
    Kubectl delete pods icp-ds-0 -n kube-system
   ```

   Nesse exemplo, `icp-ds-0` é o nome do pod não responsivo.

   Importante: Em instalações de HA, pode ser necessário excluir o pod para cada nó principal.

4. No nó principal, obtenha o endereço IP do pod ou pods de substituição executando o seguinte comando:

   ```
   kubectl get pods -o wide -n kube-system | grep "icp-ds"
   ```

   A saída se assemelha ao texto a seguir:

   ```
   icp-ds-0                                                  1/1       Running       0          1d
   10.1.231.172   10.10.2
   ```

5. No nó principal, efetue ping dos pods novamente e verifique o endereço IP para cada pod `icp-ds` executando o seguinte comando para cada endereço IP:

   ```
   ping 10.1.231.172
   ```

Se todos os pods `icp-ds` estiverem responsivos, será possível acessar a console de gerenciamento do IBM Cloud Private quando esse pod entrar no estado disponível.

Ativar o Controlador de ingresso para usar um novo prefixo de anotação

- A anotação de ingresso NGINX contém um novo prefixo na versão 0.9.0 que é usado no `nginx.ingress.kubernetes.io` do IBM Cloud Private 3.2.0. Essa mudança usa a sinalização para evitar quebras em implementações que estão em execução.
    - Para evitar dividir um controlador de ingresso NGINX em execução, inclua a sinalização `--annotations-prefix=ingress.kubernetes.io` na implementação do controlador de ingresso nginx. Por padrão, o produto aceita a sinalização no controlador de ingresso do IBM Cloud Private.
- Se desejar usar a nova anotação de ingresso, atualize o controlador de ingresso removendo a sinalização `--annotations-prefix=ingress.kubernetes.io`. Para remover a sinalização, execute os seguintes comandos:

  Nota: Execute os seguintes comandos a partir do nó principal.

  Para o Linux®, execute o seguinte comando:

  ```
  kubectl edit ds nginx-ingress-lb-amd64 -n kube-system
  ```

  Para Linux® on Power® (ppc64le) execute o comando a seguir:

  ```
  kubectl edit ds nginx-ingress-lb-ppc64le -n kube-system
  ```

  Salve e saia para implementar a mudança. O controlador de ingresso é reiniciado para receber a nova configuração.

Os dados de monitoramento não serão retidos se você usar um volume fornecido dinamicamente durante o upgrade

Se você usar um volume persistente fornecido dinamicamente para armazenar dados de monitoramento, os dados serão perdidos após o upgrade do serviço de monitoramento de 2.1.0.2 para 2.1.0.3.

Não é possível reiniciar o nó ao usar o armazenamento do vSphere que não possui uma réplica

Encerrar um cluster em um ambiente IBM Cloud Private que usa o vSphere Cloud move o pod para outro nó em seu cluster. No entanto, o volume do vSphere que o pod usa no nó original não é removido do nó. Pode ocorrer um erro quando você tentar reiniciar o nó.

Para resolver o problema, primeiro remova o volume do nó. Em seguida, reinicie o nó.

Os rótulos truncados são exibidos no painel para algumas linguagens

Se você acessar o painel do IBM Cloud Private em idiomas diferentes do inglês a partir do navegador Mozilla Firefox em um sistema que usa um sistema operacional Windows™, alguns rótulos podem ficar truncados.

Os nomes de repositórios do Helm não podem conter caracteres DBCS GB18030

Não use caracteres DBCS GB18030 no nome do repositório do Helm ao incluir o repositório.

O cluster GlusterFS se torna inutilizável se você configurar um vSphere Cloud Provider após a instalação do IBM Cloud Private

Por padrão, o kubelet usa o endereço IP do nó como o nome do nó. Quando você configura um vSphere Cloud Provider, o kubelet usa o nome do host do nó como o nome do nó. Se você teve seu cluster GlusterFS configurado durante a instalação do IBM Cloud Private, o Heketi cria uma topologia usando o endereço IP do nó.

Ao configurar um vSphere Cloud Provider depois de instalar o IBM Cloud Private, o cluster GlusterFS se torna inutilizável porque o kubelet identifica nós por seus nomes de host, mas o Heketi ainda usa endereços IP para identificar os nós.

Se você planejar usar o GlusterFS e um vSphere Cloud Provider no cluster do IBM Cloud Private, certifique-se de configurar `kubelet_nodename: hostname` no arquivo `config.yaml` durante a instalação.

A origem de dados do Prometheus é perdida durante um retrocesso do

IBM Cloud Private

Ao retroceder do IBM Cloud Private Versão 3.2.0 para o 3.1.2, a origem de dados do Prometheus no Grafana é perdida. Os painéis do Grafana não exibem nenhuma métrica.

Para resolver o problema, inclua novamente a origem de dados do Prometheus concluindo as etapas na seção Configurar manualmente uma origem de dados do Prometheus no Grafana.

A varredura de imagens de arquitetura cruzada do orientador de vulnerabilidade não funciona com versões do `glibc` anteriores à 2.22

O Vulnerability Advisor (VA) agora suporta a varredura de imagem de arquitetura cruzada com o QEMU (Quick EMUlator). É possível varrer as imagens de arquitetura de CPU do Linux® on Power® (ppc64le) com o VA em execução em nós do Linux®. Como alternativa, é possível varrer as imagens de arquitetura de CPU Linux com o VA em execução em nós do Linux® on Power® (ppc64le).

Ao fazer a varredura de imagens do Linux, deve-se utilizar o `glibc` versão 2.22 ou posterior. Ao usar uma versão do `glibc` anterior à 2.22, é possível que a varredura não funcione quando o orientador de vulnerabilidade for executado em nós do Linux® on Power® (ppc64le). As versões do Glibc anteriores à 2.22 fazem determinados syscalls (time/vgetcpu/getttimeofday) utilizando mecanismos vsyscall. A implementação do syscall tenta acessar o endereço estático codificado permanentemente, o qual a QEMU falha ao converter durante a execução no modo de emulação.

O contêiner falha ao operar ou ocorre um pânico do kernel

O erro a seguir pode ocorrer a partir do console do nó ou do log do kernel do IBM Cloud Private:

```
kernel:unregister_netdevice: waiting for <eth0> to become free.
```

Se você receber esse erro, o log exibirá `kernal:unregister_netdevice: esperando a liberação de <eth0>` e `os contêineres falham ao operar`. Continuar a solucionar problemas. Se você atender a todas as condições necessárias, reinicialize o nó.

Visualize https://github.com/kubernetes/kubernetes/issues/64743 para saber sobre o bug do kernel Linux que causa o erro.

Falha intermitente ao efetuar login no console de gerenciamento em clusters de HA que usam NSX-T 2.3 ou 2.4

Em clusters de HA que usam NSX-T 2.3 ou 2.4, talvez você não possa efetuar login no console de gerenciamento. Depois de especificar as credenciais de login, você será redirecionado para a página de login. Talvez seja necessário efetuar login várias vezes até que você consiga. Esta questão é intermitente.

A política do Vulnerability Advisor é reconfigurada para a configuração padrão após o upgrade de 3.1.2 no cluster ppc64le

Se você ativou o Vulnerability Advisor (VA) em seu cluster do Linux® on Power® (ppc64le) na 3.1.2, a política do Vulnerability Advisor será reconfigurada para a configuração padrão ao fazer upgrade para a 3.2.0. Para corrigir esse problema, reconfigure a política do VA no console de gerenciamento.

Os contêineres podem travar ao executar o IBM Cloud Private em guests do KVM on

POWER.

Se você estiver executando o IBM Cloud Private em guests do KVM on Power, alguns contêineres poderão travar devido a um problema na maneira como a Memória de Transação é manipulada. Como solução alternativa para esse problema, use um dos métodos a seguir:

- Desligue o suporte de Memória de Transação para guests do KVM on Power.
- Se você estiver usando o emulador Oemu diretamente para executar a máquina virtual, ative a opção `cap-htm=off`.
- Se você estiver usando a biblioteca libvirt, inclua o atributo XML a seguir na definição de domínio:

```
<features>
  <htm state='on'/>
</features>
```

Consulte a Documentação do libvirt para obter as instruções detalhadas sobre como incluir esse atributo libvirt. **Nota:** este problema é específico para guests do KVM on Power e não ocorre ao usar o bare metal POWER9 ou as LPARs do POWER9 PowerVM.

Os pods ELK de criação de log estão no estado *CrashLoopBackOff*

Os pods ELK de criação de log continuam aparecendo no estado *CrashLoopBackOff* após o upgrade para a versão atual e o aumento de memória.

Esse é um problema conhecido do no Elasticsearch 5.5.1.

Nota: se você tiver mais de um data-pod, repita as etapas de 1 a 8 para cada pod. Por exemplo, *logging-elk-data-0*, *logging-elk-data-1* ou *logging-elk-data-2*.

Conclua as etapas a seguir para resolver esse problema.

1. Verifique o log para descobrir o arquivo problemático que contém o problema de permissão.

   ```
   java.io.IOException: failed to write in data directory
   [/usr/share/elasticsearch/data/nodes/0/indices/dT4Nc7gvRLCjUqZQ0rIUDA/0/translog] write
   permission is required
   ```

2. Obtenha o endereço IP do nó de gerenciamento no qual o pod *logging-elk-data-1* está em execução.

   ```
   kubectl -n kube-system get pods -o wide | grep logging-elk-data-1
   ```

3. Use SSH para efetuar login no nó de gerenciamento.

4. Navegue para o diretório `/var/lib/icp/logging/elk-data`.

   ```
   cd /var/lib/icp/logging/elk-data
   ```

5. Localize todos os arquivos `.es_temp_file`.

   ```
   find ./ -name "*.es_temp_file"
   ```

6. Exclua todos os arquivos `*.es_temp_file` que você localizar na etapa 5.

   ```
   rm -rf *.es_temp_file
   ```

7. Exclua o pod *logging-elk-data-1* antigo.

   ```
   kubectl -n kube-system delete pods logging-elk-data-1
   ```

8. Aguarde de 3 a 5 minutos para que o novo pod *logging-elk-data-1* seja reiniciado.

   ```
   kubectl -n kube-system get pods -o wide | grep logging-elk-data-1
   ```

Os logs não funcionam depois que os pods de criação de log são reiniciados

Talvez ocorram os problemas a seguir:

- A IU da web do Kibana mostra o status de funcionamento do Elasticsearch como vermelho.
- As mensagens de log do pod do cliente do Elasticsearch indicam que o Search Guard não está inicializado. Observe que o mesmo erro se repete a cada alguns segundos. As mensagens são semelhantes às seguintes:

  ```
  [2018-11-08T20:43:54,380][ERROR][c.f.s.a.BackendRegistry  ] Not yet initialized (you may need
  to run sgadmin)
  [2018-11-08T20:43:54,487][ERROR][c.f.s.a.BackendRegistry  ] Not yet initialized (you may need
  to run sgadmin)
  [2018-11-08T20:43:54,488][ERROR][c.f.s.a.BackendRegistry  ] Not yet initialized (you may need
  to run sgadmin)
  ```

- Se o Vulnerability Advisor (VA) estiver instalado, uma mensagem de erro aparecerá em seus logs do VA que se assemelhem ao seguinte:

```
2018-10-31 07:25:12,083 ERROR 229 <module>: Error: TransportError(503, u'Search Guard not
initialized (SG11). See https://github.com/floragunncom/search-guard-
docs/blob/master/sgadmin.md', None)
```

Para resolver esse problema, conclua as etapas a seguir para executar uma tarefa de inicialização do Search Guard:

1. Salve a tarefa de inicialização do Search Guard existente em um arquivo.

   ```
   kubectl get job.batch/logging-elk-elasticsearch-tls-init -n kube-system -o yaml > sg-init-
   job.yaml
   ```

   Criação de log no IBM Cloud Private versão 3.2.0 mudada para remover a tarefa após a conclusão. Se você não tiver uma tarefa existente da qual extrair as configurações para um arquivo, será possível salvar o seguinte arquivo YAML no arquivo `sg-init-job.yaml`.

   ```
   apiVersion: batch/v1
   kind: Job
   metadata:
     labels:
       app: logging-ibm-<RELEASE_NAME>-elasticsearch
       chart: ibm-icplogging-2.2.0 # Verify this is the correct version of logging installed
       component: searchguard-init
       heritage: Tiller
       release: logging
     name: searchguard-init-job
     namespace: kube-system
   spec:
     backoffLimit: 6
     completions: 1
     parallelism: 1
     template:
       metadata:
         creationTimestamp: null
         labels:
           app: logging-ibm-<RELEASE_NAME>-elasticsearch
           chart: ibm-icplogging
           component: searchguard-init
           heritage: Tiller
           job-name: logging-ibm-<RELEASE_NAME>-elasticsearch-tls-init
           release: logging
           role: initialization
       spec:
         affinity:
           nodeAffinity:
             requiredDuringSchedulingIgnoredDuringExecution:
               nodeSelectorTerms:
               - matchExpressions:
                 - key: beta.kubernetes.io/arch
                    operator: In
                    values:
                   - amd64
                   - ppc64le
                   - s390x
                 - key: management
                    operator: In
                    values:
                   - "true"
          containers:
         - env:
           - name: APP_KEYSTORE_PASSWORD
             value: Y2hhbmdlbWU=
           - name: CA_TRUSTSTORE_PASSWORD
             value: Y2hhbmdlbWU=
           - name: ES_INTERNAL_PORT
             value: "9300"
           image: ibmcom/searchguard-init:2.0.1-f2 # This value may be different from the one on
   your system; double check by running docker image | grep searchguard-init
           imagePullPolicy: IfNotPresent
           name: searchguard-init
           resources: {}
           terminationMessagePath: /dev/termination-log
           terminationMessagePolicy: File
           volumeMounts:
           - mountPath: /usr/share/elasticsearch/config/searchguard
   ```

```
      name: searchguard-config
    - mountPath: /usr/share/elasticsearch/config/tls
      name: certs
      readOnly: true
  dnsPolicy: ClusterFirst
  restartPolicy: OnFailure
  schedulerName: default-scheduler
  securityContext: {}
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: NoSchedule
    key: dedicated
    operator: Exists
 volumes:
  - configMap:
      defaultMode: 420
      name: logging-ibm-<RELEASE_NAME>-elasticsearch-searchguard-config
    name: searchguard-config
  - name: certs
    secret:
      defaultMode: 420
      secretName: logging-ibm-<RELEASE_NAME>-certs
```

Notas:

1. Verifique se a versão de criação de log correta está instalada: `chart: ibm-icplogging-2.2.0`
2. Esta imagem pode ser diferente daquela em seu sistema: `image: ibmcom/searchguard-init:2.0.1-f2`. Execute o comando `docker image | grep searchguard-init` para confirmar se a imagem correta está instalada no sistema.
3. Siga as convenções de nomenclatura de instalação para nomear o arquivo `logging-ibm-<RELEASE_NAME>-elasticsearch....`

2. Edite o arquivo de tarefas.

   1. Remova tudo em metadata.*, exceto para os parâmetros a seguir:
      - `metadata.name`
      - `metadata.namespace`
      - `metadata.labels.*`
   2. Mude `metadata.name` e `spec.template.metadata.job-name` para novos nomes.
   3. Remova `spec.selector` e `spec.template.metadata.labels.controller-uid`
   4. Remova `status.*`

3. Salve o arquivo.

4. Execute a tarefa.

   ```
   kubectl apply -f sg-init-job.yaml
   ```

Tempos limites e telas em branco ao exibir mais de 80 namespaces

Se um cluster tiver um número grande de namespaces, mais de 80, poderão ocorrer os problemas a seguir:

- A página de visão geral do namespace pode atingir o tempo limite e exibir uma tela em branco.
- A página *Configuração de implementação do gráfico* pode atingir o tempo limite e não carregar todos os namespaces no menu suspenso. Apenas o namespace `default` é mostrado para a implementação.

A criptografia do tráfego de rede de dados do cluster com IPsec não funciona no sistema operacional

SLES 12 SP3

O strongSwan versão 5.3.3 ou superior é necessário para implementar a configuração de malha de IPsec para a criptografia de tráfego de rede de dados do cluster. No SUSE Linux Enterprise Server (SLES) 12 SP3, a versão do strongSwan padrão é 5.1.3, que não é adequada para a configuração de malha do IPsec.

A clonagem de um nó do trabalhador do IBM Cloud Private não é suportada

O IBM Cloud Private não suporta a clonagem de um nó do trabalhador do IBM Cloud Private existente. Não é possível mudar o nome do host e o endereço IP de um nó em seu cluster existente.

Deve-se incluir um novo nó do trabalhador. Para obter mais informações, consulte Incluindo um nó do cluster do IBM Cloud Private.

A procura LDAP não mostra sugestões automaticamente no keypress

Ao incluir usuários ou grupos de usuários em sua equipe, é possível procurar por usuários e grupos individuais. À medida que você digita na barra de procura LDAP, sugestões que estão associadas à consulta de procura não aparecem automaticamente. Deve-se pressionar a tecla Enter para obter resultados do servidor LDAP. Para obter mais informações, consulte Criar equipes .

As APIs do Key Management Service retornam um erro 502 Bad Gateway

Ao chamar qualquer API do Key Management Service, você vê um erro `502 Bad Gateway`. Esse erro é devido a um problema com a integração do serviço PEP ao IAM. Instale o caminho `key-management-pep-3.1.2-21233-20190208.tar.gz` do Fix Central para resolver o problema. Depois de aplicar a correção, não será necessário reimplementar suas liberações do Helm. Você deve reaplicar a correção se substituir seu nó de gerenciamento.

O Elasticsearch não funciona com o GlusterFS

O Elasticsearch não funciona corretamente com o GlusterFS que está configurado em um ambiente do IBM® Cloud Private. Esse problema é devido ao seguinte erro `AlreadyClosedException`. Para obter mais informações, consulte Red Hat Bugzilla-Bug 1430659.

```
[ 2019-01-17T10:53:49, 750 ] [ WARN ] [o.e.c.a.s.ShardStateAction ] [ logging-elk-master-
7df4b7b7bdfc-5spqc ] \ [logstash-2019.01.16][3] received shard failed for shard id [[logstash-
2019.01.16][3]], allocation id \
[n9ZpABWfS4qJCyUIfEgHWQ], primary term [0], message [shard failure, reason \
[already closed by tragic event on the index writer]], \
failure [AlreadyClosedException[Underlying file changed by an external force at 2019-01-
17T10:44:48.410502Z,\
(lock=NativeFSLock(path=/usr/share/elasticsearch/data/nodes/0/indices/R792nkojQ7q1UCYSEO4trQ/3/index
/write.lock,\
impl=sun.nio.ch.FileLockImpl[0:9223372036854775807 exclusive valid],creationTime=2019-01-
17T10:44:48.410324Z))]]
org.apache.lucene.store.AlreadyClosedException: Underlying file changed by an external force at
2019-01-17T10:44:48.410502Z,\
(lock=NativeFSLock
(path=/usr/share/elasticsearch/data/nodes/0/indices/R792nkojQ7q1UCYSEO4trQ/3/index/write.lock [
0:9223372036854775807 exclusive valid ] ,creationTime=2019-01-17T10:44:48.48.410324Z))
```

O recurso do IAM que foi incluído a partir da CLI é sobrescrito pela console de gerenciamento]

Se você atualizar um recurso de equipe que possui um recurso de liberação do Helm designado a ele a partir da interface da linha de comandos (CLI) e da console de gerenciamento, o recurso será não designado. Se você gerenciar recursos de liberação do Helm, inclua o recurso a partir da CLI. Se gerenciar os recursos da liberação do Helm a partir da console de gerenciamento, você pode observar que um recurso da liberação do Helm está listado de forma incorreta como um *Namespace*. Para obter mais informações, consulte Gerenciando liberações do Helm.

Gerencie o recurso de liberação do Helm a partir da CLI para obter as informações mais precisas do recurso da equipe. Para obter mais informações, consulte Trabalhando com gráficos.

Os pods mostram CreateContainerConfigError

Depois de instalar o IBM Cloud Private, os seguintes pods mostram o erro `CreateContainerConfigError`.

```
# kubectl get pods -o wide --all-namespaces |grep -v "Running" |grep -v "Completed"
NAMESPACE      NAME                                      READY    STATUS
kube-system    logging-elk-kibana-init-6z95k             0/1      CreateContainerConfigError
kube-system    metering-dm-79d6f5894d-q2qpm              0/1      Init:CreateContainerConfigError
kube-system    metering-reader-4tzgz                     0/1      Init:CreateContainerConfigError
kube-system    metering-reader-5hjvm                     0/1      Init:CreateContainerConfigError
kube-system    metering-reader-gsm44                     0/1      Init:CreateContainerConfigError
kube-system    metering-ui-7dd45b4b6c-th2pg              0/1      Init:CreateContainerConfigError
kube-system    secret-watcher-6bd4675db7-mcb64           0/1      CreateContainerConfigError
kube-system    security-onboarding-262cp                 0/1      CreateContainerConfigError
```

O problema ocorre quando os pods são incapazes de criar o segredo da chave de API do IAM.

Para resolver o problema, reinicie o pod `auth-pdp`.

Conclua as etapas a seguir:

1. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Obtenha o ID do pod `auth-pdp` e anote-o.

   ```
   kubectl -n kube-system get pods -o wide | grep auth-pdp
   ```

3. Exclua o pod `auth-pdp`.

   ```
   kubectl -n kube-system delete pod <auth-pdp-pod-id>
   ```

4. Aguarde dois minutos e verifique o status do pod.

   ```
   kubectl -n kube-system get pods -o wide | grep auth-pdp
   ```

   O status do pod é mostrado como `Em execução`.

Alguns pods não estão iniciando ou registram erros de handshake de TLS no ambiente IBM Power

Em alguns casos, quando você está usando o tunelamento IP-IP em um ambiente IBM Power, alguns de seus Pods não iniciam ou contêm entradas de log que indicam erros de handshake de TLS. Se você observar algum desses problemas, conclua as seguintes etapas para resolvê-lo:

1. Execute o comando `ifconfig` ou o comando `netstat` para visualizar as estatísticas do dispositivo de túnel. O dispositivo de túnel é geralmente é chamado *tunl0*.
2. Observe as mudanças na contagem reduzida de TX que é exibida durante a execução do comando `ifconfig` ou do comando `netstat`.

   Se você usar o comando `netstat`, insira um comando semelhante ao seguinte:

   ```
   netstat --interface=tunl0
   ```

   A saída deve ser semelhante ao seguinte conteúdo:

   ```
   Kernel Interface table
   Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
   tunl0     1300   904416      0      0 0        714067      0    806      0 ORU
   ```

   Se você usar o comando `ifconfig`, execute um comando semelhante ao seguinte:

   ```
   ifconfig tunl0
   ```

   A saída deve ser semelhante ao seguinte conteúdo:

   tunl0: flags=193 mtu 1300

   ```
   inet 10.1.125.192  netmask 255.255.255.255
   tunnel   txqueuelen 1000  (IPIP Tunnel)
   RX packets 904377  bytes 796710714 (759.8 MiB)
   RX errors 0  dropped 0  overruns 0  frame 0
   TX packets 714034  bytes 125963495 (120.1 MiB)
   TX errors 0  dropped 806 overruns 0  carrier 0  collisions 0
   ```

3. Execute o comando novamente e anote a mudança na contagem reduzida de TX que é exibida durante a execução do comando `ifconfig`, ou na contagem TX-DRP que é exibida durante a execução do comando `netstat`.

   Se o valor estiver aumentando continuamente, há um problema de MTU. Para resolvê-lo, ative `tcp_mtu_probing` e reduza o valor de MTU do dispositivo de túnel.

4. Execute os seguintes comandos para ativar `tcp_mtu_probing`:

   ```
   echo 1 > /proc/sys/net/ipv4/tcp_mtu_probing
   echo 1024 > /proc/sys/net/ipv4/tcp_base_mss
   ```

5. Inclua as seguintes linhas no arquivo `/etc/sysctl.conf` para tornar as configurações permanentes para futuras reinicializações do sistema:

   ```
   net.ipv4.tcp_mtu_probing =1
   net.ipv4.tcp_base_mss = 1024
   ```

6. Conclua as seguintes etapas para mudar a MTU de túnel IP-IP do Calico após sua implementação:

   1. Atualize a configuração para `veth_mtu` executando o seguinte comando:

      ```
      kubectl edit cm calico-config -n kube-system
      ```

   2. Reinicie os PODs calico-node para que as mudanças entrem em vigor inserindo o seguinte comando:

      ```
      kubectl patch ds calico-node -n kube-system -p '{"spec":{"template":{"spec":
      {"containers":[{"name":"calico-node","env":[{"name":"RESTART_","value":"'$(date
      +%s)'"}]}]}}}}'
      ```

7. Aplique estas configurações ao arquivo `sysctl.conf` e a todos os nós no cluster.

Limitações conhecidas do IBM Cloud Private no Linux on IBM Z and LinuxONE

O IBM Cloud Private no Linux on IBM Z and LinuxONE tem as seguintes limitações:

- Uma arquitetura combinada, como o nó principal no IBM Z ou no LinuxONE e os nós do trabalhador ou do proxy no Linux ou no Linux® on Power® (ppc64le), não pode ser usada em um ambiente de produção, já que ela é um recurso de visualização de tecnologia.
- O componente de criação de log do IBM Cloud Private nos nós de gerenciamento da plataforma IBM® Z é suportado apenas como uma visualização de tecnologia na liberação da 3.2.0. Para obter informações adicionais, consulte Criação de log do IBM Cloud Private com o IBM® Z.
- Para obter uma lista de plataformas e recursos suportados, consulte Sistemas operacionais e plataformas suportados.
- Para obter os requisitos de hardware para o ambiente do Linux on IBM Z and LinuxONE, consulte Requisitos do ambiente do Linux on IBM Z and LinuxONE.

A verificação de atividade ou de prontidão de pod pode falhar porque o Docker

falhou ao executar alguns comandos no contêiner

Depois de fazer upgrade do IBM Cloud Private versão 3.1.0, 3.1.1 ou 3.1.2 para a versão 3.2.0, as verificações de atividade ou de prontidão para alguns pods podem falhar. Essa falha também pode ocorrer ao implementar uma carga de trabalho no cluster ou ao encerrar ou reiniciar o nó de gerenciamento no cluster. Esse problema pode ocorrer nos pods do Prometheus, nos pods do Grafana ou em outros pods.

Dependendo de fatores, como a estabilidade de rede, as análises de prontidão e de atividade podem levar mais tempo para iniciar do que o tempo permitido antes que uma solicitação de prontidão seja enviada. Se elas não forem iniciadas quando uma solicitação for enviada, elas não retornarão um status de pronto.

A solicitação de status retornada é semelhante ao exemplo a seguir:

```
# kubectl get pods -o wide --all-namespaces |grep monitor
kube-system    monitoring-grafana-59bfb7859b-f9zrd                2/3     Running     0
43m     10.1.1.1   9.1.1.1    <none>           <none>
kube-system    monitoring-prometheus-75b7444496-zzl7b                        3/4     Running     0
43m     10.1.1.1    9.1.1.1    <none>           <none>
```

Verifique o log de eventos para o pod para ver se há entradas que são semelhantes ao conteúdo a seguir:

```
Events:
  Type     Reason     Age                    From                   Message
  ----     ------     ----                   ----                   -------
  **Warning  Unhealthy  2m29s (x23 over 135m)  kubelet, 9.1.1.1  Readiness probe errored: rpc error:
code = DeadlineExceeded desc = context deadline exceeded
  Warning  Unhealthy  2m13s (x23 over 135m)  kubelet, 9.1.1.1  Liveness probe errored: rpc error:
code = DeadlineExceeded desc = context deadline exceeded**
```

Para resolver esse problema, remova o pod com falha e permita que ele seja implementado novamente. Também é possível reiniciar o serviço do Docker no cluster.

O emissor ACME HTTP não pode emitir certificados nos clusters do OpenShift

O IBM Cloud Private Versão 3.2.0 não aplica as permissões necessárias à conta de serviço padrão para o serviço do gerenciador de certificados nos clusters do OpenShift. Essa limitação evita que o emissor ACME HTTP seja capaz de processar solicitações de desafio, o que evita que os certificados sejam emitidos a partir desse emissor.

A imagem do emissor ACME HTTP não é copiada para os nós do trabalhador

O emissor ACME HTTP é incluído no IBM Cloud Private Versão 3.2.0. É possível configurar o emissor ACME HTTP em seu cluster para criar certificados a partir de uma autoridade de certificação (CA) confiável. Esse recurso é opcional. Se você optar por configurar esse recurso em seu cluster, deverá concluir qualquer uma das seguintes etapas:

- Enviar por push manualmente a imagem do Docker a seguir para todos os nós do trabalhador em seu cluster:

  `ibmcom/icp-cert-manager-acmesolver:0.7.0`

  (OR)

- Criar um segredo de pull de imagem e associá-lo à conta de serviço padrão para o namespace no qual você está criando os certificados. Para obter mais informações, consulte Incluir ImagePullSecrets em uma conta de serviço.

A anotação de rewrite-target de ingresso NGINX falha ao fazer upgrade para o

IBM Cloud Private Versão 3.2.0

O IBM® Cloud Private Versão 3.2.0 usa o NGINX Ingress Controller Versão 0.23.0. A partir do NGINX Ingress Controller Versão 0.22.0, as definições de ingresso que usam a anotação `nginx.penetrs.kubernetes.io/rewrite-target` não são compatíveis com uma versão anterior. Para obter mais informações, consulte Regravar o destino.

Ao fazer upgrade para o IBM Cloud Private Versão 3.2.0, deve-se substituir a anotação `ingress.kubernetes.io/rewrite-target` pela parte de código a seguir:

```
ingress.kubernetes.io/use-regex: "true"
ingress.kubernetes.io/configuration-snippet: |
  rewrite "(?i)/old/(.*)" /new/$1 break;
  rewrite "(?i)/old$" /new/ break;
```

Função estabilizada, descontinuada e removida

As atualizações frequentes no IBM® Cloud Private e mudanças na tecnologia requerem que alguma função seja descontinuada ou removida do suporte.

Um item *estabilizado* não foi planejado para ser descontinuado ou removido de uma liberação subsequente de um produto, mas não é mais atualizado nem desenvolvido.

Um item *descontinuado* não está mais disponível para configurar novas instâncias, mas pode continuar sendo usado em instâncias existentes. Considere identificar uma maneira alternativa de realizar as tarefas que estão sendo feitas com uma função descontinuada. As funções descontinuadas podem ser removidas do produto.

Um item *removido* não pode ser usado com o IBM Cloud Private. A funcionalidade que estava disponível não está mais disponível. Muitas vezes, há uma maneira alternativa para realizar a mesma tarefa usando um método diferente.

As tabelas a seguir identificam itens ou funções que estão sendo eliminados ou que não são mais suportados no IBM Cloud Private.

Função descontinuada

Categoria	Item afetado	Status	Data de vigência ou versão	Ação recomendada	Informações adicionais (se disponíveis)

Categoria	Item afetado	Status	Data de vigência ou versão	Ação recomendada	Informações adicionais (se disponíveis)
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/users/<userId>/getHighestRole`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/highestRole`	Nada disponível
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/users/<userId>/getHighestRoleForCRN?crn=<resource crn>`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/highestRole?crn=<resource crn>`	Nada disponível
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/users/<userId>/getTeamResources`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/resources`	Nada disponível
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/k8resources/getK8Resources`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/resources?resourceType=namespace`	Nada disponível
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/users/<userId>/getTeams`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams`	Nenhum disponível.
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/users/<userId>/ldapList`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/directories`	Nada disponível
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/users/<userId>/getTeamRoleMappings`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/roleMappings`	Nada disponível

Categoria	Item afetado	Status	Data de vigência ou versão	Ação recomendada	Informações adicionais (se disponíveis)
APIs - `platform-identity-manager`	`GET /idmgmt/identity/api/v1/service/teamRoleBindings`	Reprovado	versão 3.2.0	Substituído por `GET /idmgmt/identity/api/v1/teams/roleMappings? userid= <userId>`	Nada disponível
APIs internas - `iam-token-service`	`POST /oidc/introspect` `POST /oidc/token` `GET /oidc/keys`	Reprovado	versão 3.2.0	Substituído por `POST /iam/oidc/introspect` `POST /iam/oidc/token` `GET /iam/oidc/keys`	As APIs descontinuadas e substituídas estão disponíveis na versão 3.2.0. A remoção das APIs descontinuadas está planejada para uma liberação subsequente.
Gerenciador de Certificados	ca.crt dentro do tls.cert	Reprovado	versão 3.2.0	O ca.crt era empacotado	
dentro do tls.cert, mas não é mais. O certificado de CA é fornecido no segredo do certificado como ca.crt	Nenhum disponível.				
Gerenciador de Certificados	ClusterIssuer padrão	Reprovado	versão 3.2.0	O ClusterIssuer padrão,	

icp-ca-issuer, que retém um certificado de CA autoassinado e um par de chaves, foi descontinuado. Crie seu próprio Emissor ou ClusterIssuer. | Criando seus próprios Emissores autoassinados e de CA. | Servidor API do Kubernetes | Parâmetro `experimental-encryption-provider-config` | Reprovado | versão 3.2.0 | Use o parâmetro `encryption-provider-config` com a versão `apiserver.config.k8s.io/v1` no lugar. | Gerenciando segredos e Criptografando segredos do Kubernetes com o plug-in do Key Management Service.| Log| A funcionalidade para instalar instâncias de criação de log adicionais com a segurança desativada| Reprovado | versão 3.2.0 | Atualize as implementações de criação de log não seguras para o nível 3.2.0 e use os parâmetros do gráfico para ativar a segurança. | Nada disponível | IBM Multicloud Manager | `mcm-inception` | Reprovado | versão 3.2.0 | `mcm-inception` é removido do gráfico do servidor IBM Multicloud Manager | Nada disponível | Portas | Porta 9443 do IAM | Reprovado | versão 3.1.2 |O acesso à porta do IAM foi descontinuado | Integração do IAM com o OpenShift |

Função removida

Tabela 2. Função removida

Categoria	Item afetado	Status	Data de vigência ou versão	Ação recomendada	Informações adicionais (se disponíveis)
Gráfico Helm	ibm-icplogging	Removido	8 de março de 2019	Use o serviço de criação de log interno.	Criação de log do IBM Cloud Private
Gráfico Helm	ibm-icpmonitoring	Removido	8 de março de 2019	Use o serviço de monitoramento interno.	Monitoramento do IBM Cloud Private

Categoria	Item afetado	Status	Data de vigência ou versão	Ação recomendada	Informações adicionais (se disponíveis)
Gráfico Helm	ibm-icplogging-kibana	Removido	8 de março de 2019	Use o serviço interno do Kibana.	Kibana
Gráfico Helm	Heapster	Removido	versão 3.2.0	Use o Servidor de Métricas interno	Nada disponível
Gráfico Helm	Unified-router	Removido	versão 3.2.0	Use o Servidor de Métricas interno	Nada disponível
IBM Cloud Private	`icp-router`	Removido	versão 3.2.0	`icp-router` é removido	Use `ingresso de gerenciamento`
IBM Multicloud Manager	CRD de conformidade	Removido	versão 3.2.0	O CRD de Conformidade foi removido.	Use o CRD de Políticas de Conformidade
IBM Multicloud Manager	Prometheus e MongoDB	Removido	versão 3.2.0	Prometheus e MongoDB são removidos do gráfico do servidor IBM Multicloud Manager	Nada disponível
Portas	Porta 9443 do IAM	Reprovado	versão 3.1.2	O acesso à porta do IAM foi descontinuado	Integração do IAM com o OpenShift

Informações Iniciais

O IBM Cloud Private é uma solução de classe corporativa e uma plataforma para desenvolvimento e gerenciamento de aplicativos conteinerizados locais.

O IBM Cloud Private suporta a opção de desenvolvimento de aplicativo com o Kubernetes, Cloud Foundry e modelos de programação baseados em função. O IBM Cloud Private facilita o desenvolvimento de aplicativos em um ambiente compartilhado, de vários locatários e suporta as arquiteturas Linux® x86_64, Linux® on Power® (ppc64le) e Linux® on IBM® Z and LinuxONE.

- Visão geral do IBM Cloud Private
- Pacotes configuráveis do IBM Cloud Private
- Gráficos do IBM Cloud Private
- IBM Cloud Private e DevOps
- Arquitetura
- Considerações sobre plataforma do IBM Cloud Private para preparação para o RGPD
- Considerações da plataforma IBM Cloud Private para conformidade com o FIPS
- Considerações de plataforma do IBM Cloud Private para preparação de PCI
- Linguagens suportadas

Visão geral do IBM Cloud Private

O IBM® Cloud Private é uma solução de classe corporativa de última geração, predefinida e uma plataforma para desenvolver e gerenciar aplicativos conteinerizados. Ele é um ambiente integrado para gerenciar contêineres que inclui o orquestrador de contêiner Kubernetes, um registro de imagem privado, uma console de gerenciamento e as estruturas de monitoramento.

- Por que utilizar o IBM Cloud Private?
- Componentes de software livre
- Recursos e benefícios chave

Por que usar o IBM Cloud Private?

O IBM Cloud Private entrega uma solução de contêiner gerenciado pelo cliente para empresas. Ele também está disponível em uma edição de comunidade, IBM® Cloud Private-CE, que fornece uma oferta limitada que está disponível gratuitamente e é ideal para ambientes de teste.

As empresas usam a plataforma IBM Cloud Private para os seguintes casos de uso:

- Desenvolver e executar aplicativos nativos em nuvem de produção em uma nuvem privada
- Integrar e usar com segurança dados e serviços de origens externas para a nuvem privada

- Refatorar e modernizar aplicativos corporativos legados na nuvem

O IBM Cloud Private suporta a opção no desenvolvimento de aplicativo com o Kubernetes, Cloud Foundry e modelos de programação baseados em função. O IBM Cloud Private é uma plataforma de nuvem dos tipos contêiner privado como um serviço (CaaS), plataforma como um serviço (PaaS) e infraestrutura como um serviço (IaaS).

O IBM Cloud Private é diferenciado fornecendo serviços de aplicativo de produção, tempos de execução de aplicativo, serviços de dados e analítica, serviços de sistema de mensagens, serviços de armazenamento em cache, e outros, que são necessários para os desenvolvedores inovarem de forma rápida e iterativa com base em suas necessidades de negócios.

Componentes de software livre

Para a melhor experiência no uso do IBM Cloud Private, deve-se entender como o Kubernetes, o Docker e o Helm funcionam. Esses componentes de software livre são fundamentais para a plataforma do IBM Cloud Private. Você usa as implementações do Kubernetes para colocar instâncias de aplicativos, que são construídas nos gráficos Helm que fazem referência às imagens do Docker. Os gráficos Helm contêm os detalhes sobre seu aplicativo e as imagens do Docker contêm todos os pacotes de software que seus aplicativos precisam executar. É possível aprender mais sobre esses componentes na documentação para cada componente:

- Documentação do Helm
- Introdução, Parte 1: Orientação e configuração na documentação do Docker
- Informações básicas do Kubernetes na documentação do Kubernetes

Recursos e benefícios-chave

O IBM Cloud Private versão 3.2.0 possui os recursos e funções chave a seguir:

- Um instalador unificado
- Monitoramento e criação de log robustos com pilha ELK
- Monitoramento e alertas
- Medição
- Identidade e acesso
- Segurança
- IBM Consultor de Vulnerabilidade
- IBMCloud Automation Manager
- IBM Cloud Transformation Advisor
- IBM Microclimate
- IBM Cloud Private console de gerenciamento
- Kubernetes
- Registro de imagem de Docker privado
- Helm
- Catalog
- Serviço do Kubernetes Catalog para gerenciar brokers de serviço

Um instalador unificado

Configure rapidamente um cluster baseado em Kubernetes que contenha nós principais, do trabalhador, de proxy e opcionais, de gerenciamento e do Consultor de Vulnerabilidade, usando um instalador baseado em Ansible. Esse instalador baseado em Ansible é rápido e simples de usar. Execute alguns comandos simples em um único nó de inicialização e seu cluster estará funcionando em alguns minutos.

Monitoramento e criação de log robustos com pilha ELK

Cada contêiner produz logs. Os logs são críticos para a depuração e post-mortem em falhas de produção. Os aplicativos de doze fatores se dividem em muitos microsserviços, o que aumenta o número de logs entre os contêineres que você precisa depurar. Além disso, muitos logs são gravados em arquivos dentro do contêiner. O IBM Cloud Private usa a pilha ELK (Elasticsearch, Logstash, Kibana) e o Filebeat. Esse processo de monitoramento e criação de log fornece um armazenamento centralizado para todos os logs e métricas, melhor desempenho e maior estabilidade quando você acessa e consulta logs e métricas.

Também é possível instalar o Kibana ou o Grafana para consultar os dados no banco de dados Elasticsearch. É possível usar os resultados dessas consultas para produzir gráficos e relatórios intuitivos.

Monitoramento e alertas

Cada contêiner deve ter seu funcionamento monitorado. As análises básicas de vivacidade no Kubernetes asseguram que os pods com falha sejam reiniciados. No entanto, esse monitoramento é apenas o início de seu desafio de monitoramento em uma plataforma conteinerizada.

Cada contêiner de aplicativo em cada contêiner de middleware produz métricas de funcionamento. O IBM Cloud Private configura coletores do Prometheus customizados para métricas customizadas. As métricas customizadas ajudam a fornecer insights e construir blocos para alertas e painéis customizados do cliente. O IBM® Cloud Private usa uma pilha do Prometheus e do Grafana para monitoramento do sistema.

Medição

Cada contêiner deve ser gerenciado para uso sob licença. É possível usar o serviço de medição para visualizar e fazer download de métricas de uso detalhadas para seus aplicativos e cluster. As medições precisas são visíveis por meio da IU de medição e os dados são mantidos por até três meses. Os relatórios de resumo mensais também estão disponíveis para download e são mantidos por até 24 meses.

Identidade e acesso

O gerenciamento de identidade e de acesso assegura uma identidade consistente em todos os serviços da plataforma. O IBM Cloud Private apresenta o conceito de equipes sobre as funções/funções de cluster brutas do Kubernetes. As equipes ligam uma coleção de recursos, tanto dentro quanto fora do Kubernetes, a um conjunto de usuários com funções definidas. O modelo de equipe é baseado no modelo de controle de acesso do IBM UrbanCode Deploy.

Segurança

O IBM Cloud Private garante a segurança dos dados em trânsito e dos dados em repouso para todos os serviços da plataforma. Todos os serviços expõem os terminais de rede por meio de TLS e armazenam dados criptografados inativos. Todos os serviços devem fornecer logs de auditoria para ações que são executadas, quando elas foram executadas, e quem executou a ação. O modelo de segurança assegura trilhas de auditoria consistentes para todos os serviços de plataforma e conformidade em todo o middleware.

IBMConsultor de Vulnerabilidade

Os contêineres estão constantemente em mudança. Vulnerabilidades devem ser identificadas em uma base contínua. Os principais benefícios do Vulnerability Advisor incluem:

- Varredura de imagem para identificação de riscos de segurança
- Identificação de violações de política
- Determinação de melhorias de melhor prática
- Execução de ações corretivas

IBMCloud Automation Manager

Os contêineres são tudo, no entanto, nem tudo está em um contêiner. O IBM Cloud Automation Manager (CAM) é uma plataforma de gerenciamento de autoatendimento de múltiplas nuvens em execução no IBM® Cloud Private, que permite que os desenvolvedores e administradores atendam às demandas de negócios com mais potencialidade. Essa plataforma permite que você gerencie e entregue de modo eficiente serviços por meio de automação de ponta a ponta, enquanto permite que os desenvolvedores construam aplicativos alinhados com políticas corporativas.

IBM Cloud Transformation Advisor

A maioria dos aplicativos atuais não estão em contêineres e os clientes precisam de ajuda com a modernização de cargas de trabalho. O IBM Cloud Transformation Advisor permite insights sobre aplicativos existentes. O Transformation Advisor é uma ferramenta que usa informações sobre seu ambiente e aplicativos do WebSphere. Essas entradas são combinadas com regras e insights adquiridos de anos de trabalho com aplicativos IBM WebSphere e IBM WebSphere para fornecer recomendações para sua jornada de nuvem.

Benefícios:

- Incluído e implementado no IBM® Cloud Private
- Realiza introspecção de implementações existentes do IBM WebSphere
- Fornece recomendações para modernização de aplicativo

IBMMicroclimate

Transformando ideias inovadoras em valor de negócios entregues por meio de contêineres. O IBM Microclimate permite uma criação rápida de novos aplicativos. O Microclimate é um ambiente de desenvolvimento de ponta a ponta que pode ser usado para criar, editar e implementar aplicativos rapidamente. Os aplicativos são executados em contêineres desde o primeiro dia e podem ser entregues em produção no Kubernetes por meio de um pipeline de DevOps automatizado usando o Jenkins. O Microclimate pode ser instalado localmente ou no IBM® Cloud Private.

IBM Cloud Private console de gerenciamento

Gerencie, monitore e solucione problemas de seus aplicativos e cluster por meio de uma única console de gerenciamento centralizada e segura.

Kubernetes

Para executar um contêiner na produção, o Kubernetes traz as primitivas de orquestração para suportar diferentes estilos de cargas de trabalho:

- `ReplicaSets` stateless
- `StatefulSets` stateful
- `Jobs` em lote
- `DaemonSets` do sistema

Registro de imagem de Docker privado

O registro do Docker privado integra-se à API de registro do Docker V2 para fornecer um serviço de registro local que funciona da mesma maneira que o serviço de registro baseado em nuvem, Docker Hub. Esse registro local tem todos os mesmos recursos que o Docker Hub, mas também é possível restringir quais usuários podem visualizar ou obter imagens desse registro.

Helm

Helm, o sistema de gerenciamento de pacote nativo do Kubernetes, é usado para gerenciamento de aplicativos dentro de um cluster do IBM Cloud Private. A comunidade GitHub do Helm administra e expande continuamente um conjunto de aplicativos Kubernetes testados e pré-configurados. É possível incluir itens desse catálogo de aplicativos estáveis em seu cluster do console de gerenciamento. A instalação desse catálogo da comunidade do Helm fornece mais de 80 aplicativos Kubernetes extras que estão prontos para implementação em seu cluster. Para visualizar uma lista de todos os aplicativos estáveis que estão disponíveis no repositório do Helm, consulte Gráficos estáveis do Helm.

Os gráficos Helm descrevem até mesmo os aplicativos mais complexos, fornecem instalação de aplicativo repetida e servem como um ponto único de autoridade. Os gráficos Helm são fáceis de atualizar com upgrades e ganchos customizados no local. Também é fácil criar uma versão, compartilhar e hospedar os gráficos em servidores públicos ou privados. É possível usar o `helm rolback` para retroceder para uma versão mais antiga de uma liberação com facilidade.

Catalog

O IBM Cloud Private fornece um Catalog de conteúdo IBM e de terceiro fácil de usar, estender e editar. A seguir estão alguns conceitos principais:

- Gráficos: um pacote configurável de recursos do Kubernetes
- Repositório: uma coleção de gráficos
- Liberações: uma instância de gráfico carregada no Kubernetes. O mesmo gráfico pode ser implementado várias vezes e cada vez se torna sua própria liberação

O Catalog fornece um local centralizado do qual é possível procurar e instalar pacotes em seu cluster.

Os pacotes para produtos IBM adicionais estão disponíveis em repositórios curados que são incluídos na lista de repositórios padrão do IBM Cloud Private. Seu ambiente deve estar conectado à Internet para que você acesse os gráficos para esses pacotes. Para visualizar uma lista de todos os gráficos do IBM Cloud Private, consulte Gráficos estáveis da IBM.

Kubernetes Service Catalog para gerenciar brokers de serviço

O IBM Cloud Private suporta o serviço do Kubernetes Catalog. É possível configurar os aplicativos do broker de serviço para gerenciar os recursos e detalhes do Service Catalog.

O componente Service Catalog inclui os seguintes recursos do Kubernetes:

- ClusterServiceBrokers

- ClusterServiceClasses
- ClusterServicePlans
- ServiceInstances
- ServiceBindings

O broker de serviço é um componente que implementa a API do broker de serviço para visualizar os serviços e planos disponíveis, criar uma instância a partir de serviços e planos disponíveis e criar ligações para conectar-se à instância de serviço. Para obter mais informações, consulte Serviço Catalog .

Pacotes Configuráveis do IBM Cloud Private

O IBM Cloud Private está disponível para compra com outros produtos IBM, incluindo middleware IBM e outros produtos de software.

Todos os pacotes configuráveis do IBM Cloud Private contêm a plataforma principal do IBM Cloud Private e os apps de destaque disponíveis gratuitamente, que é possível acessar por meio do Catalog. Cada pacote configurável contém um software autorizado diferente que pode ser instalado separadamente a partir do Passport Advantage ou incluído no Catalog após a instalação da plataforma IBM Cloud Private.

IBM Cloud Private Cloud Native

Tabela 1. Componentes do IBM Cloud Private Cloud Native

Software	Descrição	Método de instalação
Cloud Automation Manager	O IBM Cloud Automation Manager é uma solução de gerenciamento de nuvem no IBM Cloud Private para implementar a infraestrutura de nuvem em múltiplas nuvens com uma experiência do usuário otimizada.	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog
IBM WebSphere Application Server Liberty	O IBM WebSphere Application Server Liberty combinado com uma rica paleta de tecnologias de middleware disponíveis dentro do Catalog de conteúdo do IBM Cloud Private reduz a quantia de integração necessária para integrar middleware, como bancos de dados, soluções de armazenamento em cache, soluções de sistema de mensagens.	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog
Microclimate	O Microclimate oferece uma experiência de usuário completa para o desenvolvimento e a implementação de aplicativos modernos e acessível por padrão no IBM Cloud Private. O Microclimate oferece uma experiência de primeira classe para a implementação e o gerenciamento de aplicativos em contêiner.	• Instalação separada não IBM • Inclua no IBM Cloud Private Catalog

IBM Cloud Private Enterprise

Software	Descrição	Método de instalação

Software	Descrição	Método de instalação
Cloud Automation Manager	O IBM Cloud Automation Manager é uma solução de gerenciamento de nuvem no IBM Cloud Private para implementar a infraestrutura de nuvem em múltiplas nuvens com uma experiência do usuário otimizada.	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog
IBM API Connect Professional	O IBM API Connect Professional fornece uma solução abrangente para gerenciar o ciclo de vida inteiro da API, da criação ao gerenciamento.	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog
IBM MQ Advanced	O IBM MQ Advanced é uma solução de middleware de sistema de mensagens robusta que fornece sistema de mensagens simples, seguro, escalável e confiável para conectar e integrar aplicativos,	

sistemas e serviços dentro e entre várias plataformas, incluindo ambientes no local e implementações na nuvem.|

- Pacote de instalação separado que está disponível no Passport Advantage
- Inclua no IBM Cloud Private Catalog

| | IBM WebSphere Application Server Liberty|O IBM WebSphere Application Server Liberty combinado com uma rica paleta de tecnologias de middleware disponíveis dentro do Catalog de conteúdo do IBM Cloud Private reduz a quantia de integração necessária para integrar middleware, como bancos de dados, soluções de armazenamento em cache, soluções de sistema de mensagens.|

- Pacote de instalação separado que está disponível no Passport Advantage
- Inclua no IBM Cloud Private Catalog

|| | IBM WebSphere Application Server Network Deployment|IBM WebSphere Application Server Network Deployment fornece um ambiente de tempo de execução avançado e flexível para implementações de aplicativos de larga escala. Ele oferece disponibilidade quase contínua com desempenho avançado e recursos de gerenciamento para aplicativos missões críticas.|

- Pacote de instalação separado que está disponível no Passport Advantage
- Inclua no IBM Cloud Private Catalog

|| | Microclima |O Microclimate oferece uma experiência de usuário completa para o desenvolvimento e a implementação de aplicativos modernos e acessível por padrão no IBM Cloud Private. O Microclimate oferece uma experiência de primeira classe para a implementação e o gerenciamento de aplicativos em contêiner.|

- Instalação separada não IBM
- Inclua no IBM Cloud Private Catalog

|

Componentes opcionais

Também é possível incluir os componentes a seguir em seu pacote configurável. Entre em contato com seu representante de vendas do IBM

Tabela 3. Componentes opcionais

Software	Descrição	Método de instalação
IBM® Cloud Private Cloud Foundry	A plataforma IBM® Cloud Private Cloud Foundry traz o mesmo nível de recursos do aplicativo Cloud Foundry que você experimenta no IBM Cloud e no IBM Cloud Dedicated para seu data center. Ele está disponível para ser executado no VMware vSphere.	Pacote de instalação separado que está disponível no Passport Advantage

Software	Descrição	Método de instalação
Data Science Experience Local	O IBM Data Science Experience (DSX) Local é uma solução corporativa em instalações para cientistas de dados e engenheiros de dados. Ele oferece um conjunto de ferramentas de ciências de dados que são integradas com tecnologias proprietárias IBM.	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog
IBM DB2 Direct Advanced	O IBM DB2 Direct Advanced é a plataforma de dados de transformação para cargas de trabalho transacionais e analíticas na era digital, de nuvem e cognitiva. Ele fornece disponibilidade contínua de dados para manter operações transacionais e analíticas com o máximo de eficiência para assegurar que o acesso aos dados não seja impactado por qualquer tempo de inatividade planejado ou não planejado	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog • Usar o Cloud Automation Manager
IBM UrbanCode Deploy	O IBM UrbanCode Deploy é uma solução de automação de liberação de aplicativos que combina forte visibilidade, rastreabilidade e recursos de auditoria. Para obter mais informações sobre como usar o IBM UrbanCode Deploy com contêineres, consulte Introdução ao IBM UrbanCode Deploy e contêineres.	• Pacote de instalação separado que está disponível no Passport Advantage • Inclua no IBM Cloud Private Catalog

IBM Certified Containers e o IBM Cloud Paks

Os IBM Certified Containers e os IBM Cloud Paks são softwares implementáveis que estão disponíveis no IBM® Cloud Private Catalog. Os IBM Certified Containers e os IBM Cloud Paks atendem a critérios específicos para empacotamento e implementação de software conteinerizado. Eles são construídos com padrões abertos e são integrados aos serviços de plataforma para operações de gerenciamento e ciclo de vida.

IBM Certified Containers

Os IBM Certified Containers são mais do que um gráfico do Helm simples. Os IBM Certified Containers aceleram o time-to-value e melhoram a prontidão corporativa a um custo menor que os contêineres em si.

Os IBM Certified Containers podem ser criados pelo IBM ou ser uma solução de software de terceiro que é oferecida por Parceiros do IBM. Consulte Suporte do IBM para saber como o IBM suporta contêineres.

Um IBM Certified Container é um IBM Certified Container Certificado ou um IBM Certified Container. Um IBM Certified Container Certificado atende aos requisitos adicionais para soluções de software conteinerizadas de classificação corporativa.

Tabela 1. Recursos de um IBM Certified Container

Área	Recurso
Velocidade	
	Implementações pré-configuradas baseadas no conhecimento do produto
Manutenção	
	Upgrades contínuos
	Recuperações
	Testes de segurança/vulnerabilidade
	Integrações com serviços de gerenciamento para criação de log, monitoramento, medição e segurança fornecem controle e gerenciamento de cargas de trabalho de produção

Para obter mais informações sobre os IBM Certified Containers, consulte Identificando o IBM Certified Containers.

IBM Cloud Paks

Em alguns casos, uma única oferta do IBM Certified Container não pode fornecer toda a funcionalidade que é necessária para resolver requisitos complexos. Talvez você queira ou precise de várias ofertas que geralmente são implementadas juntas para atenderem a um requisito. Algumas dessas ofertas que são comumente instaladas juntas são fornecidas como um pacote chamado de IBM Cloud Pak.

Cada IBM Cloud Pak contém os componentes a seguir:

Componente	Descrição
Gráfico do Helm e imagem do Docker do Solution Cloud Pak	
	O Solution Cloud Pak fornece uma interface para instalar e manter as ofertas em um

IBM Cloud Pak. Após a implementação do gráfico do Helm e da imagem do Docker do Solution Cloud Pak, é possível usar o painel do Solution Cloud Pak para instalar o IBM Certified Container individual que está contido no IBM Cloud Pak. | | IBM Cloud Pak | | | | Pelo menos um IBM Certified Container está incluído no pacote. Cada um inclui sua própria imagem do gráfico do Helm e do Docker que fornece as informações que são necessárias para sua implementação em seu ambiente do IBM Cloud Private. |

As ofertas em um IBM Cloud Pak são implementadas com um único gráfico do Helm, portanto, elas têm uma experiência de implementação comum. É possível instalar as ofertas que são fornecidas com o IBM Cloud Pak individualmente usando o painel do Solution Cloud Pak.

Consulte Identificando o IBM Cloud Paks para obter mais informações sobre o IBM Cloud Paks.

IBM Cloud Private gráficos

Nota: alguns desses gráficos não são distribuídos com o IBM Cloud Private e são licenciados sob termos e condições separados. É possível fazer o download de gráficos adicionais a partir do repositório IBM/charts para uso com o IBM® Cloud Private. Para obter mais informações sobre um gráfico específico, consulte o arquivo `readme.md` para o gráfico, que está no repositório de `gráficos`.

- Serviços principais
- Abrir Software de Origem
- Software IBM Enterprise

Serviços principais

Para obter mais informações sobre esses serviços principais, consulte Componentes.

Para revisar os tópicos de visão geral para esses componentes principais, consulte Serviços principais. Para obter mais informações sobre esses serviços, também é possível ver o arquivo `readme.md` para o gráfico, que está no repositório de `gráficos`.

Tabela 1. Gráficos de Helm de serviços principais

Serviços	Componentes
Aplicativo	
	Prometheus
Criação de Log	
	ELK (Elasticsearch, Logstash, Kibana)
Medição	
	Insights do Produto
Segurança	
	Identidade e acesso
	Gerenciamento
	Consultor de Vulnerabilidade

Software livre

Tabela 2. Gráficos Helm de software livre

Serviço	Software
Cadeias de ferramentas e tempos de execução	
	Jenkins
	Abrir Liberty
	Tempo de execução do Swi
	Tempo de execução do Node.js
Sistema de mensagens	
	RabbitMQ
Serviços de Dados	
	MongoDB
	PostgreSQL
	Redis
Serviços de Caching	
	Hazelcast IMDG
servidores HTTP	
	Nginx
Ferramentas	
	Terminal da Web
	Skydive

Software Corporativo IBM

Um conjunto atualizado com frequência de software corporativo IBM também está disponível como gráficos Helm no catálogo do IBM Cloud Private. Para obter informações adicionais sobre esses serviços, consulte Serviços apresentados e o arquivo leia-me para os gráficos.

IBM e DevOps

O IBM fornece uma experiência integrada do DevOps para todos os aplicativos.

O DevOps é uma abordagem para a entrega de software simples e ágil que promove uma colaboração mais estreita entre as operações de linhas de negócios, desenvolvimento e TI. Historicamente, o desenvolvimento e as operações, e até mesmo os testes, eram operações isoladas. O DevOps faz com que eles se unam para melhorar e reduzir o tempo necessário para atender ao feedback do cliente. Com entrega, implementação e monitoramento contínuos de aplicativos, as empresas podem:

- Responder ao mercado e criar experiências do usuário envolventes com mais rapidez.
- Implementar continuamente o software em ambientes de desenvolvimento, de teste e de produção
- Escalar o DevOps com sucesso sem interromper os negócios
- Conduzir a consistência do ambiente desde o teste até a produção e também em nuvens públicas e privadas
- Construir uma cultura de startup que reúna negócios, desenvolvimento e operações
- Facilitar a melhora da qualidade e da estabilidade do aplicativo por meio de liberações frequentes

- Ajudar a reduzir os custos por meio de uma melhor eficiência e redução de indisponibilidades

Para aperfeiçoar a implementação entre nuvens públicas e privadas e ambientes e mainframes virtualizados, o IBM UrbanCode Deploy, um complemento para o IBM, ajuda a automatizar a implementação de mudanças em contêineres, aplicativos, middleware e banco de dados em ambientes de desenvolvimento, de teste e de produção.

Também é possível instalar o IBM Microclimate no IBM. O conjunto do IBM UrbanCode é um líder de mercado para soluções de automação de implementação e liberação. O conjunto do IBM UrbanCode é projetado para facilitar feedback rápido e entrega contínua em ambientes de desenvolvimento ágeis ao fornecer trilhas de auditoria, controle de versões e aprovações necessários na produção. Usando o IBM UrbanCode Deploy, o Release e o Build, as empresas conseguem reduções de custo rápidas e duradouras, além de retorno de investimento. O IBM Urbancode Deploy orquestra e automatiza a implementação de aplicativos, bancos de dados e configurações em ambientes de desenvolvimento, teste e produção.

O IBM Microclimate é um ambiente de desenvolvimento de ponta a ponta no qual é possível criar, editar e implementar aplicativos rapidamente. Os aplicativos são executados em contêineres desde o primeiro dia e podem ser entregues em produção no Kubernetes por meio de um pipeline de DevOps automatizado usando o Jenkins. O Microclimate pode ser instalado localmente ou no IBM. Para obter mais informações, consulte Microclimate.

Jenkins é um gráfico Helm de software livre que é uma opção de instalação opcional. Ele estende o instalador do Ansible para fornecer uma opção em vez de percorrer o catálogo depois que o IBM é instalado.

Arquitetura

Um cluster do IBM® Cloud Private possui quatro classes principais de nós: inicialização, principal, trabalhador e proxy.

É possível especificar opcionalmente os nós de gerenciamento, Vulnerability Advisor (VA) e etcd em seu cluster.

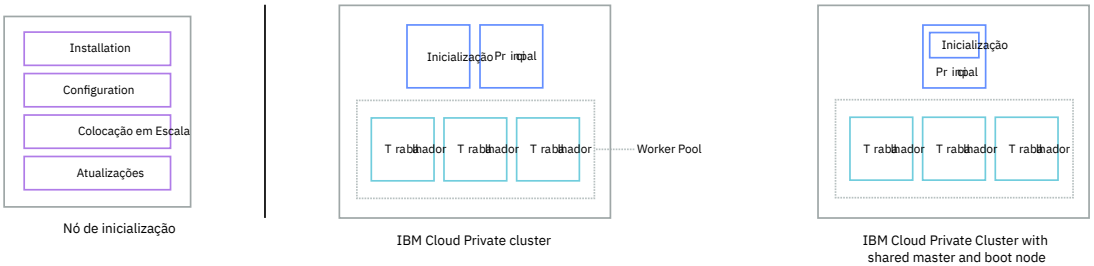
Você determina a arquitetura de seu cluster do IBM Cloud Private antes de instalá-lo. Após a instalação, é possível incluir ou remover apenas nós do trabalhador, de proxy, de gerenciamento e do VA de seu cluster. Não é possível converter um cluster padrão em um cluster de alta disponibilidade ou incluir mais nós principais em um cluster de alta disponibilidade.

Nota: Nas imagens a seguir, os clusters representam configurações mínimas do IBM Cloud Private. As configurações de produção reais podem variar.

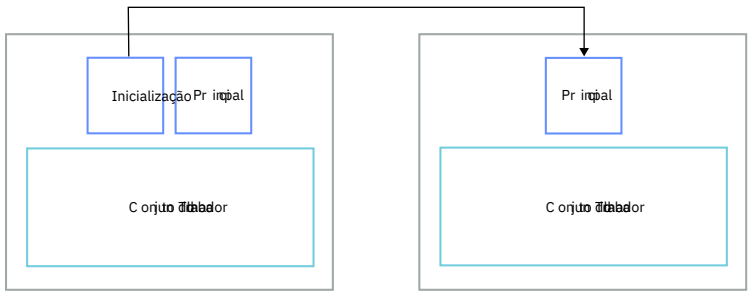
- Nó de inicialização
- Nó principal
- Nó do trabalhador
- Nó do proxy
- Nó de gerenciamento
- Nó do VA
- nó etcd
- Arquiteturas de cluster

Nó de inicialização

Nó de inicialização do



Um nó de inicialização ou de autoinicialização é usado para executar a instalação, a configuração, o ajuste de escala de nó e as atualizações de cluster. Somente um nó de inicialização é necessário para um cluster. É possível usar um único nó para o principal e de inicialização.

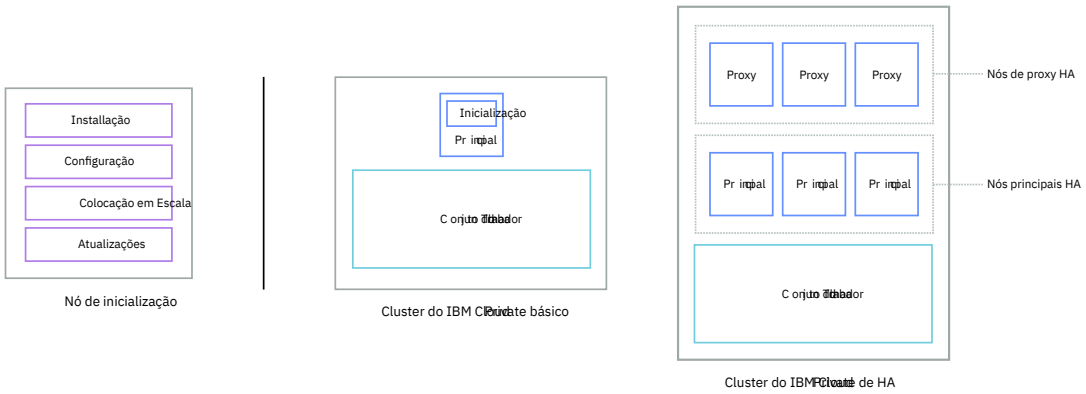


Nó de inicialização único do

É possível usar um nó único de inicialização para vários clusters. Nesse caso, o de inicialização e o principal não podem estar em um único nó. Cada cluster deve ter seu nó principal. No nó de inicialização, deve-se ter um diretório de instalação separado para cada cluster. Se você estiver fornecendo sua própria autoridade de certificação (CA) para autenticação, será necessário ter um domínio de CA separado para cada cluster.

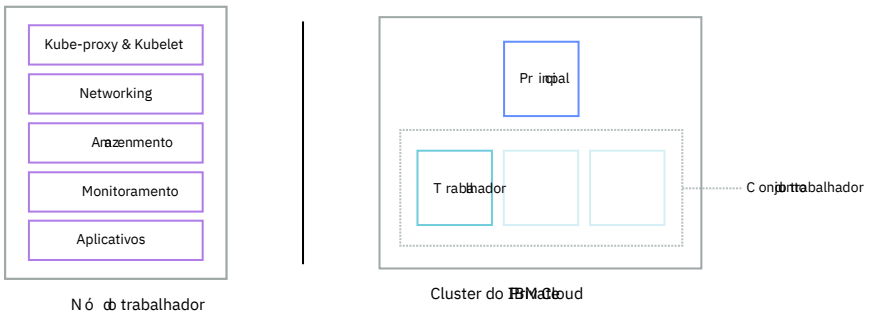
Nó principal

Nó principal do



Um nó principal fornece serviços de gerenciamento e controla os nós do trabalhador em um cluster. Os nós principais hospedam processos que são responsáveis pela alocação de recursos, manutenção de estado, planejamento e monitoramento. Como um ambiente de alta disponibilidade (HA) contém vários nós principais, se o nó principal inicial falhar, a lógica de failover promoverá automaticamente um nó diferente para a função principal. Os hosts que podem agir como mestres são chamados de candidatos mestres.

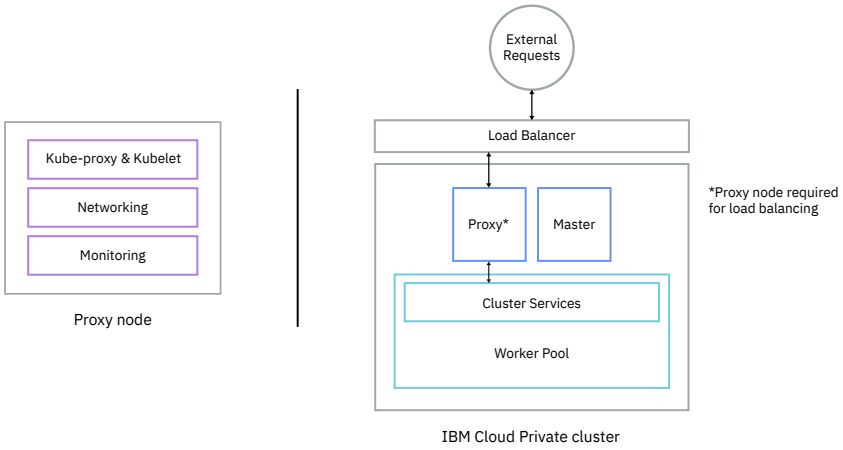
Nó do trabalhador



Nó do trabalhador do

Um nó do trabalhador é um nó que fornece um ambiente conteinerizado para a execução de tarefas. Com o aumento das demandas, é possível incluir facilmente mais nós do trabalhador no cluster para melhorar o desempenho e a eficiência. Um cluster pode conter qualquer número de nós do trabalhador, mas é necessário que haja um mínimo de nós do trabalhador.

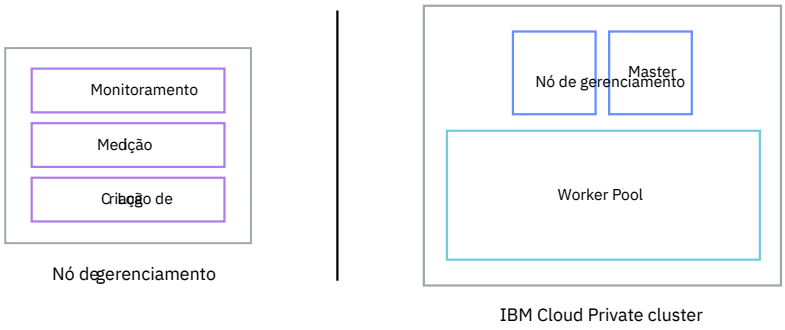
Nó do proxy



Nó de proxy do

Um nó do proxy é um nó que transmite uma solicitação externa para os serviços criados dentro do cluster. Como um ambiente de alta disponibilidade (HA) contém vários nós do proxy, se o nó do proxy principal falhar, a lógica de failover promoverá automaticamente um nó diferente para a função do proxy. Embora você possa usar um único nó como principal e de proxy, é melhor usar nós do proxy dedicados para reduzir a carga no nó principal. Um cluster deve conter pelo menos um nó do proxy caso seja necessário fazer o balanceamento de carga dentro do cluster.

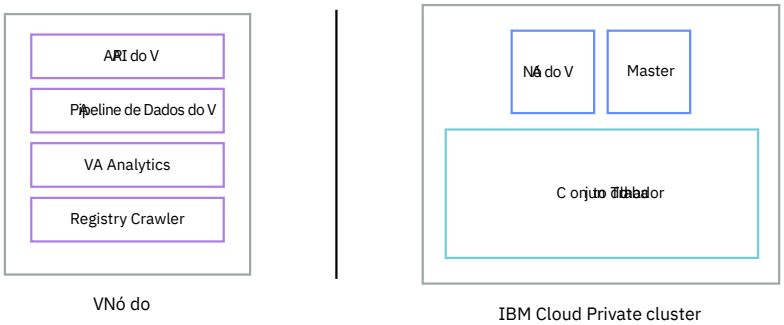
Nó de gerenciamento



Nó de gerenciamento do

Um nó de gerenciamento é um nó opcional que apenas hospeda serviços de gerenciamento como monitoramento, medição e criação de log. Ao configurar nós de gerenciamento dedicados, é possível evitar que o nó principal se torne sobrecarregado. É possível ativar o nó de gerenciamento apenas durante a instalação do IBM Cloud Private.

Nó do VA



VNó do

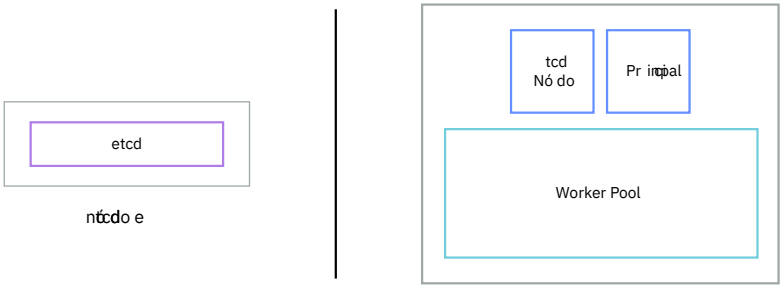
IBM Cloud Private cluster

Nó do VA do

Um nó do VA (Consultor de Vulnerabilidade) é um nó opcional que é usado para executar os serviços do Consultor de Vulnerabilidade. Os serviços do Consultor de Vulnerabilidade são intensivos em recurso. Se você usar o serviço do Consultor de

Vulnerabilidade, especifique um nó do VA dedicado. Para obter mais informações sobre o Consultor de vulnerabilidade, consulte Consultor de Vulnerabilidade.

nó etcd



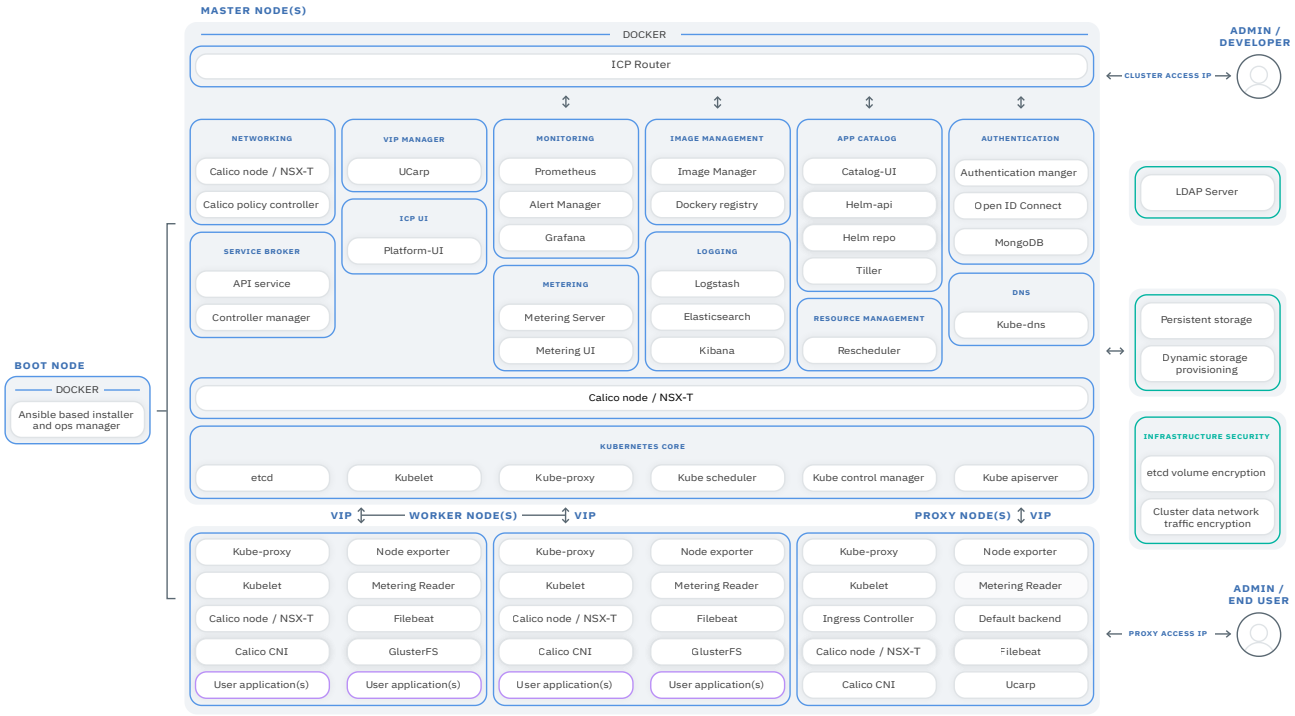
IBM Cloud Private cluster

Nó etcd do

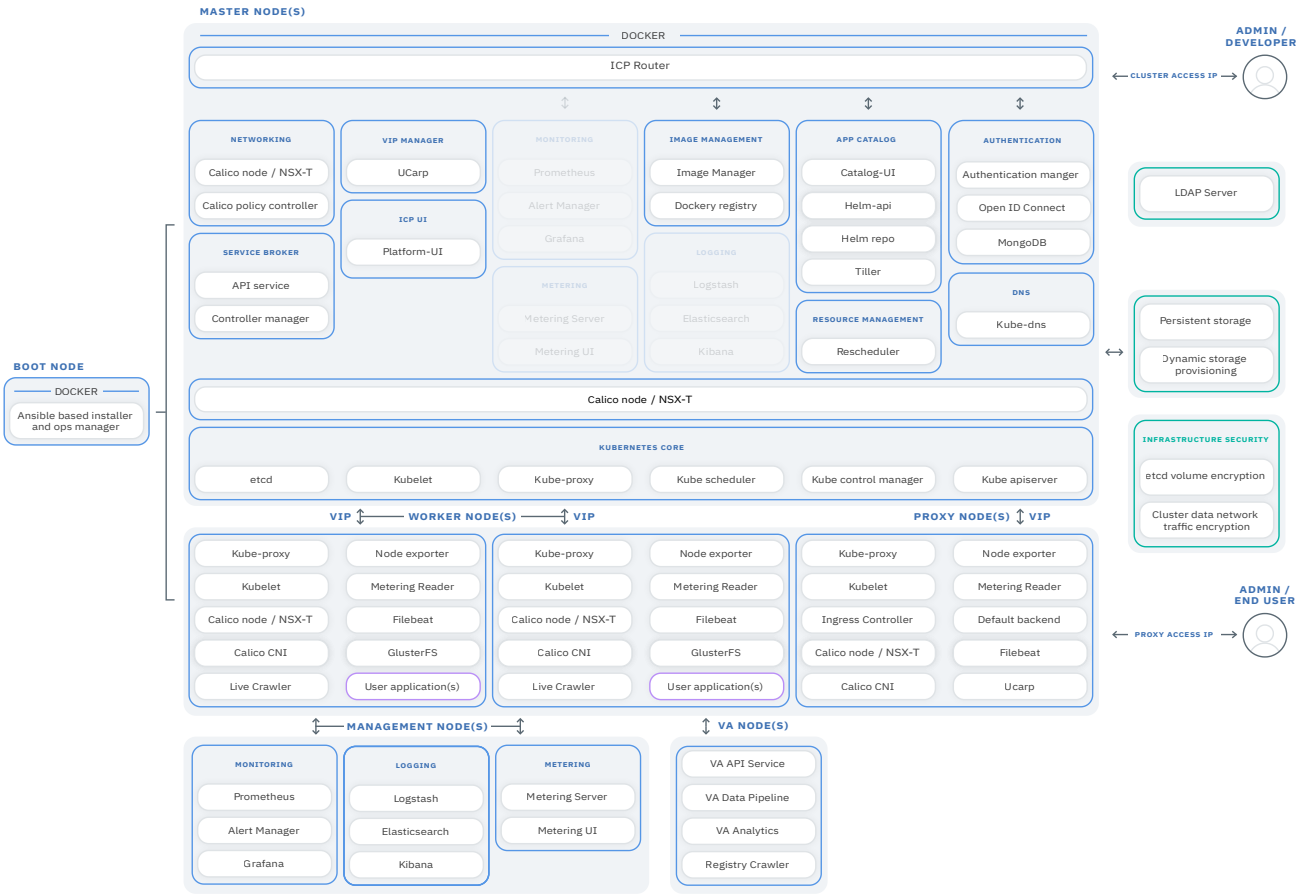
Um nó etcd é um nó opcional que é usado para executar o armazenamento de valor da chave distribuída `etcd`. Configurar um nó etcd em um cluster do IBM Cloud Private que possui muitos nós, como 100 ou mais, ajuda a melhorar o desempenho do `etcd`. Para obter mais informações sobre como configurar um nó etcd, consulte Configurando as funções do nó no arquivo host.

Arquitetura

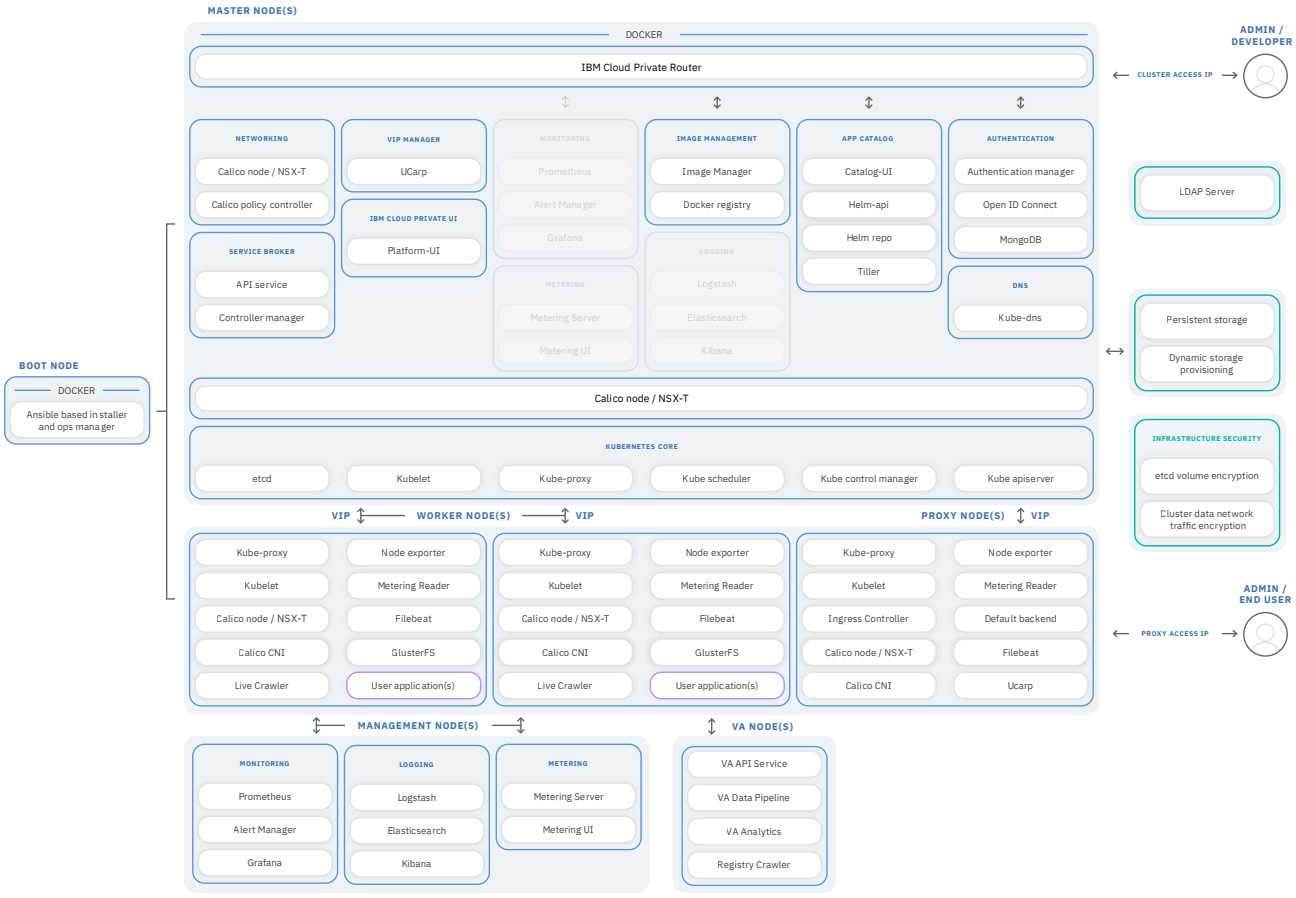
Se você usar nós do proxy em seu cluster, a arquitetura será semelhante ao diagrama a seguir:



Se você usar nós de gerenciamento em seu cluster, a arquitetura assemelha-se ao diagrama a seguir:



Se você usar nós do VA em seu cluster, a arquitetura será semelhante ao diagrama a seguir:



Componentes IBM® Cloud Private

O IBM Cloud Private possui dois componentes principais: um gerenciador de contêiner (Docker) e um orquestrador de contêiner (Kubernetes).

Outros componentes de um cluster do IBM Cloud Private trabalham com os componentes principais para fornecer serviços, como autenticação, armazenamento, rede, criação de log e monitoramento. Uma console de gerenciamento de cluster também é fornecida, que serve como um local de gerenciamento centralizado para os serviços.

Para obter mais informações sobre modelos de arquitetura e tipos de nós, consulte Arquitetura.

Nota: componentes de gerenciamento, como monitoramento, medição e criação de log, são executados no nó de gerenciamento. Se nenhum nó de gerenciamento está presente em seu cluster, os componentes de gerenciamento são executados no nó principal.

- Componentes
- Serviços e dependências de componentes
- Componentes do Vulnerability Advisor (VA) (recurso opcional)

Componentes

Visualize a tabela a seguir para obter uma descrição dos componentes do nó do IBM Cloud Private.

Componente	Versão	Função
Gerenciador de alertas	0.15.0	Manipula alertas enviados pelo servidor Prometheus. Ele envia dados de

deduplicação, agrupamento e roteamento de alerta para a integração do receptor correta, como slack, email ou PagerDuty.| |Ansible basedauth-apikeys, installer e gerenciador de operações|2.5.0|Implementa o IBM Cloud Private nos nós principal e do trabalhador. O nó de inicialização também é usado para escalar o tamanho do cluster on demand e para atualizações de rolagem.| |Criação de log da auditoria|3.2.0|Encaminha logs de auditoria gerados pelo servidor de API e serviços de plataforma do Kubernetes para os servidores Elasticsearch e SIEM.| |Gerenciador de autenticação|3.2.0|Fornece uma API HTTP para gerenciar usuários. Os protocolos são implementados em um modo RESTful. O OpenID Connect é usado para autenticação.| |Calico

(nó)|3.5.2|Define as configurações de rede Calico em cada nó. Para obter informações adicionais sobre os componentes do Calico, consulte v3.5.2 .| |calicoctl|3.5.2|Uma ferramenta do cliente que é usada para criar, ler, atualizar e excluir objetos do Calico a partir da linha de comandos.| |Calico (CNI)|3.5.2|Configura os plug-ins CNI de rede em cada nó.| |calico (kube-controllers)|3.5.2|Um centro controlador que configura a política de rede no cluster do IBM Cloud Private.| |console de gerenciamento do catálogo|3.2.0| Interface com o usuário do catálogo para visualizar, implementar e gerenciar cargas de trabalho do Kubernetes| |Gerenciador de certificados|0.7.0|Um componente que gerencia o ciclo de vida de certificados.| |CoreDNS|1.2.6|Fornece descoberta de serviço para aplicativos Kubernetes.| |Registro de Docker|18.06.2|Registro de imagem privado que é usado para armazenar arquivos de imagem de contêiner em um repositório de imagem. A versão de distribuição e registro do Docker é API V2.| |Backend Padrão|1,5|Componente secundário do controlador de ingresso que ajuda no roteamento de conexões de entrada para serviços em seu cluster.| |Elasticsearch|5.5.1|Armazena logs e métricas de sistema e aplicativo. O Elasticsearch também fornece uma API avançada que pode ser usada para consultar seus logs e métricas.| |etcd|3.2.24|Armazenamento de valor da chave distribuído que mantém dados de configuração.| |Filebeat|5.5.1|Coleta os logs para todos os componentes do sistema e contêineres de aplicativo do usuário que estão em execução em cada nó.| |GlusterFS|4.1.5|Um sistema de arquivos de armazenamento.| |Grafana|5.2.0|A visualização e monitoramento de dados com suporte para o Prometheus como origem de dados.| |console de gerenciamento de política|3.2.0| A interface com o usuário de política para visualizar, implementar e gerenciar políticas.| |Heapster|1.4.0.2|Conecta-se ao kubelet em execução em cada nó do trabalhador e coleta métricas de nó e de contêiner. Essas métricas incluem CPU, memória e uso de rede.| |Heketi|8.0.0|CLI para gerenciar GlusterFS.| |Helm (Tiller)|2.12.3|Gerencia gráficos do Kubernetes (pacotes).| |IBM Cloud Private console de gerenciamento|3.2.0|Um portal da web que é baseado na GUI de Abertura do DC/OS. Essa console de gerenciamento se conecta ao nó principal inicial usando o IP virtual (VIP) fornecido pelo gerenciador de VIP.| |Image Manager|2.2.5|Gerencia as imagens, fornecendo recursos estendidos para o registro do Docker. Esses recursos incluem autorização para operações de push, pull e remoção. O gerenciador de imagem também fornece autorização para a catalogação de bibliotecas de imagens.| |Indices-cleaner|1.0|Limpa os dados do Elasticsearch.| |Istio|1.0.6|Istio é uma plataforma aberta que pode ser usada para conectar, proteger, controlar e observar microsserviços. Com o Istio, é possível criar uma rede de serviços implementados que incluem o balanceamento de carga, a autenticação de serviço para serviço, o monitoramento e muito mais, sem mudar o código de serviço.| |Key Management Service|3.1.1| Provisionar e gerenciar chaves de criptografia.| |Kibana|5.5.1|Uma interface com o usuário que fornece acesso fácil aos dados armazenados no Elasticsearch, além da capacidade de criar visualizações e painéis desses dados.| |Kubelet|1.13.5|Supervisiona os componentes do sistema do cluster.| |Servidor API do Kubernetes|1.13.5|Fornece uma API de REST para validar e configurar dados para objetos do Kubernetes. Esses objetos do Kubernetes incluem pods, serviço e controladores de replicação.| |Kubernetes Control Manager|1.13.5|Mantém o estado compartilhado do cluster Kubernetes monitorando e ajustando o estado atual para garantir que o padrão de serviço necessário esteja em vigor. Essa manutenção é feita por meio do servidor de API do Kubernetes.| |Kubernetes Pause|3,1|Armazena o endereço IP para pods e configura o namespace de rede para outros contêineres que se juntam ao pod.| |Kubernetes Proxy|1.13.5|Pega o tráfego que é direcionado nos serviços do Kubernetes e o encaminha para os pods apropriados. O Kubernetes Proxy é iniciado pelo minion do Kubernetes.| |Kubernetes Scheduler|1.13.5|Designa pods para nós do trabalhador com base na política de planejamento.| |kube_state_metrics|1.2.0|Comunica-se com o servidor de API do Kubernetes para gerar métricas sobre o estado de objetos do Kubernetes.| |Logstash|5.5.1|Transforma e encaminha os logs que são coletados por Filebeat para Elasticsearch.| |Hub multicluster|3.2.0|Fornece o painel de gerenciamento e o serviço de procura para clusters.| |Medição|3.2.0|Coleta métricas de uso para seus aplicativos e cluster.| |Servidor de métricas| 0.3.1 | O Metrics Server é um agregador de todo o cluster de dados de uso de recurso. O Horizontal Pod Autoscaler (HPA) depende da API do Metrics para obter as métricas do nó.| |MongoDB|3.6|Banco de dados que é usado pelo OIDC, serviço de medição (IBM ® Cloud Product Insights), servidor de repositório Helm e servidor da API Helm.| |Controlador de ingresso NGINX|0.23.0|Usado para balanceamento de carga dos serviços do NodePort Kubernetes.| |nvidia-device-plugin| 1.2 | Forneça o recurso GPU para o cluster do Kubernetes.| |OpenID Connect (OIDC)|1.0|Protocolo de identidade sobre OAuth 2.0. O perfil do WebSphere Liberty é usado como o provedor OIDC. O perfil do Liberty pode ser configurado para se integrar com um servidor LDAP corporativo existente.| |API de plataforma/CLI cloudctl|3.2.0|Entrega downloads de binários da CLI, incluindo a CLI cloudctl e a API de back-end do cloudctl.| |Plataforma console de gerenciamento|3.2.0|Fornece o console de gerenciamento para recursos dentro do cluster.| |Componentes do Prometheus|

- Prometheus (2.8.0)
- collectd_exporter (0.4.0)
- node_exporter (0.16.0)
- configmap_reload (0.2.2)
- elasticsearch-exporter(1.0.2)
- kube-state-metrics-exporter (1.3.0)

|Coleta métricas de destinos configurados em intervalos fornecidos, avalia expressões de regras, exibe os resultados e pode acionar alertas se alguma condição é observada como sendo verdadeira.| |IBM Cloud Private ingresso de gerenciamento|2.2.3|Hospeda a console de gerenciamento e age como o proxy reverso para a API de todos os componentes do sistema.| |Serviço Catalog|0.1.40|Implementa o Open Service Broker API para fornecer integração do broker de serviço para IBM Cloud Private| |Serviço de funcionamento do sistema|3.2|Fornece o status de funcionamento dos componentes do cluster, como status do nó, status dos serviços de gerenciamento, detalhes de falha do pod| |UCarp|1.5.2|Usado para gerenciar o IP virtual (VIP) no nó principal. Esse componente ajuda a manter a alta disponibilidade (HA) no cluster. O UCarp requer um ambiente principal HA para iniciar.| |Roteador unificado|3.2.0|Usado para suportar o funcionamento de backend da console de gerenciamento do IBM

Cloud Private.| |vip_manager|1,1|| |Terminal da Web|3.2.0|Fornece o back-end para o recurso web-terminal na console de gerenciamento.|

Serviços e dependências de componentes

Visualize a tabela a seguir para obter uma lista de componentes e serviços de gerenciamento associados e dependências de componentes do IBM Cloud Private.

Nota: os serviços de gerenciamento em **negrito** identificam o serviço primário para o componente.

Componente	Serviços	Dependência
Gerenciador de alertas	**monitoramento	

|| |Criação de log de auditoria|audit-logging**|Gerenciador de certificados| |Gerenciador de autenticação|`auth-apikeys`,** auth-idp**, `auth-pap`, `auth-pdp`, `secret-watcher`|Gerenciador de certificados, Calico, MongoDB| |Calico (nó)|**calico, `calico-route-reflector`|| |calicoctl|**calico|| |Calico (CNI)|**calico|| |calico (kube-controllers)|**calico|| |Catalog console de gerenciamento|**catalog-ui|Servidor de API do Kubernetes, Gerenciador de autenticação, Helm (tiller), console de gerenciamento da plataforma| |Gerenciador de certificados|**ibm-cert-manager|| |CoreDNS|**kube-dns|| |Backend Padrão|**nginx-ingress|| |Elasticsearch|**log |Gerenciador de autenticação| |Filebeat|**log || |GlusterFS|**storage-glusterfs|| |Grafana|**monitoramento |Gerenciador de autenticação| |console de gerenciamento de política|**grc-ui|Servidor de API do Kubernetes, Gerenciador de autenticação, console de gerenciamento da plataforma| |Heapster|**heapster|| |Heketi|**storage-glusterfs|| |Helm (Tiller)|**tiller, `helm-api`, `helm-repo`, `mgmt-repo`|Gerenciador de autenticação, Gerenciador de certificados, Back-end padrão, MongoDB, API de plataforma| |IBM Cloud Private console de gerenciamento|**platform-ui, `catalog-ui`|Gerenciador de autenticação, MongoDB| |Image Manager|**image-manager|Gerenciador de certificados| |Indices-cleaner|**log || |Istio|**istio-citadel, `istio-egressgateway`, `istio-galley`, `istio-ingressgateway`, `istio-pilot`, `istio-policy`, `istio-sidecar-injector`, `istio-statsd-prom-bridge`, `istio-telemetry`, `jaeger-agent`, `jaeger-collector`, `jaeger-query`, `kiali`, `kiali-jaeger`, `prometheus`, `tracing`, `zipkin`, `grafana`|| |Serviço de gerenciamento de chaves|**key-management, `key-management-hsm`, `kmsplugin`|Gerenciador de autenticação, MongoDB| |Kibana|**log |Gerenciador de autenticação| |Logstash|**log || |Hub multicluster|**multicluster-hub, search|Servidor de API do Kubernetes, Gerenciador de autenticação, Helm (tiller)| |Medição|**medição |Gerenciador de autenticação, MongoDB, ingresso de gerenciamento do IBM Cloud Private| |Servidor de métricas|**metrics-server, `custom-metrics-adapter`|Gerenciador de autenticação| |MongoDB|**mongodb|| |Controlador de ingresso NGINX|**nginx-ingress|Backend Padrão| |OpenID Connect (OIDC)|**auth-idp|Gerenciador de autenticação| |API de plataforma/CLI cloudctl|** platform-api |Servidor de API do Kubernetes, Gerenciador de autenticação| |Plataforma console de gerenciamento|**platform-ui|Servidor de API do Kubernetes, Gerenciador de autenticação, console de gerenciamento de catálogo, Gerenciador de imagem| |Componentes do Prometheus|**monitoring, `monitoring-crd`|Metrics-server, Gerenciador de autenticação| |IBM Cloud Private ingresso de gerenciamento|**icp-management-ingress|Gerenciador de certificados| |Serviço Catalog|**service-catalog|Servidor de API do Kubernetes, Metrics-server, CoreDNS | |Serviço de funcionamento do sistema|**system-healthcheck-service|Servidor de API do Kubernetes, ingresso de gerenciamento do IBM Cloud Private | |Roteador unificado|**unified-router|| |Terminal da Web|**web-terminal**|Servidor de API do Kubernetes, API de plataforma, Gerenciador de autenticação|

Componentes do Vulnerability Advisor (VA) (recurso opcional)

Componente	Versão	Local	Função
Kafka	0.10.0.4	Nó do VA	Componente do pipeline de dados que é usado para a ingestão de dados e curadoria.
VA-Minio	RELEASE.2019-04-09T01-22-30Z.1	Nó do VA	O componente de armazenamento de dados de objetivo que é usado para indexação e consulta de dados do Vulnerability Advisor.
VA-minioCleaner	RELEASE.2019-04-03T17-59-57Z.1	Nó do VA	Usado para gerenciar o tamanho de dados do Vulnerability Advisor e limpar dados antigos. O curador VA-minioCleaner é implementado como uma CronJob.

Componente	Versão	Local	Função
Componentes do Security Analytics Service (SAS) • Servidor de API do SAS	3.2.0	Nó do VA	Componentes de serviço de frontend do Vulnerability Advisor. Os componentes do SAS fornecem APIs de RESTful para os crawlers do Vulnerability Advisor e o painel do Vulnerability Advisor. Os crawlers enviam informações de contêiner e imagem escaneadas, que são conhecidas como quadros, para o pipeline de dados do Vulnerability Advisor usando as APIs do SAS. O painel do Vulnerability Advisor também usa APIs do SAS para relatar descobertas do Vulnerability Advisor.
Statsd	0.7.2.1	Nó do VA	Usado pelo serviço do Vulnerability Advisor para monitoramento do sistema interno.
Anotadores de VA • Anotador de Arquivo MA • Anotador de MA de processo • Anotador de Conformidade do VA • Analisador de Configuração do VA • Anotador de Senha do VA • Anotador do Rootkit do VA • Anotador de Vulnerabilidade do VA	3.2.0	Nó do VA	Componentes de pipeline de dados do Vulnerability Advisor que melhoram a segurança de contêineres escaneados e dados de imagem usando várias análises, incluindo análise de vulnerabilidade, verificação de conformidade, análise de senha, análise de configuração e detecção de rootkit. Esses anotadores usam informações de segurança e conformidade internas e externas para melhorar a segurança de seus contêineres e imagens.
Indexadores do VA • Indexador de COS do VA • Indexador Genérico do VA	3.2.0	Nó do VA	Os componentes de pipeline de dados que são usados para indexar descobertas do Vulnerability Advisor no backend do Vulnerability Advisor.
Usncrawler do VA	3.2.0	Nó do VA	Componente do pipeline de dados que é usado para alimentar e agregar os avisos de segurança externa para os componentes de análise do Vulnerability Advisor.

Componente	Versão	Local	Função
Crawlers do VA	3.2.0	todos os nós	Os coletores de dados do Vulnerability Advisor, também conhecidos como crawlers, que inspecionam contêineres em execução e imagens de airgap. Esses crawlers extraem informações do sistema e do aplicativo que são usadas por todos os componentes de análise do Vulnerability Advisor. Os crawlers em tempo real e de métricas são executados em nós do trabalhador e são implementados como DaemonSets. Os crawlers de registro são executados como imagens de implementação e varreduras separadas que são implementadas no registro de imagem do IBM Cloud Private.
Controlador mcm MA	3.2.0	Nó do VA	Controlador de política MA que é usado para obter o resultado MA

e executar políticas MA do MCM no cluster do ICP.| | Zookeeper | 3.4.10 |Nó do VA | Usado pelo componente kafka no Orientador de Vulnerabilidade. |

Considerações sobre plataforma do IBM Cloud Private para preparação para o GDPR

Aviso

Este documento tem como objetivo ajudá-lo em suas preparações para prontidão do GDPR. Ele fornece informações sobre recursos da plataforma IBM Cloud Private que podem ser configurados e os aspectos de uso do produto, que devem ser considerados para ajudar sua organização com prontidão do GDPR. Essas informações não são uma lista exaustiva, devido às muitas formas que os clientes podem escolher e configurar recursos e à grande variedade de maneiras que o produto pode ser usado em si e com aplicativos e sistemas de terceiros.

Os clientes são responsáveis por assegurar sua própria conformidade com várias leis e regulamentações, inclusive com a General Data Protection Regulation da União Europeia. Os clientes são responsáveis apenas por obter aviso de uma consultoria jurídica competente quanto à identificação e interpretação de quaisquer leis e regulamentações relevantes que possam afetar os negócios dos clientes e quaisquer ações que os clientes possam precisar tomar para obedecerem a tais leis e regulamentações.

Os produtos, serviços e outros recursos descritos neste documento não são adequados para todas as situações dos clientes e podem ter disponibilidade restringida. A IBM não fornece aviso jurídico, contábil ou de auditoria nem representa ou garante que seus serviços ou produtos assegurarão que os clientes estejam em conformidade com qualquer lei ou regulamentação.

Índice

GDPR

O General Data Protection Regulation (GDPR) foi adotado pela União Europeia ("EU") e aplica-se a partir de 25 de maio de 2018.

Por que é importante? GDPR

O GDPR estabelece uma estrutura regulamentar de proteção de dados mais forte para processamento de dados pessoais de indivíduos. GDPR traz:

- Novos direitos e aprimorado para indivíduos
- Definição ampliada de dados pessoais
- Novas obrigações para processadores
- Potencial para multas financeiras significativas por não conformidade
- Notificação de violação de dados obrigatórios

Leia mais sobre o GDPR

- Portal de informações do GDPR da EU
- Website ibm.com/GDPR

Configuração do produto - considerações para Prontidão do GDPR

As seções a seguir descrevem aspectos de gerenciamento de dados na plataforma IBM Cloud Private e fornecem informações sobre recursos para ajudar os clientes com os requisitos do GDPR.

Ciclo de Vida de Dados

O IBM Cloud Private é uma plataforma de aplicativo para desenvolver e gerenciar aplicativos conteinerizados no local. Ele é um ambiente integrado para gerenciar contêineres que inclui o Kubernetes do orquestrador de contêiner, um registro de imagem privado, um console de gerenciamento e estruturas de monitoramento.

Dessa forma, a plataforma IBM Cloud Private lida principalmente com dados técnicos que estão relacionados à configuração e ao gerenciamento da plataforma, alguns dos quais podem estar sujeitos ao GDPR. A plataforma IBM Cloud Private também lida com informações sobre os usuários que gerenciam a plataforma. Esses dados serão descritos neste documento para o reconhecimento de clientes responsáveis por atender aos requisitos do GDPR.

Esses dados são persistidos na plataforma em sistemas de arquivos locais ou remotos como arquivos de configuração ou em bancos de dados. Os aplicativos que são desenvolvidos para serem executados na plataforma IBM Cloud Private podem lidar com outras formas de dados pessoais sujeitos ao GDPR. Os mecanismos que são usados para proteger e gerenciar os dados da plataforma também estão disponíveis para aplicativos que são executados na plataforma. Mecanismos adicionais podem ser necessários para gerenciar e proteger dados pessoais coletados por aplicativos executados na plataforma IBM Cloud Private.

Para entender melhor a plataforma IBM Cloud Private e seus fluxos de dados, deve-se entender como o Kubernetes, o Docker e o Helm funcionam. Esses componentes de software livre são fundamentais para a plataforma do IBM Cloud Private. Você usa as implementações do Kubernetes para colocar instâncias de aplicativos, que são construídas nos gráficos Helm que fazem referência às imagens do Docker. Os gráficos Helm contêm os detalhes sobre seu aplicativo e as imagens do Docker contêm todos os pacotes de software que seus aplicativos precisam executar.

O IBM Cloud Private inclui um catálogo de software e serviços conteinerizados da IBM na lista de repositórios padrão do IBM Cloud Private. Para visualizar uma lista de todos os gráficos do IBM Cloud Private, consulte IBM/gráficos. Para considerações sobre GDPR para os produtos no catálogo, consulte a documentação para esses produtos. As informações sobre os pacotes configuráveis do IBM Cloud Private disponíveis, que contêm a plataforma principal do IBM Cloud Private e software autorizado disponível, estão disponíveis aqui Pacotes configuráveis do IBM Cloud Private. Alguns dos aplicativos disponíveis no catálogo são software livre. É responsabilidade do cliente determinar e implementar os controles apropriados do GDPR para software livre. As informações sobre esses pacotes estão incluídas na entrada do catálogo.

A documentação sobre a plataforma IBM Cloud Private pode ser localizada na Coleção do IBM Cloud Private no IBM Knowledge Center.

Que tipos de dados fluem pela plataforma IBM Cloud Private

Como uma plataforma, o IBM Cloud Private lida com várias categorias de dados técnicos que poderiam ser considerados dados pessoais, como um ID de usuário de administrador e senha, IDs de usuário de serviço e senhas, endereços IP e nomes de nó do Kubernetes. A plataforma IBM Cloud Private também lida com informações sobre os usuários que gerenciam a plataforma. Os aplicativos que são executados na plataforma podem apresentar outras categorias de dados pessoais desconhecidos para a plataforma.

As informações sobre como esses dados técnicos são coletados/criados, armazenados, acessados, assegurados, registrados e excluídos são descritas em seções posteriores deste documento.

Dados pessoais usados para contato on-line com a IBM

Os clientes do IBM Cloud Private podem enviar comentários/feedback/solicitações on-line para entrar em contato com a IBM sobre assuntos do IBM Cloud Private de várias maneiras, principalmente:

- A comunidade pública IBM Cloud Private-CE (Community Edition) Slack
- Área de comentários públicos nas páginas da documentação do produto IBM Cloud Private no IBM Knowledge Center
- Comentários públicos no espaço do IBM Cloud Private de dW Answers

Geralmente, somente o nome do cliente e o endereço de e-mail são usados para permitir respostas pessoais para o assunto do contato e o uso de dados pessoais em conformidade com o IBM Online Privacy Statement em .

Coleta de Dados

A plataforma IBM Cloud Private não coleta dados pessoais sensíveis. Ele cria e gerencia dados técnicos, como um ID de usuário administrador e senha, IDs de usuário de serviço e senhas, endereços IP e nomes de nós do Kubernetes, que podem ser considerados dados pessoais. A plataforma IBM Cloud Private também lida com informações sobre os usuários que gerenciam a plataforma. Todas essas informações são acessíveis somente pelo administrador do sistema por meio de um console de gerenciamento com controle de acesso baseado na função ou pelo administrador do sistema por meio de login em um nó da plataforma IBM Cloud Private.

Os aplicativos que são executados na plataforma IBM Cloud Private podem coletar dados pessoais.

Quando você avalia o uso da plataforma IBM Cloud Private executando aplicativos conteinerizados e sua necessidade de atender aos requisitos de GDPR, deve-se considerar os tipos de dados pessoais que são coletados pelo aplicativo e os aspectos de como esses dados são gerenciados, como:

- Como os dados são protegidos enquanto fluem para/do aplicativo? Os dados são criptografados em trânsito?
- Como os dados são armazenados pelo aplicativo? A dados criptografados em repouso.
- Como as credenciais usadas para acessar o aplicativo são coletadas e armazenadas?
- Como as credenciais usadas pelo aplicativo para acessar origens de dados são coletadas e armazenadas?
- Como os dados coletados pelo aplicativo são removidos conforme necessário?

Esta não é uma lista definitiva dos tipos de dados que são coletados pela plataforma IBM Cloud Private. Ela é fornecida como um exemplo para consideração. Se você tiver quaisquer perguntas sobre os tipos de dados, entre em contato com a IBM.

Armazenamento de dados

A plataforma IBM Cloud Private persiste dados técnicos que estão relacionados à configuração e ao gerenciamento da plataforma em armazenamentos stateful em sistemas de arquivos locais ou remotos como arquivos de configuração ou em bancos de dados. Deve-se considerar assegurar todos os dados em repouso. A plataforma IBM Cloud Private suporta criptografia de dados em repouso em armazenamentos stateful que usam `dm-crypt`. Para obter mais informações, consulte Criptografando volumes usando dm-crypt.

Os itens a seguir destacam as áreas em que os dados são armazenados, que você pode desejar considerar para o GDPR.

- **Dados de configuração da plataforma:** a configuração da plataforma IBM Cloud Private pode ser customizada atualizando um arquivo YAML de configuração com propriedades para configurações gerais, Kubernetes, logs, rede, Docker e outras configurações. Esses dados são usados como entrada para o instalador da plataforma IBM Cloud Private para implementar um ou mais nós. As propriedades também incluem um ID de usuário de administrador e senha que são usados para autoinicialização. Para obter mais informações, consulte Customizando o cluster.
- **Dados de configuração do Kubernetes:** os dados de estado do cluster do Kubernetes são armazenados em um armazenamento de chave-valor distribuído, `etcd`. Para obter mais informações, consulte Componentes.
- **Dados de autenticação do usuário, incluindo IDs do usuário e senhas:** o gerenciamento de ID do Usuário e senha é manipulado por meio de um diretório LDAP corporativo do cliente. Os usuários e grupos que são definidos no LDAP podem ser incluídos em equipes da plataforma IBM Cloud Private e designados a funções de acesso. A plataforma IBM Cloud Private armazena o endereço de e-mail e o ID do usuário do LDAP, mas não armazena a senha. A plataforma IBM Cloud Private armazena o nome do grupo e, após o login, armazena em cache os grupos disponíveis aos quais um usuário pertence. A associação ao grupo não é persistida em qualquer forma de longo prazo. Deve-se considerar assegurar os dados do usuário e do grupo em repouso no LDAP corporativo. A plataforma IBM Cloud Private também inclui um serviço de autenticação, OpenID Connect (OIDC) que interage com o diretório corporativo e mantém tokens de acesso. Esse serviço usa MongoDB como um armazenamento auxiliar. Para obter mais informações, consulte Configurando a conexão LDAP.
- **Dados de autenticação de serviço, incluindo IDs do usuário e senhas:** as credenciais que são usadas pelos componentes da plataforma IBM Cloud Private para acesso entre componentes são definidas como Segredos do Kubernetes. Todas as

definições de recurso do Kubernetes são persistidas no armazenamento de dados de chave-valor `etcd`. Os valores de credenciais iniciais são definidos nos dados de configuração da plataforma como arquivos YAML de configuração de Segredo do Kubernetes. Para obter mais informações, consulte Gerenciando Segredos.

- **dados do gráfico Helm:** a plataforma IBM Cloud Private inclui um catálogo de software conteinerizados e serviços que você pode procurar e instalar em seu cluster de gráficos Helm. O serviço Helm persiste os dados de configuração em um armazenamento auxiliar do MongoDB. Para obter mais informações, consulte Gerenciando gráficos e apps e Componentes.
- **Sistema de arquivos de armazenamento GlusterFS:** é possível usar o armazenamento GlusterFS em seus clusters. Deve dar-se atenção à criptografia dos volumes nos quais o armazenamento do GlusterFS é implementado. Para obter mais informações, consulte GlusterFS e Criptografando volumes usando dm-crypt.
- **Dados de monitoramento:** é possível usar o monitoramento de plataforma IBM Cloud Private para monitorar o status de seu cluster e aplicativos. Este serviço usa o Grafana e o Prometheus para apresentar informações detalhadas sobre nós do cluster e contêineres. Pilhas adicionais de monitoramento podem ser implementadas para monitoramento de aplicativo. Os dados de monitoramento podem ser persistidos usando `PersistentVolumes` do Kubernetes. Para obter mais informações, consulte Serviço de monitoramento do IBM Cloud Private e Monitoramento de cluster do IBM Cloud Private.
- **Dados de medição:** é possível usar o serviço de medição IBM Cloud Private para visualizar e fazer download de métricas de uso detalhado para seus aplicativos e cluster. O serviço de medição usa o MongoDB como um armazenamento de dados auxiliar para persistir dados de métrica. Para obter mais informações, consulte Serviço de medição do IBM Cloud Private.
- **Dados de criação de log:** a plataforma IBM Cloud Private usa uma pilha ELK para logs do sistema. ELK é uma abreviação de três produtos, Elasticsearch, Logstash e Kibana, que são construídos pela Elastic e juntos formam uma pilha de ferramentas que podem ser usadas para transmitir, armazenar, procurar e monitorar logs. A pilha do ELK que é fornecida com a plataforma IBM Cloud Private usa as imagens da pilha do ELK oficial que são publicadas pelo Elastic. A criação de log é configurada por padrão para os serviços da plataforma IBM Cloud Private. Pilhas adicionais de ELK podem ser implementadas para criação de log de aplicativo. Para obter mais informações, consulte IBM Cloud Private log .

Acesso de Dados

Os dados da plataforma IBM Cloud Private podem ser acessados por meio do conjunto definido de interfaces do produto a seguir.

- Interface com o usuário da Web (o console de gerenciamento)
- Kubernetes `kubectl` da CLI
- IBM Cloud Private CLI
- CLI do Helm

Essas interfaces são projetadas para permitir que você faça mudanças administrativas em seu cluster do IBM Cloud Private. O acesso de administração ao IBM Cloud Private pode ser assegurado e envolve três estágios lógicos e ordenados quando uma solicitação é feita: autenticação, mapeamento de função e autorização.

Autenticação

O gerenciador de autenticação da plataforma IBM Cloud Private aceita as credenciais do usuário do console de gerenciamento e encaminha as credenciais para o provedor de backend OIDC, que valida as credenciais do usuário com relação ao diretório corporativo. O provedor OIDC então retorna um cookie de autenticação (`auth-cookie`) com o conteúdo de um JSON Web Token (`JWT`) para o gerenciador de autenticação. O token JWT persistir informações como o ID do usuário e o endereço de e-mail, além de associação ao grupo no momento da solicitação de autenticação. Esse cookie de autenticação é, então, enviado de volta para o console de gerenciamento. O cookie é atualizado durante a sessão. Ele é válido por 12 horas após você sair do console de gerenciamento ou fechar o navegador da web.

Para todas as solicitações de autenticação subsequentes feitas por meio do console de gerenciamento, o servidor front-end NGINX decodifica o cookie de autenticação disponível na solicitação e valida a solicitação chamando o gerenciador de autenticação.

A CLI da plataforma IBM Cloud Private requer que o usuário forneça credenciais para efetuar login.

A CLI `kubectl` também requer credenciais para acessar o cluster. Essas credenciais podem ser obtidas do console de gerenciamento e expiram após 12 horas. Acesso através de contas de serviço é suportado.

O acesso à CLI do Helm utiliza certificados para acessar o cluster.

Mapeamento de função

A plataforma IBM Cloud Private suporta o controle de acesso baseado na função (RBAC). No estágio de mapeamento de função, o nome do usuário que é fornecido no estágio de autenticação é mapeado para uma função de usuário ou grupo. As funções são usadas ao autorizar quais atividades administrativas podem ser realizadas pelo usuário autenticado.

Autorização

As funções da plataforma IBM Cloud Private controlam o acesso às ações de configuração de cluster para recursos do catálogo e do Helm e para recursos do Kubernetes. Várias funções IAM (Identity and Access Management) são fornecidas, incluindo Administrador de cluster, Administrador, Operador, Editor, Visualizador. Uma função é designada a usuários ou grupos de usuários quando você os inclui em uma equipe. O acesso da equipe a recursos pode ser controlado pelo namespace.

Segurança de Pod

As políticas de segurança de pod são usadas para configurar o controle de nível do cluster sobre o que um pod pode fazer ou o que ele pode acessar. Para obter mais informações, consulte

- Assegurando seu cluster
- Acessando o cluster
- Gerenciando seu cluster com o IBM Cloud Private CLI
- Trabalhando com gráficos.

Processamento de Dados

Os usuários do IBM Cloud Private podem controlar a maneira pela qual dados técnicos relacionados à configuração e ao gerenciamento são processados e assegurados por meio da configuração do sistema.

O **Controle de acesso baseado na função** (RBAC) controla quais dados e funções podem ser acessados pelos usuários.

As **Políticas de segurança de pod** são usadas para configurar o controle de nível do cluster sobre o que um pod pode fazer ou o que ele pode acessar.

Os **Dados em trânsito** são protegidos usando `TLS` e `IPSEC`. O `HTTPS` (`TLS` subjacente) é usado para assegurar a transferência de dados entre o cliente do usuário e os serviços de backend. Os usuários podem especificar o certificado raiz para usar durante a instalação. Todo o tráfego de dados entre nós pode ser criptografado pronto para utilização usando `IPSEC` sem mudar quaisquer aplicativos.

A proteção **Dados em repouso** é suportada usando `dm-crypt` para criptografar dados.

Os períodos de **Retenção de dados** para criação de log (ELK) e monitoramento (Prometheus) são configuráveis e a exclusão de dados é suportada por meio das APIs fornecidas.

Esses mesmos mecanismos de plataforma que são usados para gerenciar e assegurar os dados técnicos da plataforma IBM Cloud Private podem ser usados para gerenciar e assegurar dados pessoais para aplicativos desenvolvidos pelo usuário ou fornecidos pelo usuário. Os clientes podem desenvolver suas próprias capacidades para implementar controles adicionais.

Para obter mais informações, consulte

- Assegurando seu cluster
- Criptografando volumes usando dm-crypt
- Criptografando o tráfego de rede de dados do cluster com o IPsec
- Especificando sua própria autoridade de certificação (CA) para serviços do IBM Cloud Private

Exclusão de Dados

A plataforma IBM Cloud Private fornece comandos, interfaces de programação de aplicativos (APIs) e ações da interface com o usuário para excluir dados que são criados ou coletados pelo produto. Essas funções permitem que os usuários excluam dados técnicos, como IDs de usuário e senhas do serviço, endereços IP, nomes de nós do Kubernetes ou quaisquer outros dados de configuração da plataforma, bem como informações sobre usuários que gerenciam a plataforma.

Áreas da plataforma IBM Cloud Private a serem consideradas para suporte de exclusão de dados:

- O período de retenção de dados para dados de criação de log (ELK) é configurável.
- O período de retenção de dados para dados de monitoramento (Prometheus) é configurável.
- Os dados de criação de log podem ser excluídos da pilha ELK usando APIs do Elasticsearch.
- Os dados de monitoramento podem ser excluídos do Prometheus usando APIs do Prometheus.
- Todos os dados técnicos que estão relacionados à configuração da plataforma podem ser excluídos por meio do console de gerenciamento ou da API do Kubernetes `kubectl`.

Áreas da plataforma IBM Cloud Private a serem consideradas para suporte de exclusão de dados da conta:

- Todos os dados técnicos que estão relacionados à configuração da plataforma podem ser excluídos por meio do console de gerenciamento ou da API do Kubernetes `kubectl`.

A função para remover dados do ID do usuário e da senha que são gerenciados por meio de um diretório LDAP corporativo seria fornecida pelo produto LDAP usado com a plataforma IBM Cloud Private.

Os dados pessoais que são persistidos por criação de log e monitoramento da plataforma consistem em endereços IP de componentes do cluster e alguns nomes de usuário e IDs do usuário. Os aplicativos desenvolvidos pelo usuário ou fornecidos pelo usuário podem incluir outros dados pessoais em seu uso de criação de log e monitoramento. Os mesmos mecanismos que são usados para exclusão de dados de criação de log e monitoramento do sistema podem ser usados para dados de criação de log e monitoramento do aplicativo. Os dados pessoais que são coletados por aplicativos fora desses serviços irão requerer mecanismos fornecidos pelo aplicativo para excluir dados. Para obter mais informações, consulte

- IBM Cloud Private log
- IBM Cloud Private Serviço de monitoramento
- Documentação do Prometheus
- Logs e métricas de gerenciamento para Prometheus.

Monitoramento de Dados

- A plataforma IBM Cloud Private fornece um serviço de monitoramento para monitorar o status de seu cluster e aplicativos. Este serviço usa o Grafana e o Prometheus para apresentar informações detalhadas sobre nós do cluster e contêineres. O monitoramento pode ser configurado para gerar alertas ou integrado a provedores de alerta externos. Plataforma de monitoramento é ativada por padrão. Pilhas adicionais de monitoramento podem ser implementadas para monitoramento de aplicativo. Para obter mais informações, consulte Serviço de monitoramento do IBM Cloud Private e Monitoramento de cluster do IBM Cloud Private.
- O IBM Cloud Private fornece um serviço de medição para visualizar e fazer download de métricas de uso detalhado para seus aplicativos e cluster. A medição é ativada por padrão para todos os aplicativos do contêiner implementados. Para obter mais informações, consulte IBM Cloud Private serviço de medição.
- A plataforma IBM Cloud Private fornece um serviço de criação de log que é baseado na pilha ELK para transmitir, armazenar, procurar e monitorar logs. A pilha do ELK que é fornecida com a plataforma IBM Cloud Private usa as imagens da pilha do ELK oficial que são publicadas pelo Elastic. A criação de log é configurada por padrão para coletar logs do sistema para os serviços da plataforma IBM Cloud Private. Pilhas adicionais de ELK podem ser implementadas para criação de log de aplicativo. Para obter mais informações, consulte Criação de log do IBM Cloud Private.

Capacidade para restringir o uso de dados pessoais

Usando os recursos resumidos neste documento, a plataforma IBM Cloud Private permite que um usuário final restrinja o uso de quaisquer dados técnicos dentro da plataforma que sejam considerados dados pessoais.

Sob o GDPR, os usuários têm direitos para acessar, modificar e restringir o processamento. Consulte as outras seções deste documento para controlar o seguinte:

- Direito de acesso
  - Os administradores da plataforma IBM Cloud Private podem usar os recursos da plataforma IBM Cloud Private para fornecer aos indivíduos o acesso aos seus dados.
  - Os administradores da plataforma IBM Cloud Private podem usar os recursos da plataforma IBM Cloud Private para fornecer aos indivíduos as informações sobre quais dados a plataforma IBM Cloud Private retém sobre o indivíduo.
- Certo modificar
  - Os administradores da plataforma IBM Cloud Private podem usar os recursos da plataforma IBM Cloud Private para permitir que um indivíduo modifique ou corrija seus dados.
  - Os administradores da plataforma IBM Cloud Private podem usar os recursos da plataforma IBM Cloud Private para corrigir os dados de um indivíduo para eles.
- Certo para restringir o processamento
  - Os administradores da plataforma IBM Cloud Private podem usar os recursos da plataforma IBM Cloud Private para parar o processamento de dados de um indivíduo.

Apêndice - Dados registrados pela plataforma IBM Cloud Private

Como uma plataforma, o IBM Cloud Private lida com várias categorias de dados técnicos que poderiam ser considerados dados pessoais, como um ID de usuário de administrador e senha, IDs de usuário de serviço e senhas, endereços IP e nomes de nó do Kubernetes. A plataforma IBM Cloud Private também lida com informações sobre usuários que gerenciam a plataforma. Os

aplicativos que são executados na plataforma pode apresentar outras categorias de dados pessoais que são desconhecidos para a plataforma.

Este apêndice inclui detalhes sobre dados que são registrados pelos serviços da plataforma.

IBM Cloud Private segurança

- O dados são registrados
  - ID do Usuário, nome do usuário e endereço IP de usuários que efetuaram login
- Quando os dados são registrados
  - Com pedidos de login
- Onde dados são registrados
  - No logs de auditoria em `/var/lib/icp/audit`
  - No logs de auditoria em `/var/log/audit`
- Como excluir dados
  - Procure os dados específicos do usuário e exclua o registro do log de auditoria

Para obter informações adicionais, consulte

- Clusters do IBM Cloud Private de alta disponibilidade
- Gerando Kubernetes logs de auditoria
- Autenticação e Autorização logs de auditoria

IBM Cloud Private plataforma da API

- O dados são registrados
  - ID do usuário, nome do usuário e endereço IP do cliente em logs do contêiner
  - Dados de estado do cluster do Kubernetes no servidor `etcd`
  - Credenciais do OpenStack e do VMware no servidor `etcd`
- Quando os dados são registrados
  - Com pedidos de API
  - As credenciais armazenadas do `credentials-set` comando
- Onde dados são registrados
  - Em logs de contêiner, Elasticsearch e servidor `etcd`.
- Como excluir dados
  - Exclua logs de contêiner (`platform-api`, `platform-deploy`) de contêineres ou exclua as entradas de log específicas do usuário do Elasticsearch.
  - Limpe os pares chave-valor `etcd` selecionados usando o comando `etcdctl rm`.
  - Remova as credenciais chamando o comando `credentials-unset`.

Para obter mais informações, consulte

- Criação de log do Kubernetes
- etcdctl

IBM Cloud Private de monitoramento

- O dados são registrados
  - Endereço IP, nomes de pods, liberação, imagem
  - Dados extraídos de aplicativos desenvolvidos pelo cliente podem incluir dados pessoais
- Quando os dados são registrados
  - Quando o Prometheus extrai métricas de destinos configurados
- Onde dados são registrados
  - No servidor Prometheus ou volumes persistentes configurados
- Como excluir dados
  - Procure e exclua dados usando a API do Prometheus

Para obter informações adicionais, consulte

- Gerenciamento de logs e métricas para Prometheus
- Documentação do Prometheus

IBM Cloud Private Kubernetes

- O dados são registrados
  - Topologia de implementação do cluster (informações do nó para principal, trabalhador, proxy, va)
  - Configuração de serviço (mapa de configuração do k8s) e segredos (segredos do k8s)
  - ID do Usuário no log api-server
- Quando os dados são registrados
  - Ao implementar um cluster
  - Ao implementar um aplicativo do catálogo do Helm
- Onde dados são registrados
  - Topologia de implementação do cluster no `etcd`
  - Configuração e segredo para aplicativos implementados em `etcd`
- Como excluir dados
  - Use o console de gerenciamento do IBM Cloud Private
  - Procure e exclua dados usando o console de gerenciamento do k8s (`kubectl`) ou a API de REST `etcd`
  - Procure e exclua dados do log api-server usando a API do Elasticsearch

Tenha cuidado ao modificar a configuração de cluster do Kubernetes ou excluir dados do cluster.

Para obter informações adicionais, consulte

- O Kubectl do Kubernetes

IBM Cloud Private API Helm

- O dados são registrados
  - Nome de usuário e função
- Quando os dados são registrados
  - Quando um usuário recupera gráficos ou repositórios que são incluídos em uma equipe
- Onde dados são registrados
  - Logs de implementação api-helm, Elasticsearch
- Como excluir dados
  - Procure e exclua dados do log helm-api usando a API do Elasticsearch

IBM Cloud Private Service Broker

- O dados são registrados
  - ID do usuário (somente no nível de log de depuração 10, não no nível de log padrão)
- Quando os dados são registrados
  - Quando solicitações de API são feitas no broker de serviço
  - Quando o broker de serviço acessa o catálogo de serviços
- Onde dados são registrados
  - Log de contêiner do broker de serviço, Elasticsearch
- Como excluir dados
  - Procure e exclua o log api-server usando a API do Elasticsearch
  - Procure e exclua o log do contêiner api-server

```
  kubectl logs $(kubectl get pods -n kube-system | grep  service-catalogapiserver | awk
'{print $1}') -n kube-system | grep admin
```

Para obter informações adicionais, consulte

- O Kubectl do Kubernetes

IBM Cloud Private considerações de plataforma para conformidade FIPS

O Federal Information Processing Standards (FIPS) são padrões de tecnologia da informação que são desenvolvidos pelo governo federal dos Estados Unidos relacionados à codificação e criptografia de dados.

Para as cifras que são suportadas pelo FIPS, consulte os documentos a seguir:

- Para o Red Hat Enterprise Linux (RHEL), consulte Guia de Segurança.
- Para o Ubuntu, consulte Centro de recurso de segurança de computador.
- Para o SUSE Linux Enterprise Server (SLES), consulte Computer Security Resource Center.

É possível atender aos requisitos FIPS para o IBM Cloud Private usando os procedimentos a seguir:

- Criptografando o tráfego de rede de dados do cluster com o IPsec
- Criptografando volumes usando dm-crypt
- Criptografando o tráfego de rede para terminais externos e a console de gerenciamento, o serviço de ingresso, o gerenciador de imagem, o registro do Docker e o gerenciador de autenticação. Para obter mais informações, consulte o parâmetro `fips_enabled` na página Customizando o cluster com o arquivo config.yaml. Para ativar ou desativar o modo FIPS após a instalação IBM Cloud Private, consulte Desativando e ativando o modo FIPS.
- Criptografando segredos do Kubernetes com o plug-in Key Management Service
- Exemplo: Ativando FIPS no IBM Cloud Private

IBM Cloud Private considerações de plataforma para preparação de PCI

O Payment Card Industry Data Security Standard (PCI DSS) é uma coleção de objetivos e requisitos correspondentes para proteção de um ambiente de dados do titular do cartão. O ambiente de dados do titular do cartão, conforme definido pelo PCI Security Standards Council, representa "pessoas, processos e tecnologia que armazenam, processam ou transmitem dados do titular do cartão ou que afetam a segurança dos dados do titular do cartão". O DSS é dividido em 6 objetivos de controle e 12 requisitos de nível superior.

A IBM contratou uma empresa QSA de terceiros, Weaver (que trabalha com vários aspectos do IBM Cloud) para revisar a plataforma IBM® Cloud Private (ICP) e desenvolver diretrizes de PCI para usuários do ICP. O resultado é um white paper que descreve considerações e orientações para organizações que estão considerando a plataforma IBM Cloud Private e como ela pode ajudar a suportar a implementação de requisitos do PCI DSS 3.2.1. Cada cliente é responsável por determinar se o ambiente e a configuração do IBM Cloud Private atendem aos requisitos do Payment Card Industry Data Security Standard (PCI DSS) 3.2.1.

Para obter mais informações, consulte o Guia de implementação do IBM Cloud Private Platform PCI DSS 3.2.1.

Idiomas Suportados

A interface com o usuário do IBM® Cloud Private 3.2.0 está localizada para vários idiomas. A documentação do produto também é liberada em vários idiomas.

Tabela 1. Idiomas para os quais a interface com o usuário do IBM Cloud Private está localizada

Linguagem	Código de Idioma
Português do Brasil	pt_br
Inglês	en
Francês	Fr
Alemão	De
Italiano	it
Japonês	Ja
Coreano	Ko
Chinês Simplificado	zh_CN
Chinês tradicional	zh_TW
Espanhol	Es

Tabela 2. A documentação do produto IBM® Cloud Private 3.2.0 está disponível nestes idiomas

Linguagem	Código de Idioma
Português do Brasil	pt_br
Inglês	Pt-br
Francês	Fr
Alemão	De
Japonês	Ja
Coreano	Ko
Chinês Simplificado	zh_CN

Linguagem	Código de Idioma
Espanhol	Es

Nota: a documentação do produto IBM Cloud Private é traduzida para as geografias participantes, mas a versão em inglês é atualizada continuamente. As discrepâncias entre o inglês e as versões traduzidas podem aparecer entre os ciclos de tradução. Verifique a versão em inglês para ver se alguma discrepância foi resolvida depois que as versões traduzidas foram publicadas.

Planejando seu Cluster

Antes de instalar o IBM® Cloud Private, planeje seu cluster. Algumas partes da instalação não podem mudar a instalação após a instalação.

- Requisitos do sistema
- Dimensionando seu cluster
- Preparando-se para proteger seu cluster
- Terminais do IBM Cloud Private
- ConfigMap de configuração de cluster

Requisitos do Sistema

As definições de configuração de software, hardware e sistema que são necessárias para configurar um cluster do IBM® Cloud Private.

- Requisitos e recomendações de hardware
- Sistemas operacionais e plataformas suportados
- Navegadores suportados
- Versões suportadas do Docker
- Sistemas de arquivos e armazenamento suportados
- IaaS, hypervisors e ambientes suportados
- Portas necessárias

Requisitos e recomendações de hardware

Revise os requisitos mínimos de CPU, Memória, RAM e espaço em disco para configurar e executar clusters do IBM® Cloud Private.

Nota: assegure-se de revisar e verificar se você atende aos requisitos de memória aumentados. Para obter mais informações, consulte a seção *Requisitos de hardware*.

As tabelas a seguir listam os requisitos mínimos do sistema por nó para execução do IBM Cloud Private. O requisito mínimo para o IBM Cloud Private é um nó (e proxy) principal, um nó de gerenciamento e um nó do trabalhador.

- Requisitos de hardware
  - Requisitos de nó único
  - Requisitos de Multi-node
- Requisitos do Espaço em Disco
  - Requisitos de espaço em disco no momento da instalação
  - Requisitos de espaço em disco de tempo
  - Requisitos de memória e CPU no momento da instalação
  - Requisitos de memória e CPU do tempo de execução
- Requisitos do ambiente PowerVM
- Requisitos de ambiente do Linux on IBM Z and LinuxONE

Requisitos de hardware

Requisitos de nó único

| Requisito| Todos os serviços de gerenciamento ativados | Todos os serviços de gerenciamento incluindo criação de log desativados| |------|--------|-------| | Número de hosts | 1 | 1 | |Núcleos | 8 ou mais | 8 ou mais | | CPU | >=2,4 GHz | >=2,4 GHz| |RAM | 32 GB ou mais | 16 GB ou mais | | Espaço livre em disco para instalar | >=200 GB | >=150 GB |

Nota para CPUs:

- Para um cluster do Linux® x86_64, use uma CPU que suporte o SSE 4.2.
- Para um cluster do Linux® on Power® (ppc64le), use uma CPU que seja versão Power8 ou superior.
- Para um cluster do Linux® on IBM® Z and LinuxONE, use uma CPU que seja da versão EC12 ou mais recente ou qualquer sistema LinuxONE.

Requisitos de Multi-node

Nota: se você não usar um nó de gerenciamento em seu cluster de múltiplos nós, assegure-se de que o nó principal atenda aos requisitos do nó de gerenciamento, além do nó principal.

Tabela 2. Requisitos mínimos de hardware para um cluster com vários nós

Requisito	Nó de inicialização	Nó principal	Nó do proxy	Nó do trabalhador	Nó de gerenciamento	Nó do VA	nó etcd
Número de hosts	1	1, 3 ou 5	1 ou mais	1 ou mais	1 ou mais	1, 3 ou 5	1 ou mais números ímpares de nós
Núcleos	1 ou mais	8 ou mais	2 ou mais	2 ou mais	8 ou mais	8 ou mais	1 ou mais
CPU	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz
RAM	>=4 GB	>=16 GB	>=4 GB	>=4 GB	• >=16 GB • >= 32 GB (ambiente IBM Power)	>=16 GB	>=4 GB
Espaço livre em disco para instalar	>=100 GB	• >=300 GB • >=800 GB (ambiente de produção)	• > =150 GB • >=350 GB (ambiente de produção)	>=150 GB	>=300 GB	• >=300 GB • >=800 GB (ambiente de produção)	>=100 GB

Notas:

1. Para CPUs:
   - Para um cluster do Linux x86_64, use uma CPU que suporte o SSE 4.2.
   - Para um cluster do Linux on Power (ppc64le), use uma CPU que seja versão Power8 ou superior.
   - Para um cluster do Linux on IBM Z and LinuxONE, use uma CPU que seja da versão EC12 ou mais recente ou qualquer sistema LinuxONE.
2. Um núcleo do processador virtual (VPC) é uma unidade de medida que é usada para determinar o custo de licenciamento de produtos IBM. Ele é baseado no número de núcleos virtuais (vCPUs) que estão disponíveis para o produto. Uma vCPU é um núcleo virtual que é designado a uma máquina virtual ou a um núcleo de processador físico quando o servidor não é particionado para máquinas virtuais. Uma vCPU é equivalente a uma CPU do Kubernetes. Para obter mais detalhes, consulte Significado de CPU no Kubernetes.
3. Se você desativar a criação de log e/ou o monitoramento durante a instalação, será possível economizar alguma RAM e CPU. Se você deseja ativar a criação de log e/ou o monitoramento, consulte os tamanhos de implementação de amostra em Dimensionando seu cluster.
4. Por padrão, o `systemReserved` e o `kubeReserved` reserva 0.2 GHz de processamento de CPU e 512 MB de memória. É possível reservar mais recursos para tornar a Plataforma Kubernetes mais estável, especialmente na plataforma Power. **Lembre-se:** o recurso reservado adicional deve ser considerado durante o planejamento de seu requisito de hardware.
5. Para a plataforma Power, o exemplo a seguir contém os valores sugeridos. Consulte Reconfigurando Kubelet em um cluster em tempo real para conhecer as etapas que são necessárias para a configuração do valor. Para um nó de gerenciamento único, que requer 16 GB de memória, você deve expandir seu recurso de host de gerenciamento para usar pelo menos 20 GB de memória antes de reconfigurar o Kubelet em um cluster ativo.

```
systemReserved:
 cpu: "500m"
 memory: "1500Mi"
 ephemeral-storage: "1Gi"
kubeReserved:
 cpu: "500m"
 memory: "1500Mi"
 ephemeral-storage: "1Gi"
```

Requisitos do Espaço em Disco

Requisitos de espaço em disco de tempo de instalação

Tabela 3. Requisitos mínimos de armazenamento para o momento da instalação

Local	Espaço em disco mínimo	Nó	Descrição
Diretório para colocar imagens off-line	50 GB	Nó de inicialização	O diretório é usado para armazenar arquivos de instalação.
Diretório para carregar imagens off-line	100 GB	Nó de inicialização	O diretório é usado para carregar as imagens off-line pelo Docker

Nota: os requisitos de espaço em disco no momento da instalação são necessários para uma instalação bem-sucedida. Em Instalando o IBM Cloud Private, o `diretório para colocar imagens off-line` é `/opt/ibm-cloud-private-3.2.0` e o `diretório para carregar imagens off-line` é o diretório no qual você coloca o arquivo de instalação.

Requisitos de espaço em disco de tempo de execução

Local	Espaço em disco mínimo	Espaço em disco ideal	Nó
/	300 GB 200 GB 300 GB	>=800 GB >=600 GB >=1000 GB	Principal

e gerenciamento
Trabalhador, proxy e etcd
VA| | `/tmp/` | 50 GB | >=50 GB | Todos os nós | | `/var/` | 200 GB
150 GB
200 GB | >=700 GB
>=550 GB
>=900 GB |Principal e gerenciamento
Trabalhador, proxy e etcd
VA | | `/var/lib/docker`| 100 GB | >=400 GB| Todos os nós| | `/var/lib/etcd`| 10 GB | >=20 GB | Mestre ou etcd | | `/var/lib/etcd-wal`| 2 GB | >=4 GB | Mestre ou etcd | | `/var/lib/icp`| 50 GB | > =150 GB|Principal e gerenciamento| | `/var/lib/icp/va`| 100 GB | >=350 GB | VA | | `/var/lib/kubelet`| 30 GB | > =150 GB|Todos os nós | | `/var/log/cloudsight`| 10 GB | > =10 GB | VA|

Notas:

- Os requisitos de espaço em disco que são mencionados na tabela 4 incluem o espaço dos subdiretórios e podem ser reduzidos de modo apropriado quando os subdiretórios estão localizados em outro lugar.
- O espaço em disco mínimo é o espaço mínimo para execução. É recomendado seguir os requisitos de espaço em disco ideais no ambiente de produção.
- Se várias funções de cluster estiverem instaladas em um nó, o requisito do disco será a soma do requisito do disco para cada função. No ambiente de produção, não é recomendado instalar várias funções de cluster em um nó.
- Se o nó etcd estiver separado, o diretório `/var/lib/etcd` estará no nó etcd.
- Nos nós do trabalhador, o diretório `/var/lib/docker` requer mais espaço em disco, pois as imagens de produção são colocadas dentro.
- O diretório `/var/lib/registry` é uma montagem compartilhada de um sistema de arquivos compartilhado externo e precisa de pelo menos 50 GB, caso o cluster seja um cluster de combinação. Ele deve ser grande o suficiente para hospedar todas as imagens do Docker que você pretende armazenar no registro de imagem privado.
- O diretório `/var/lib/kubelet` precisa de pelo menos 10 GB de espaço em disco. Se você ativar o Vulnerability Advisor, o nó do VA precisará de >=100 GB de espaço em disco.

Importante: o diretório `/var` é o local de armazenamento padrão para a maioria das imagens do Docker e dos contêineres que são usados em seu cluster do IBM Cloud Private. Os diretórios a seguir são usados pelo instalador, mas não requerem quantidades

significativas de espaço em disco:

- `/etc/cfc` - este diretório armazena o arquivo-chave de certificação e configuração do IBM Cloud Private.
- `/opt/ibm/cfc` - este diretório armazena os arquivos de licença do IBM Cloud Private.

Para evitar problemas de espaço em disco, monte os diretórios de armazenamento padrão em caminhos separados que tenham capacidades do disco maiores. Para obter mais informações sobre como montar o diretório de armazenamento do Docker (`/var/lib/docker`), consulte Especificando um diretório de armazenamento do Docker padrão usando montagem bind. Também é possível usar esse método de ligação para montar os outros diretórios de armazenamento padrão do IBM Cloud Private. Para evitar problemas de espaço em disco em seu cluster, você pode desejar usar uma montagem bind para montar os diretórios a seguir:

- Etcd - `/var/lib/etcd`
- VA - `/var/lib/icp`
- Serviço Kubelet - `/var/lib/kubelet`

Para obter mais informações sobre como montar os diretórios de armazenamento padrão, consulte Especificando outros diretórios de armazenamento padrão usando montagem bind.

Requisitos de CPU e de memória do tempo de instalação

Nota: as tarefas na tabela a seguir são executadas uma vez durante o processo de instalação.

Tabela 5. Requisitos de memória e CPU no momento da instalação

Componente	CPU	Memória
`client-registration` tarefa IAM	100 milinúcleos (m)	128 MB
`security-onboarding` tarefa IAM	20 m	50 MB
`iam-onboarding` tarefa IAM	20 m	50 MB

Requisitos de CPU e de memória de tempo de execução

Componente	CPU	Memória
`auth-idp` serviço IAM	210 m	660 MB
`auth-pap` serviço IAM	70 m	220

MB| | `auth-pdp` serviço IAM| 30 m | 50 MB| | `secret-watcher` serviço IAM| 10 m | 10 MB| | `system-healthcheck-service` | 25 m | 32 Mi|

Requisitos do ambiente PowerVM

Os valores na tabela a seguir se aplicam especificamente aos ambientes PowerVM. Eles não se aplicam aos ambientes de Kernel-based Virtual Machine (KVM) ou aos ambientes bare-metal.

Tabela 7. Requisitos mínimos de hardware para um cluster multinós em um ambiente PowerVM

Requisito	Nó de inicialização	Nó principal	Nó do proxy	Nó do trabalhador	Nó de gerenciamento	Nó do VA	nó etcd
Número de hosts	1	1, 3 ou 5	1 ou mais	1 ou mais	1 ou mais	1, 3 ou 5	1 ou mais números ímpares de nós
vCPUs	1 ou mais	2 ou mais	1 ou mais	1 ou mais	2 ou mais	• 2 ou mais • 4 ou mais (ambiente de produção)	1 ou mais

Requisito	Nó de inicialização	Nó principal	Nó do proxy	Nó do trabalhador	Nó de gerenciamento	Nó do VA	nó etcd
Unidades de Processador	0,5 ou mais	• 2 ou mais • 4 ou mais (ambiente de produção)	• 1 ou mais • 2 ou mais (ambiente de produção)	1 ou mais	• 2 ou mais • 4 ou mais (ambiente de produção)	• 2 ou mais • 4 ou mais (ambiente de produção)	1 ou mais

Recomendações para ambientes PowerVM:

- Pelo menos quatro máquinas virtuais (VMs - também chamadas de LPARs) são recomendadas; com o nó principal, o nó de gerenciamento, o nó do orientador de vulnerabilidade e os nós do trabalhador em máquinas virtuais separadas. Em ambientes de alta escala, o nó do etcd também deve estar em uma MV separada.
- O uso de processadores compartilhados e ilimitados é recomendado para permitir a consolidação de CPUs conforme necessário. Se processadores dedicados forem usados, será necessário seguir as diretrizes para Cores (vCPUs). Para obter as recomendações do conjunto de processadores compartilhados, consulte: Processadores compartilhados .
- Em um ambiente de larga escala, provavelmente mais vCPUs e unidades de processador serão necessárias. No entanto, é necessário manter a proporção de unidade entre o vCPU e o processador que está listada na tabela anterior. Por exemplo, o uso de oito vCPUs para seu nó de gerenciamento requer quatro unidades de processador.
- Consulte Definindo as funções de nó no arquivo de hosts para obter mais informações sobre como configurar nós do etcd.

Requisitos do Ambiente do Linux on IBM Z and LinuxONE

Os valores na tabela a seguir aplicam-se especificamente a ambientes Linux on IBM Z and LinuxONE.

Nota:

- Deve-se usar uma arquitetura s390x separada Linux® LPAR ou zKVM guest para construir imagens do Docker para seus aplicativos.

Tabela 8. Requisitos mínimos de hardware para um cluster multinós no IBM® Z e no LinuxONE

Requisito	Nó de inicialização	Nó principal	Nó do proxy	Nó do trabalhador	Nó de gerenciamento
Número de hosts	1	1	1 ou mais	1 ou mais	1 ou mais
Núcleos (IFLs)	1	2	1 ou mais	1 ou mais	1 ou mais
CPU	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz	>= 2,4 GHz
RAM	>=4 GB	>=16 GB	>=4 GB	>=4 GB	>=16 GB
Espaço livre em disco para instalação	>=100 GB	>=200 GB	>=150 GB	>=150 GB	>=200 GB

Sistemas operacionais e plataformas suportados

Nota: certifique-se de revisar e verificar se você atende aos requisitos de memória aumentada. Para obter mais informações, consulte Requisitos de hardware .

O IBM® Cloud Private suporta os seguintes sistemas operacionais de 64 bits: Linux® e Linux® on Power® (ppc64le).

Tabela 1. Sistemas operacionais suportados

Plataforma	Sistema Operacional
Linux® x86_64	Red Hat Enterprise Linux (RHEL) 7.4, 7.5 e 7.6
	Ubuntu 18.04 LTS e 16.04 LTS
	SUSE Linux Enterprise Server (SLES) 12 SP4
Linux® on Power® (ppc64le) POWER8® ou mais tarde	Red Hat Enterprise Linux 7.4, 7.5*, 7.5-Alt e 7.6
	Ubuntu 18.04 LTS (apenas KVM e Bare Metal) e Ubuntu 16.04 LTS
	SUSE Linux Enterprise Server 12 SP4
Linux® on IBM® Z and LinuxONE	Red Hat Enterprise Linux 7.3, 7.4, 7.5 e 7.6
	Ubuntu 18.04 LTS e 16.04 LTS

Plataforma	Sistema Operacional
	SUSE Linux Enterprise Server 12 SP4

* Os usuários do RHEL 7.5 no POWER9™ devem certificar-se de instalar o kernel `kernel-alt` mais recente fornecido pelo RHEL.

Nota: verifique a documentação de seu sistema operacional para assegurar-se de que você está usando um nível de kernel suportado.

Os componentes do IBM Cloud Private são distribuídos como um conjunto de imagens do Docker que incorporam suas próprias dependências do sistema operacional. Recomenda-se usar um dos sistemas operacionais certificados listados na tabela anterior. No entanto, o IBM Cloud Private pode ser executado em qualquer sistema operacional Linux que suporte o Docker 1.12 e mais recente.

Tipos de nós suportados por plataforma

Tabela 2. Tipo de nó suportado por plataformas

Tipo de Nó	Linux x86_64 (x86_64)	Linux on Power (ppc64le)	Linux on IBM Z and LinuxONE (s390x)
Inicialização	S	S	S
Principal	S	S	S
Gerenciamento	S	S	S
Proxy	S	S	S
Trabalhador	S	S	S
VA	S	S	S

Importante: O VA não suporta o SLES.

Recursos suportados por plataforma

Recurso	Linux x86_64 (x86_64)	Linux on Power (ppc64le)	Linux on IBM Z and LinuxONE (s390x)	Notas
Cloud Foundry	S	N	N	Este recurso não está disponível no Community Edition.
Cloud Automation Manager	S	S	S	O IBM Cloud Automation Manager (consulte a nota 1) pode gerenciar máquinas virtuais IBM z/VM 6.4 e z/VM v7.1 (consulte as notas 2, 3, 4) usando o z/VM Cloud Manager Appliance (CMA).
[Instalação				

](../installing/install.md)| S | S | S |A instalação suportada apenas em nós principais ou nós de inicialização dedicados.| | Console de Gerenciamento |S|S|S|O console de gerenciamento é executado apenas nos nós principais.| | Criação de Log|S|S|S/A criação de log está disponível como uma visualização de tecnologia e requer mudanças na configuração de pós-instalação para ativar sua funcionalidade. Para obter informações adicionais, consulte Criação de log do IBM Cloud Private com o IBM® Z.| | Monitoramento

- Prometheus
- Grafana

|S|S|S/Enquanto Prometheus e Grafana são executados apenas nos nós principais ou de gerenciamento, os dados de nós do trabalhador são coletados usando o exportador do nó.| | Segurança e RBAC|S|S|S|| | Vulnerability Advisor|S|S|S|Este recurso não está disponível no Community Edition. **Nota**: O VA não suporta SLES.| | IPsec|S|S|S|| | Rede: Calico|S|S|S|| | Rede: NSX-T|S|N|N|| | Armazenamento: GlusterFS|S|S|S| | |Armazenamento: VMware|S|N|N|| |Armazenamento: Minio|S|S|S|| | Medição|S|S|S|| |Repositório ou API Helm|S|S|S|| | Suporte à GPU do Nvidia|S|S|N|| | Serviço de verificação de funcionamento do cluster |S|S|N| Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.|

Notas:

1. O IBM Cloud Private suporta novas versões de sistemas operacionais suportados, Kubernetes, Docker e outra infraestrutura dependente após ocorrerem novas liberações e quando elas são totalmente testadas pela equipe do IBM Cloud Private.
2. O suporte para o Cloud Manager Appliance (CMA) no z/VM v7.1 é uma oferta temporária até que uma solução estratégica de longo prazo que substitua o CMA seja disponibilizada.
3. Para o Cloud Manager Appliance on z/VM v7.1, o suporte é limitado a uma versão anterior do OpenStack e a versões anteriores do Linux Distributions para implementação, sem planos para upgrade. A versão suportada do OpenStack é

"Newton". As distribuições Linux® suportadas para implementação são:
    - RHEL 6.2, 6.3, 6.4, 6.5, 6.6 e 6.7
    - RHEL 7.0, 7.1 e 7.2
    - SLES 11.2, 11.3 e 11.4
    - SLES 12 e SLES 12.1
    - Ubuntu 16.04
4. O Cloud Manager Appliance está disponível para o z/VM v7.1 somente para o caso de uso específico com o IBM Cloud Private.

Navegadores Suportados

É possível acessar o IBM® Cloud Private console de gerenciamento a partir do Mozilla Firefox, Google Chrome, Microsoft™ Edge e Safari.

Tabela 1. Navegadores suportados

Plataforma	Navegadores Suportados
Microsoft Windows™	• Microsoft Edge-versão mais recente • Mozilla Firefox-versão mais recente para o Windows • Google Chrome-versão mais recente para o Windows
Linux®	• Mozilla Firefox-versão mais recente para o Linux • Google Chrome-versão mais recente para o Linux
MacOS	• Mozilla Firefox - versão mais recente para Mac • Google Chrome - versão mais recente para Mac • macOS Safari-versão mais recente

Nota: não é possível instalar o IBM Cloud Private em um sistema que usa um sistema operacional Windows ou Mac. No entanto, é possível acessar o IBM Cloud Private console de gerenciamento a partir de um host do Windows ou do Mac usando um navegador.

Versões suportadas do Docker

O IBM® Cloud Private suporta a integração do Docker em sistemas operacionais selecionados.

Versões do Docker

Tabela 1. Versões do Docker que são suportadas pelo IBM Cloud Private

Plataforma	Sistema Operacional	Docker CE	Docker EE
Linux®	Red Hat Enterprise Linux (RHEL) 7.4, 7.5 e 7.6	Consulte notas	17.03 a 18.03.1, 18.06.2
	Ubuntu 18.04 LTS	18.03.1, 18.06.2	18.03.1, 18.06.2
	Ubuntu 16.04 LTS	1.12 a 18.03.1, 18.06.2	17.03 a 18.03.1, 18.06.2
	SUSE Linux Enterprise Server (SLES) 12 SP3, SP4	Consulte notas	Consulte notas
Linux on POWER de 64 bits Little Endian (LE)	Red Hat Enterprise Linux 7.4, 7.5 e 7.6	Consulte notas	17.03 a 18.03.1, 18.06.2
	Ubuntu 18.04 LTS	18.03.1, 18.06.2	18.03.1, 18.06.2
	Ubuntu 16.04 LTS	1.12 a 18.03.1, 18.06.2	17.03 a 18.03.1, 18.06.2
	SUSE Linux Enterprise Server 12 SP3, SP4	Consulte notas	Consulte notas
Linux® on IBM® Z and LinuxONE	Red Hat Enterprise Linux 7.4, 7.5 e 7.6	Consulte notas	17.03 a 18.03.1, 18.06.2
	Ubuntu 18.04 LTS	18.03.1, 18.06.2	18.03.1, 18.06.2
	Ubuntu 16.04 LTS	1.12 a 18.03.1, 18.06.2	17.03 a 18.03.1, 18.06.2
	SUSE Linux Enterprise Server 12 SP3, SP4	Consulte notas	Consulte notas

Notes

- Para sistemas RHEL, é possível os pacotes do Docker IBM Cloud Private fornecidos, consulte Pacotes do Docker IBM Cloud Private fornecidos.
- O IBM Cloud Private Versão 3.2.0 foi testado somente nas versões do Docker que são descritas na *Tabela 1*. As versões posteriores do Docker pode funcionar. Para obter mais informações sobre a política de compatibilidade do Docker, consulte Mudanças drásticas e incompatibilidades.
- Para nós do SLES, é possível instalar manualmente o Docker usando as instruções de instalação do Docker na documentação do SLES ou usando o pacote do Docker IBM Cloud Private fornecido. A versão do Docker que está instalada nos nós do SLES é a versão 18.06.1.

Pacotes do Docker do IBM Cloud Private fornecidos

O IBM Cloud Private fornece pacotes do Docker que podem ser usados para instalação em nós de inicialização e do cluster.

O pacote do Docker para instalação manual não está disponível para o IBM Cloud Private-CE. Para o IBM Cloud Private-CE, deve-se instalar manualmente uma versão suportada do IBM Cloud Private do Docker por meio do website do Docker. Veja Instalando o Docker manualmente por meio do website do Docker.

- Para instalação manual em nós de inicialização e do cluster - os pacotes do Docker podem ser obtidos no website IBM Passport Advantage®. Para obter mais informações sobre como fazer download desses pacotes, entre em contato com seu representante de vendas IBM. Para concluir uma instalação manual, consulte Instalando o Docker manualmente usando o pacote do Docker do IBM Cloud Private fornecido.
- Para instalação em nós do cluster - Os pacotes do Docker estão disponíveis no diretório `/<installation_directory>/cluster/runtime-engine` e podem ser instalados automaticamente nos nós do cluster durante a instalação. As instruções para instalação nos nós do cluster são cobertas nos tópicos de instalação do IBM Cloud Private, consulte Instalando o IBM Cloud Private.

Plataforma	Sistemas Operacionais	Nome do pacote	Nome binário
Linux® x86_64	Red Hat Enterprise Linux, Ubuntu, SUSE Linux Enterprise Server	IBM Cloud Private 3.2.0 Docker for Linux®	icp-docker-18.06.2_x86_64.bin
Linux® on Power® (ppc64le)	Red Hat Enterprise Linux, Ubuntu, SUSE Linux Enterprise Server	IBM Cloud Private 3.2.0 Docker for Linux® on Power® (ppc64le)	icp-docker-18.06.2_ppc64le.bin
Linux on IBM Z and LinuxONE	Red Hat Enterprise Linux, Ubuntu, SUSE Linux Enterprise Server	IBM Cloud Private 3.2.0 Docker for Linux on IBM Z and LinuxONE	icp-docker-18.06.2_s390x.bin

Nota: apenas para o sistema operacional SLES, o arquivo binário `icp-docker-18.06.2` implementa a versão 18.06.1 do Docker.

Sistemas de arquivos e armazenamento suportados

O IBM® Cloud Private suporta vários sistemas de arquivos e tipos de armazenamento.

Requisitos do sistema de arquivos

O IBM Cloud Private requer sistemas de arquivos POSIX. Em clusters de alta disponibilidade (HA), deve-se configurar um sistema de arquivos compatível com POSIX para armazenamento compartilhado. Esse sistema de arquivos deve estar localizado fora de seu cluster do IBM Cloud Private. Para clusters de HA, consulte Clusters do IBM® Cloud Private de alta disponibilidade para obter os requisitos do sistema de arquivos em nós principais e do proxy.

Armazenamento suportado

Para obter mais informações, consulte Guia de armazenamento .

IaaS, Hypervisors e Ambientes Suportados

IaaS Suportado

O IBM Cloud Private é certificado em vários provedores de Infraestrutura como Serviço (IaaS).

Tabela 1. Versões suportadas do IaaS

IaaS	Versões
Azure	
vCenter do VMware	6.0, 6.5, 6.7
OpenStack	Mitaka e posterior
Hyper-V	10.0.15063.0 e mais recente
Nutanix AHV(X86)	20170830.94 (AOS 5.5.2)
Nutanix AHV(IBM PowerPC)	20170331.78 (AOS 5.2.1.1,5.10.0.7) e mais recente
IBM PowerVC	1.4.1 e mais recente

O Kubernetes e o Cloud Foundry demonstraram portabilidade entre vários provedores IaaS na comunidade. Embora o IBM Cloud Private seja certificado para instalações no local no VMware, OpenStack e bare metal, é possível instalá-lo em outros sistemas IaaS. Para obter informações adicionais, entre em contato com seu representante de vendas.

Hypervisors Suportados

O IBM Cloud Private é certificado nos hypervisors KVM, ESX, Nutanix Acropolis, IBM PowerVM e IBM z/VM.

Ambientes Suportados

O IBM Cloud Private pode ser executado em outros ambientes, com outros produtos instalados. Para obter detalhes, consulte Ambientes Suportados .

- Amazon Web Services (AWS)
- Azure
- IBM Cloud
- OpenShift

Portas Obrigatórias

Lista de portas necessárias que devem estar disponíveis para instalação e configuração de um cluster do IBM® Cloud Private.

Você abre as portas antes de iniciar a instalação do IBM Cloud Private e o instalador confirma se elas estão abertas.

Tipos de acesso de porta

- Interno - a porta deve ser aberta para permitir conexões dentro do cluster.
- Externo - a porta deve ser aberta para permitir conexões de fora do cluster.

Se nenhum tipo de acesso for indicado, a porta será usada somente para comunicações internas.

Importante: o IBM Cloud Private suporta um nó de gerenciamento opcional. Se o cluster não incluir um nó de gerenciamento, os componentes que são carregados no nó de gerenciamento serão carregados no nó principal. Deve-se abrir as portas de **Gerenciamento** no nó principal.

Nota: *todos os nós do cluster* se referem aos nós principal, do trabalhador, de proxy, de gerenciamento, etcd e do Vulnerability Advisor (VA). O nó de inicialização não tem requisitos de porta.

- Todos os nós do cluster para todos os nós do cluster
- Todos os nós do cluster para nós principais
- Todos os nós do cluster para nós de gerenciamento
- Todos os nós do cluster para nós do proxy
- Todos os nós do cluster ou nós do etcd para nós do etcd
- Nós principais para nós principais
- Nós principais ou nó do proxy para nós de gerenciamento
- Nós de gerenciamento para todos os nós do cluster
- Nós de gerenciamento para nós principais
- Nós de gerenciamento para nós de gerenciamento
- Nós de gerenciamento para externos
- Nós do proxy para nós de gerenciamento
- Nós externos para proxy

- Nós GlusterFS para todos os nós do cluster
- Portas necessárias para o IBM Multicloud Manager

Todos os nós do cluster para todos os nós do cluster

Porta	Protocolo	Requisito
NA	IPv4	Calico com IP-in-IP (calico_ipip_mode: Always, network_type:calico) **Nota:** Ativado por padrão.
179	TCP	Sempre para o Calico (network_type:calico)
500	TCP e UDP	IPsec (ipsec.enabled: true, calico_ipip_mode: Always, network_type:calico)
4000	TCP	Leitor de medição (management_services.metering: enabled) **Nota:**

para medição externa por meio de medição automática interna ou de proxy. | | 4500 | UDP | IPsec (ipsec.enabled: true)| | 8445| TCP | Exportador de nó (management_services.monitoring: enabled)
Nota: o valor padrão de monitoring_nodeexporter_port. | | 9091 | TCP| Calico (network_type: calico) | | 9099 | TCP| Calico (network_type: calico) | | 9100 | TCP | Exportador de nó (management_services.monitoring: enabled) | |10248-10252| TCP | Sempre para Kubernetes | |30000-32767 | TCP e UDP |Sempre para Kubernetes
Nota: Acesso externo. Essas portas devem ser abertas somente se você configurar o tipo de serviço do Kubernetes para NodePort.|

Todos os nós do cluster para nós principais

Porta	Protocolo	Requisito
8001	TCP	Sempre para o kube_apiserver_port **Nota:** Porta padrão. O

kube_apiserver_port deve estar disponível somente no nó principal. | |8080| TCP | Sempre para o console de gerenciamento
Nota: a porta insegura de ingresso de gerenciamento é igual ao valor padrão de router_http_port. Acesso interno e externo.| |8443| TCP | Sempre para o console de gerenciamento
Nota: a porta insegura de ingresso de gerenciamento é igual ao valor padrão de router_http_port. Acesso interno e externo. | |8500 | TCP | Sempre para o gerenciador de imagem
Nota: acesso interno e externo.| |8600 | TCP | Sempre para o gerenciador de imagem
Nota: acesso interno e externo.| |27017| TCP |MongoDB |

Todos os nós do cluster para nós de gerenciamento

Tabela 3. Todos os nós do cluster para nós de gerenciamento

Porta	Protocolo	Requisito
3000	TCP	Scrape de Prometheus (management_services.metering: enabled) **Nota:** para o Prometheus, extração de dados de medição do metering-dm.
5044	TCP	Logstash ativado (management_services.logging: enabled)
25826	UDP	Serviços principais Collectd exportador (management_services.monitoring: enabled)
31514	TCP	Tiller NodePort **Nota:** acesso interno e externo. A porta padrão 31514 pode ser substituída no arquivo `config.yaml` antes da instalação do IBM Cloud Private.
44134	TCP	Política de rede do Tiller **Nota:** acesso interno e externo.
44135	TCP	Política de rede do Tiller **Nota:** acesso interno e externo.

Todos os nós do cluster para nós do proxy

Tabela 4. Todos os nós do cluster para nós do proxy

Porta	Protocolo	Requisito
31380	TCP	Istio (management_services.istio: enabled) **Nota:** acesso interno e externo.
31390	TCP	Istio (management_services.istio: enabled) **Nota:** acesso interno e externo.

Todos os nós do cluster ou nós do etcd para nós do etcd

Porta	Protocolo	Requisito
2380	TCP	Sempre se o etcd estiver ativado **Note:** nós do etcd para nós do etcd.
4001	TCP	Sempre se o etcd estiver ativado **Nota:** todos os nós do cluster

para nós do etcd. |

Nós principais para nós principais

Tabela 6. Nós principais para nós principais

Porta	Protocolo	Requisito
6969	TCP	Sempre para plataforma-api
9443	TCP	WebSphere ® Application Server Liberty **Nota:** Acesso externo.
31030	TCP	Helm ativado (management_services.service-catalog: enabled)
31031	TCP	Helm ativado (management_services.service-catalog: enabled)
20358	TCP	Sempre para a porta de verificação de funcionamento de plug-in do KMS
6967, 11211	TCP	Sempre para o serviço de verificação de funcionamento

Nós principais ou nó do proxy para nós de gerenciamento

Tabela 7. Nós principais ou nó do proxy para nós de gerenciamento

Porta	Protocolo	Requisito
3000	TCP	Grafana (management_services.monitoring: ativado)
5601	TCP	Kibana (management_services.monitoring: enabled)
9093	TCP	Gerenciador de Alertas (management_services.monitoring: enabled)

Nós de gerenciamento para todos os nós do cluster

Tabela 8. Nós de gerenciamento para todos os nós do cluster

Porta	Protocolo	Requisito
8445	TCP	Exportador de nó de serviços principais (management_services.monitoring: enabled)
9103	TCP	Coletor Collectd (management_services.monitoring: enabled)
9108	TCP	Exportador Elasticsearch (management_services.monitoring: enabled

Nós de gerenciamento para nós principais

Tabela 9. Nós de gerenciamento para nós principais

Porta	Protocolo	Requisito
6969	TCP	Sempre para plataforma-api

Nós de gerenciamento para nós de gerenciamento

Tabela 10. Nós de gerenciamento para nós de gerenciamento

Porta	Protocolo	Requisito
80	TCP	Os serviços principais kube-state-metrics explorer (management_services.monitoring: enabled **Nota:** acesso interno e externo.
389	TCP	LDAP ativado (ldap_enabled: true) **Nota:** acesso interno e externo.
636	TCP	LDAPS ativado (ldap_enabled: true) **Nota:** acesso interno e externo.
3000	TCP	Sempre para plataforma-ui

Porta	Protocolo	Requisito
4000	TCP	Sempre para catálogo-ui
9093	TCP	Gerenciador de alerta de serviços principais (management_services.monitoring: ativado)
9090	TCP	Prometheus (management_services.monitoring: enabled)
9103	TCP	Serviços principais Collectd exportador (management_services.monitoring: enabled)
9108	TCP	Serviços principais Elasticsearch exportador (management_services.monitoring: enabled
9200	TCP	Elasticsearch (management_services.logging: enabled)
9300	TCP	Elasticsearch (management_services.logging: enabled)

Nó de gerenciamento para externo

Tabela 11. Nós de gerenciamento para externos

Porta	Protocolo	Requisito
9004	TCP	Obrigatório se você usar o Key Management Service com o nCipher Hardware Security Module (HSM). A porta é necessária para conexão entre o middleware do nCipher HSM no nó de gerenciamento e o dispositivo nCipher nShield Connect HSM.

Nós do proxy para nós de gerenciamento

Tabela 12. Nós do proxy para nós de gerenciamento

Porta	Protocolo	Requisito
3000	TCP	Serviços principais Grafana (management_services.monitoring: enabled)
3130	TCP	Medição do servidor de interface com o usuário (management_services.metering: enabled)
5601	TCP	Serviços principais Kibana (management_services.logging: enabled)
9093	TCP	Gerenciador de alerta de serviços principais (management_services.monitoring: ativado)
9090	TCP	Serviços principais Prometheus (management_services.monitoring: enabled)
9200	TCP	Serviços principais Elasticsearch (management_services.logging: enabled)
9300	TCP	Serviços principais Elasticsearch (management_services.logging: enabled)

Externos para nós do proxy

Tabela 13. Externos para nós do proxy

Porta	Protocolo	Requisito
80	TCP	Sempre para o serviço Ingresso **Nota:** o valor padrão de ingresso s_http_port.
443	TCP	Sempre para o serviço Ingresso **Nota:** o valor padrão de ingresso s_http_port. Acesso interno e externo.

Nós GlusterFS para todos os nós do cluster

Tabela 14. Nós GlusterFS para todos os nós do cluster

Porta	Protocolo	Requisito
2222	TCP	GlusterFS (management_services.storage-glusterfs: enabled)
24007	TCP	GlusterFS (management_services.storage-glusterfs: enabled)
24008	TCP	GlusterFS (management_services.storage-glusterfs: enabled)
49152:49251	TCP	GlusterFS (management_services.storage-glusterfs: enabled)

Portas Necessárias para IBM Multicloud Manager

Consulte a Visão geral de instalação do IBM Multicloud Manager para conhecer os tópicos de preparação e instalação do IBM Multicloud Manager.

Tabela 1. Tabela de portas do IBM Multicloud Manager

Porta	Requisito
8001	padrão para o cluster gerenciado comunicar-se com a porta do servidor da API do Kubernetes no cluster do hub

Porta	Requisito
8500	padrão para o cluster gerenciado comunicar-se com o registro do Docker do IBM Cloud Private no cluster do hub
443	padrão para que o cluster do hub se comunique com o serviço do Klusterlet no ingresso nginix do IBM Cloud Private

Dimensionando seu cluster do IBM® Cloud Private

Cada cluster do IBM Cloud Private tem suas próprias características. Há diretrizes que fornecem tamanhos de implementação de amostra. Eles têm sido classificados por tamanho e propósito. As considerações são focalizadas em clusters que são implementados em ambientes VMware ou OpenStack.

Nota: os requisitos que são listados não são requisitos mínimos.

Considerações antes de você dimenciar seu cluster

Visualize as considerações a seguir antes de dimensionar seu cluster.

Nós do Trabalhador (cargas de trabalho)

À medida que você determina o número de nós do trabalhador e as configurações de recurso, considere a carga de trabalho que está em execução.

- Se o seu cluster tiver alguns nós do trabalhador, considere aumentar o número de nós do trabalhador enquanto diminui o tamanho dos nós para um espaço de acesso adequado com eficiência, mobilidade e resiliência.
- Acomodar a mobilidade da carga de trabalho.
- Considere a memória que é necessária para um tipo específico de carga de trabalho.
- Considere a memória que é necessária para outras estruturas de aplicativo.
- O máximo de pods por nó é 500 e o máximo de pods por núcleo de CPU é 10.
- O tamanho do cluster depende do número do nó do trabalhador. O número de pods depende do tipo de aplicativo e da configuração do nó do trabalhador.

Nós do proxy

As considerações a seguir são para dimensionamento do nó do proxy. Os nós do proxy podem ser incluídos a qualquer momento.

- Considere o dimensionamento total do recurso versus o número de nós.
- Deve-se editar o controlador de ingresso para corresponder à sua carga de trabalho por meio de seu mapa de configuração.
- Seu endereço IP virtual do proxy faz referência apenas a um único nó por vez.
- (Opcional) Considere um balanceador de carga para difundir a carga de trabalho para seus nós do proxy com o endereço IP externo.
- O Istio também usa o nó do proxy para executar os gateways Ingressos e Egresso.

Nós de gerenciamento

Clusters maiores com mais carga de trabalho requerem nós de gerenciamento maiores. Os nós de gerenciamento podem ser incluídos a qualquer momento se eles foram originalmente externalizados.

Para nós de proxy, considere os requisitos de espaço de acesso para transportar a carga de trabalho devido a uma falha do nó.

Nota: Em um ambiente de Alta Disponibilidade (HA), a memória mínima necessária para permitir redundância e failover pode variar. Isso depende de quais componentes e serviços você escolher instalar e de quantos nós de gerenciamento você possui.

Por exemplo, se você instalar a criação de log mas desativar o monitoramento, dois nós de gerenciamento de 16 GB deixariam espaço não solicitado suficiente. Quando um nó é finalizado, o espaço fica disponível para pods de criação de log de chave executarem failover em nós de gerenciamento restantes.

Se o monitoramento e outros componentes opcionais estiverem ativados, pode não haver memória livre suficiente para permitir o failover. Os pods adicionais consomem memória e também devem ser reiniciados no nó restante. Considere incluir memória ou aumentar o número de nós de gerenciamento.

Considerações de Cluster Grande

Considere os recursos do IBM Cloud Private ao planejar um cluster grande. O IBM Cloud Private possui cargas de trabalho extras de capacidade corporativa além do Vanilla Kubernetes, incluindo mais serviços, como o Calico (MeSH entre nós). Além disso, considere o monitoramento, a criação de logs e a avaliação de vulnerabilidade.

A seguir estão mais considerações:

- A malha entre nós começa a falhar com 700 nós no cluster. Deve-se criar um refletor de roteador para Daemons do BGP.
- Considere usar etcd fora de seus nós principais se você planeja ter um cluster com vários nós do trabalhador. Um cluster etcd separado é ideal para reduzir o impacto sobre o nó principal.
- Certifique-se de implementar o balanceamento de carga em seu nó principal.
- O número de serviços em seu cluster afeta a carga em cada nó. Em clusters grandes com mais de 5.000 serviços, deve-se executar seus nós no modo IP Virtual Server (IPVS).

Nota: o IPVS obtém considerações extras para implementação.

Implementações de Amostra

Visualize as implementações de amostra a seguir de ambientes de tamanhos diferentes.

- Pequeno ambiente do IBM Cloud Private (meio de resiliência)
- Ambiente IBM Cloud Private médio (mídia de resiliência)
- Ambiente do IBM Cloud Private grande (alta resiliência)

Pequeno ambiente do IBM Cloud Private (meio de resiliência)

Tipo de Nó	Número de nós	CPU	Memória (GB)	Disco (GB)
Inicialização	1	2	8	250
Principal	3	16	32	500
Gerenciamento	2	8	16	500
Proxy	2	4	16	400
Trabalhador - cargas de trabalho Java	3 + (Máx: 20)	8	32	400

Para criar um ambiente de teste com esse cluster específico, é possível implementar um único nó principal e diminuir os nós do proxy (resiliência baixa). Para fornecer a maior flexibilidade para seu ambiente, não se deve combinar tipos de nós. Os trabalhadores incluídos são formados para cargas de trabalho Java. Consulte a seção Nós do Trabalhador .

Ambiente do IBM Cloud Private Médio (resiliência média)

Tipo de Nó	Número de nós	CPU	Memória (GB)	Disco (GB)
Inicialização	1	2	8	250
Principal	3	16	32	500
Gerenciamento	2	16	32	500
Proxy	2	4	16	400
Trabalhador - cargas de trabalho Java	3 + (Máx: 20)	8	32	400
Trabalhador	5 + (Máx: 70)	8	32	400
VA	3	6	24	500

Para aumentar o nível de resiliência do cluster, inclua dois nós principais extras.

Ambiente IBM Cloud Private grande (resiliência alta)

Sizing cluster for 500 worker nodes :

Tipo de Nó	N£mero	CPU	Memória (GB)	Disco (GB)
Nó de inicialização	1	4	8	250

Tipo de Nó	N£mero	CPU	Memória (GB)	Disco (GB)
Nó principal	3	16	128	500
Nó de gerenciamento	2	16	128	500
Nó do proxy	2	4	16	256
Nó do VA	3	6	48	500
Nó do trabalhador	500	8	32	400

Sizing cluster for 1000 worker nodes :

Tipo de Nó	Número de nós	CPU	Memória (GB)	Disco (GB)
Inicialização	1	4	8	250
Principal	3	16	128	500
Gerenciamento	5	16	128	500
Proxy	2	4	16	256
VA	3	6	48	500
Trabalhador	1000	8	32	400

Para aumentar o nível de resiliência do cluster, deve-se implementar e gerenciar cargas de trabalho entre múltiplos clusters do IBM Cloud Private.

Preparando para proteger seu cluster

Ao instalar o IBM® Cloud Private, você cria uma conexão segura a partir do nó de inicialização para todos os outros nós em seu cluster. É possível configurar o SSH ou configurar a autenticação de senha em seu cluster. Para obter as etapas de instalação, consulte Instalando as edições Cloud Native, Enterprise e Community do IBM Cloud Private.

Após a instalação, é possível proteger o acesso ao seu cluster por meio do controle de acesso baseado na função (RBAC), da conexão única (SSO) e do Lightweight Directory Access Protocol (LDAP). É possível gerenciar repositórios Helm e criar namespaces, equipes e políticas de segurança para pods e contêineres. Para obter informações adicionais, consulte o Guia de segurança.

Considerações de Segurança

À medida que projeta sua segurança para o IBM Cloud Private, você deve entender as seguintes informações:

- O tipo de cargas de trabalho que serão executadas.
- O tipo de interações externas necessárias para os aplicativos.
- A maneira de interagir com a plataforma e as interfaces que serão utilizadas.
- As diferentes integrações para o Gerenciamento de Identidade e de Acesso (IAM) e outros componentes de segurança.
- O isolamento e a segmentação que são necessários para os diferentes grupos e equipes.
- As necessidades de proteção de dados para as diferentes cargas de trabalho.
- A maneira como os contêineres precisam ser protegidos e monitorados para mutações.

Interagindo com o IBM Cloud Private

Entender quem e como as pessoas interagem com o cluster ajuda-o a configurar o IAM. Para determinar o acesso e as autorizações, você deve entender o modelo de operações de destino para os clusters que estão sendo usados. Pergunte as seguintes questões:

- Quem irá fazer o gerenciamento de cluster?
- Quem está implementando e gerenciando os aplicativos?
- Quais operações são necessárias para as equipes?
- É uma abordagem de diversos locatários que inclui a criação de log e o monitoramento necessários?
- Quais integrações de autenticação são necessárias?

Preste atenção especial aos diferentes pontos de integração, como a integração contínua e a entrega contínua (CICD), uma vez que esses pontos determinam as diferentes contas de serviço e as autorizações de que as contas precisam.

Quando usar uma ferramenta de implementação CICD com permissões para implementar contêineres, deve-se limitar o acesso e implementações com uma conta do serviço e não permitir que nenhum outro usuário crie essas implementações.

Para obter informações adicionais, consulte Configurando a conexão LDAP e Isolamento de pod no Guia de segurança.

Manipulando certificados

Os certificados são usados para as seguintes interações:

- Comunicações intrasserviço:
  - Serviços de plataforma, como criação de log, monitoramento, medição e segurança.
  - Serviços principais, como Kubernetes e etcd.
- Terminais externos, como o gerenciador de imagem, o registro do Docker, o ingresso de gerenciamento, o Liberty e o Helm.
  - São suportados os certificados integrados (autoassinados) e fornecidos pelo usuário.
  - Usado por cargas de trabalho que estão em execução na plataforma IBM Cloud Private (ingresso de proxy)

O ciclo de vida do certificado é gerenciado manualmente ou usando o serviço Gerenciador de certificados (cert-manager) do IBM Cloud Private:

- O serviço cert-manager é usado para gerar e gerenciar certificados e inclui renovação automática.
- Baseado no projeto da comunidade do Kubernetes cert-manager.
- Tipos de emissor que são suportados: Autoridade de certificação (CA), autoassinado e servidor HashiCorp Vault.

Para obter informações adicionais, consulte os seguintes tópicos:

- Usando o Gerenciador de certificados (cert-manager) do IBM Cloud Private
- Criando certificados do Gerenciador de certificados (cert-manager) do IBM Cloud Private

Criptografia de dados

Dados em trânsito:

- O TLS e o IPSec são usados para fornecer proteção de dados em trânsito.
- O controlador de ingresso de gerenciamento exporta o TLS, que pode ser usado por APIs que usam o TLS como um front-end.
- Todo o tráfego de dados entre nós pode ser prontamente criptografado usando o IPSec, sem mudar nenhum aplicativo. Para obter detalhes, consulte Criptografando o tráfego de rede de dados do cluster com o IPsec.
- O TLS e o IPSec podem ser configurados para usar cifras em conformidade com o Federal Information Processing Standard (FIPS).

Dados inativos:

- Qualquer estado do IBM Cloud Private pode ser protegido usando uma criptografia em nível de sistema de arquivos ou em nível de dispositivo de bloco, que é descrita em Criptografando volumes usando dm-crypt.
- As cifras em conformidade com o FIPS podem ser usadas.
- O provedor de criptografia AES-CBC do Kubernetes pode ser ativado para criptografar segredos. Para obter detalhes, consulte o tópico do Kubernetes Criptografando dados secretos inativos.
- Todos os segredos do IBM Cloud Private são acessíveis somente aos usuários que têm a função de administrador de cluster ou de administrador de equipe.

É possível criptografar os sistemas de arquivos que são usados pelo IBM Cloud Private com a criptografia Linux® Unified Key Setup (LUKS) no Linux. Assegure-se de que seu sistema tenha espaço em disco disponível. O `dm-crypt` fornece criptografia transparente de dispositivos de bloco. É possível acessar os dados imediatamente após montar o dispositivo. Por padrão, a criptografia de dados em trânsito é desativada no cluster do IBM Cloud Private.

Para obter informações adicionais, consulte os seguintes tópicos:

- Ativando o FIPS em sistemas operacionais que usam o IBM Cloud Private
- Criptografando segredos do Kubernetes com o plug-in Key Management Service
- Gerenciando segredos

Protegendo suas imagens

O IBM Cloud Private tem um repositório de imagem privada que é baseado no registro do Docker no cluster. A lista a seguir destaca alguns dos benefícios de usar o IBM Cloud Private para proteger suas imagens:

- Imagens em pacote configurável: É possível importar imagens do Docker do pacote configurável para o registro privado ou importar qualquer imagem do Docker que você deseja implementar em seus nós.

- Acesso seguro: Inclua somente as imagens que você aprova, para que seus Desenvolvedores tenham imagens confiáveis e validadas a partir das quais fazer a construção.
- Local do repositório de imagem: Todas as imagens locais de registro do Docker estão localizadas no nó principal. Se houver vários nós principais, o registro será distribuído entre os nós principais no modo ativo.
- Repositórios externos: o Vulnerability Advisor não pode varrer os repositórios externos.

É possível restringir a partir de qual repositório as imagens podem ser implementadas. Também é possível cumprir políticas do Vulnerability Advisor. Se uma imagem não atender aos requisitos de política definidos, o pod não será implementado. Se você usar o Vulnerability Advisor, o repositório de imagem interno precisará ser usado.

Protegendo seus contêineres

Ao proteger seus contêineres, o objetivo principal é fornecer visibilidade, controle e analítica para acessar e cumprir a segurança e conformidade em seus aplicativos e dados que estão em execução na nuvem privada. É possível atingir esses objetivos com o Vulnerability Advisor e o Mutation Advisor.

Benefícios do Vulnerability Advisor:

- Os desenvolvedores podem projetar aplicativos seguros com pouco esforço
- Nenhuma configuração necessária
- Não são requeridos agentes ou credenciais guest pelo xSP
- Inviolável
- Suporte de avaliação de vulnerabilidade de imagens quase em tempo real
- Simples para interpretar resultados

Seus contêineres são imutáveis?

Com o IBM Cloud Private, é possível rastrear mudanças de um contêiner em arquivos e processos. O monitoramento de integridade do sistema é uma parte importante de vários requisitos de conformidade e auditoria, incluindo o Payment Card Industry Data Security Standard (PCI/DSS).

O Mutation Advisor monitora continuamente os contêineres para o estado de arquivos e processos em uma determinada amostragem. Ele relata mudanças modulares no estado na interface com o usuário do Mutation Advisor que está na lista de desbloqueio do perfil, como mudanças normais para o contêiner. Os relatórios podem ser visualizados como notificações de mutações em uma base contêiner por contêiner, e uma linha de tempo para cada contêiner.

Para obter mais informações, consulte Vulnerability Advisor .

Terminais do IBM Cloud Private

Um terminal é um endereço de destino de rede que é exposto por recursos do Kubernetes, como serviços e ingressos. As seções a seguir descrevem os terminais externos e internos disponíveis em um cluster do IBM Cloud Private.

Consulte os terminais que são criados no cluster do IBM® Cloud Private.

- Terminais externos
  - Terminal principal
  - Terminal de proxy
  - Terminal NodePort
- Terminais internos

Terminais Externos

Os terminais Principal e Proxy são os terminais externos usados para o acesso de fora do cluster. Você define esses terminais no `config.yaml` durante a instalação. Geralmente, você cria um nome completo do domínio (FQDN), que é uma entrada DNS e um certificado assinado por CA, e o aplica ao nó principal do IBM Cloud Private. O nó principal aplica o FQDN a todos os terminais principais.

Os terminais externos podem ser definidos de uma das maneiras a seguir:

- Terminal de nó único: o endereço IP ou FQDN de um nó único
- Terminais de AD que usam um endereço IP virtual (VIP): o endereço VIP ou FQDN de vários nós de AD
- Terminais de AD que usam um balanceador de carga do cluster: o endereço IP do balanceador de carga ou FQDN de vários nós de AD

Terminal principal

Todas as APIs de plataforma são acessadas por meio do nó principal ou nós diretamente por meio do ingresso de gerenciamento ou por meio do balanceador de carga de gerenciamento.

A seguir está o formato da URL para acessar o terminal principal:

```
https://<Cluster Master Host>:<Cluster Master API Port>/<API path>
```

Em que:

- O **Host Mestre do Cluster** é um dos valores a seguir:
  - O valor `cluster_CA_domain` no arquivo `config.yaml`.
  - O valor `cluster_ca_domain` no ConfigMap `ibmcloud-cluster-info`.
  - O valor `cluster_lb_address` no arquivo `config.yaml`.
  - O valor `cluster_address` no ConfigMap `ibmcloud-cluster-info`.
- A **Porta da API do Cluster Mestre** é um dos valores a seguir:
  - O valor `router_http_port` no arquivo `config.yaml`.
  - O valor `cluster_router_http_port` no ConfigMap `ibmcloud-cluster-info`.
  - O valor `router_https_port` no arquivo `config.yaml`.
  - O valor `cluster_router_https_port` no ConfigMap `ibmcloud-cluster-info`.
- A **Porta da API do Kubernetes** é um dos valores a seguir:
  - Qualquer valor `<Cluster Master API Port>`.
  - O `cluster_kube_apiserver_port` no ConfigMap `ibmcloud-cluster-info` .
  - O arquivo `kube_apiserver_secure_port` no arquivo `config.yaml` .

Terminal de proxy

O terminal do proxy é um ou mais proxies de ingresso que são expostos por cargas de trabalho que são implementadas no IBM Cloud Private por meio do recurso de ingresso.

A seguir está o formato da URL:

```
https://<Cluster Proxy Host>:<Cluster Proxy API Port>/<API path>
```

Em que:

- **Host do Proxy do Cluster** é um ou mais dos valores a seguir:
  - O valor de endereço IP `proxy_node` no arquivo `hosts`.
  - O valor `proxy_vip` no arquivo `config.yaml`.
  - O valor `proxy_lb_address` no arquivo `config.yaml`.
  - O valor `proxy_address` no ConfigMap `ibmcloud-cluster-info`.
  - Um nome de domínio completo customizado.
- A **Porta da API do Proxy do Cluster** é um dos valores a seguir:
  - O valor `ingress_https_port` do TLS no arquivo `config.yaml`.
  - O valor `proxy_ingress_https_port` do TLS no ConfigMap `ibmcloud-cluster-info`.
  - O valor `ingress_http_port` não seguro no arquivo `config.yaml`.
  - O valor `proxy_ingress_http_port` não seguro no ConfigMap `ibmcloud-cluster-info`.

Terminal NodePort

Cargas de trabalho podem definir serviços que são expostos como NodePorts. Se um serviço usar o tipo NodePort, ele efetuará bypass no terminal do proxy.

Terminais Internos

Seu cluster do IBM Cloud Private tem uma rede interna para cargas de trabalho. Os serviços devem se comunicar com as cargas de trabalho na rede de cluster interna.

Os serviços que precisam se comunicar dentro do cluster para serviços de plataforma fazem isso usando o serviço de ingresso de gerenciamento interno na rede de cluster interna, a menos que seja especificado de outra forma pela documentação da API de serviço.

O terminal para acessar serviços de plataforma é `https://icp-management-ingress.kube-system: 8443`. Esse terminal é o terminal interno para o ingresso de gerenciamento e está disponível a partir de todos os namespaces.

Para outros serviços, a seguir estão os formatos para acessar o serviço usando o nome do serviço no cluster local:

- Se o serviço estiver no mesmo namespace, o formato será `https://<service-name>:8443`.
- Se o serviço estiver em um namespace diferente, o formato será `https://<service-name>.<namespace-name>:8443`.

Para obter informações adicionais, consulte os seguintes artigos:

- DNS para Serviços e Pods
- DNS

ConfigMap de configuração de cluster

Informações de configuração sobre o cluster do IBM® Cloud Private.

O ConfigMap de `ibmcloud-cluster-info` no namespace do `kube-public` foi projetado para publicar metadados de cluster. O ConfigMap `ibmcloud-cluster-info` no namespace `kube-público` publica os metadados do cluster do IBM Cloud Private. A seguir está a estrutura ConfigMap:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: ibmcloud-cluster-info
  namespace: kube-public
data:
  edition: "{{ edition }}"
  cluster_name: "{{ clustername }}"
  cluster_ca_domain: "{ }"
  cluster_address: "{ }"
  cluster_router_http_port: "{ }"
  cluster_router_https_port: "{ }"
  cluster_kube_apiserver_port: "{ }"
  proxy_address: "{ }"
  proxy_ingress_http_port: "{ }"
  proxy_ingress_https_port: "{ }"
```

A tabela a seguir descreve os parâmetros:

Tabela 1. ibmcloud-cluster-info ConfigMap parâmetros

Parâmetro	Valor	Descrição	Uso de Parâmetro
edição	edição	IBM Cloud Private imagem	Mostra a versão do IBM® Cloud Private .
cluster_name	cluster_name	Consulte `cluster_name`.	O nome do cluster.
cluster_ca_domain	cluster_CA_domain	Consulte `cluster_CA_domain` .	Acesso externo ao ingresso de gerenciamento.
cluster_address	cluster_external_address	Consulte `cluster_lb_address` .	Endereço IP externo para os serviços de gerenciamento em seu cluster.
cluster_router_http_port	router_http_port	Consulte `router_http_port` .	A porta HTTP que é usada pelo ingresso de gerenciamento do IBM Cloud Private.
cluster_router_https_port	router_https_port	Consulte `router_https_port` .	A porta HTTPS que é usada pelo ingresso de gerenciamento do IBM Cloud Private.
cluster_kube_apiserver_port	kube_apiserver_secure_port	Consulte `kube_apiserver_secure_port` .	Configura a porta segura do apiserver do Kubernetes.
proxy*proxy*	proxy_external_address	Consulte o nó do proxy, proxy_vip ou proxy_lb_address em `Customizando o cluster com o arquivo config.yaml` e `Configurando as funções do nó nos arquivos host`.	O endereço que o controlador de ingresso usa.
proxy_ingress_http_port	entradas_http_port	Consulte `ingresso: http_port` .	Porta HTTP usada pelo controlador de ingresso NGINX.

Parâmetro	Valor	Descrição	Uso de Parâmetro
proxy_ingress_https_port	entradas_https_port	Consulte `ingresso: https_port`.	Porta HTTPS que é usada pelo controlador de ingresso NGINX.

É possível usar o ConfigMap para obter informações do cluster. Considere o seguinte exemplo:

```
cluster_address: 9.21.58.100
cluster_ca_domain: mycluster.icp
cluster_kube_apiserver_port: "8001"
cluster_router_http_port: "8080"
cluster_router_https_port: "8443"
edition: Enterprise Edition
proxy_address: 9.21.58.101
proxy_ingress_http_port: "80"
proxy_ingress_https_port: "443"
```

É possível obter as informações de ConfigMap executando o comando a seguir:

```
kubectl -n kube-public get configmap ibmcloud-cluster-info -o yaml
```

Instalação e validação

Esta seção fornece todos os detalhes para instalar e validar sua instalação do IBM® Cloud Private.

Para um fluxo de trabalho de exemplo para um cenário de múltiplos clusters, consulte Visão geral de configuração do IBM Multicloud Manager.

- Instalando o IBM Cloud Private
- Acessando seu cluster

Instalação

É possível instalar o IBM Cloud Private ou o IBM® Cloud Private-CE.

Há um instalador para o IBM Cloud Private, que configura um ambiente de produção de cluster de um ou múltiplos nós. Independente do pacote configurável selecionado, seja o Cloud Native ou o Enterprise, você segue o mesmo processo de instalação do IBM Cloud Private. É possível instalar o IBM Cloud Private nas configurações padrão ou de alta disponibilidade. Após a instalação, é possível incluir componentes extras. Para obter mais informações, consulte pacotes configuráveis do IBM Cloud Private.

A instalação do IBM® Cloud Private-CE configura um cluster único ou com vários nós apenas para propósitos de teste. Essa edição é uma opção sem encargo. Esta edição não suporta alta disponibilidade de gerenciamento. Além disso, os serviços com recursos, tais como o Cloud Foundry, o Cloud Automation Manager e o Vulnerability Advisor, não estão disponíveis. Não é possível fazer upgrade da Community Edition para as edições licenciadas do IBM Cloud Private.

- Preparando seu cluster para instalação
- Instalando as edições Cloud Native, Enterprise e Community do IBM Cloud Private
- Instalando o software IBM no IBM Cloud Private
- Opções de configuração durante a instalação

Preparando seu cluster para instalação

Revise as opções de instalação e, em seguida, instale e configure um cluster do IBM® Cloud Private.

- Preparando nós
- Configurando o Docker para o IBM Cloud Private
- Especificando um diretório de armazenamento do Docker padrão usando montagem de ligação
- Especificando outros diretórios de armazenamento padrão usando montagem de ligação
- Instalação por trás de um proxy HTTP
- Isolando ambientes de rede e de cálculo

Se você deseja ativar o IBM Multicloud Manager, consulte Preparando-se para a instalação do IBM Multicloud Manager.

Preparando nós

Antes de instalar o IBM® Cloud Private, deve-se configurar um cluster de nós do servidor.

- Prepare seu cluster para instalação
- Prepare cada nó para instalação

Prepare seu cluster para instalação

Antes de preparar seus nós para instalar o IBM Cloud Private, deve-se tomar algumas decisões sobre o cluster.

1. Revise os requisitos do sistema. Para obter mais informações sobre requisitos de software e hardware, consulte Requisitos do sistema.
2. Determine sua arquitetura de cluster e obtenha o endereço IP para todos os nós em seu cluster. Para obter mais informações sobre tipos de nó, consulte Arquitetura. Durante a instalação, você especifica os endereços IP para cada tipo de nó. Lembre-se de que depois de instalar o IBM Cloud Private, é possível incluir ou remover nós do trabalhador, proxy ou de gerenciamento de seu cluster.

3. Seu ambiente pode incluir nós com nomes de dispositivos de rede diferentes, como `enX` para RHEL ou `netX` para Ubuntu. Para configurações do Calico, certifique-se de usar `interface=<REGEX>` nas configurações de rede, conforme mostrado no exemplo a seguir.

   ```
   interface="en.*, net.*, eth.*"
   ```

   Nota: não é possível mudar as configurações de rede do Calico após a configuração inicial sem sofrer uma interrupção de rede. Deve-se planejar as configurações antes da configuração inicial de seu cluster do IBM Cloud Private.

Prepare cada nó para instalação

1. Assegure-se de que todas as portas padrão estejam abertas, mas não estejam em uso. Nenhuma regra de firewall deve bloquear essas portas. Durante a instalação, o instalador também confirma que essas portas estão abertas. Para obter mais informações sobre as portas padrão do IBM Cloud Private, consulte Portas padrão.

   Para verificar manualmente se uma porta está aberta e disponível, é possível executar um dos dois comandos a seguir, onde `port_numbers` representa a porta TCP/UDP ou as portas a serem verificadas:

   - Execute o `ss` comando:

     ```
     ss -tnlp | awk '{print $4}'| egrep -w "<port_numbers>"
     ```

     Se a porta não está em uso, a saída está vazia. Se a porta está em uso, a saída é exibida como no exemplo a seguir:

     ```
     # ss -tnlp | awk '{print $4}' | egrep -w "8001|8500|3306"
     :::8001 :::3306 :::8500
     ```

   - Ou, se você instalou os utilitários de rede, execute o comando `netstat`:

     ```
     netstat -tnlp | awk '{print $4}' | egrep -w "<port_numbers>"
     ```

     Se a porta está em uso, a saída é exibida como no exemplo a seguir:

     ```
     # netstat -tnlp | awk '{print $4}' | egrep -w "8001|8500|3306"
     :::8001 :::3306 :::8500
     ```

     Os números de porta devem ser separados com o caractere `|`.

2. Certifique-se de que todos os diretórios no nó estejam vazios e tenham espaço para a instalação. Para obter informações adicionais, consulte Requisitos e recomendações de hardware.

3. Configure o arquivo `/etc/hosts` em cada nó em seu cluster.

   1. Inclua os endereços IP e nomes de host para todos os nós no arquivo `/etc/hosts` em cada nó.
      - **Importante**: certifique-se de que o nome do host seja listado pelo endereço IP para o host local. Não é possível listar o nome do host pelo endereço de loopback, `127.0.0.1`.
      - Os nomes do host no arquivo `/etc/hosts` não podem conter letras maiúsculas.

- Se seu cluster contiver um único nó, será necessário listar o endereço IP e o nome do host.

2. Comente a linha do arquivo que inicia com `127.0.1.1` e `::1 localhost`.

   O arquivo `/etc/hosts` para um cluster que contém um nó principal, um nó do proxy e dois nós do trabalhador é semelhante ao código a seguir:

   ```
    127.0.0.1       localhost
   # 127.0.1.1     <host_name>
   # The following lines are desirable for IPv6 capable hosts
   #::1     localhost ip6-localhost ip6-loopback
   ff02::1 ip6-allnodes
   ff02::2 ip6-allrouters
   <master_node_IP_address> <master_node_host_name>
   <worker_node_1_IP_address> <worker_node_1_host_name>
   <worker_node_2_IP_address> <worker_node_2_IP_host_name>
   <proxy_node_IP_address> <proxy_node_host_name>
   ```

   Nota: enquanto o processo de instalação do IBM Cloud Private estiver em execução, o arquivo `/etc/hosts` em todos os nós do cluster será atualizado automaticamente para incluir uma entrada para o `clusterName.icp`. Isso é correlacionado ao `cluster_vip`, a menos que `cluster_vip` não esteja configurado, nesse caso, ele é correlacionado a `cluster_lb_address`. O IBM Cloud Private usa o `clustername.icp:8500/xx/xxx` no arquivo `/etc/hosts` para extrair as imagens do Docker do registro do IBM Cloud Private.

4. Em cada nó do cluster, você deve configurar um gateway padrão ou uma rota para o `service_cluster_ip_range`.

   Por exemplo, se você deseja configurar uma rota para o IPv4 padrão `service_cluster_ip_range`, execute o comando a seguir:

   ```
   ip route add 10.0.0.0/16 dev eth0
   ```

   Em que `eth0` é a interface Ethernet que é designada ao seu endereço IP público. Para obter informações adicionais sobre `service_cluster_ip_range`, consulte Configurações de rede.

5. Para ambientes do OpenStack, se o `/etc/hosts` for gerenciado pelo serviço cloud-init, será necessário evitar que o serviço cloud-init modifique o arquivo `/etc/hosts`. No arquivo `/etc/cloud/cloud.cfg`, assegure-se de que o parâmetro `manage_etc_hosts` esteja configurado como `false`:

   ```
   manage_etc_hosts: false
   ```

6. Assegure a conectividade de rede entre todos os nós em seu cluster. Confirme se cada nó está conectado a todos os outros nós no cluster.

7. Em cada nó em seu cluster, confirme se uma versão suportada do Python está instalada. O Python 2 (versões 2.6 ou 2.7) e o Python 3 (versão 3.5 ou mais recente) são suportados.

   ```
   python --version
   ```

   Nota: se o Python 3 ou mais recente for usado, o local do interpretador Python deverá ser configurado no arquivo config.yaml inserindo a linha a seguir:

   ```
   ansible_python_interpreter: /usr/bin/python3
   ```

8. Sincronize os clocks em cada nó no cluster. Para sincronizar os seus clocks, é possível usar o protocolo de tempo de rede (NTP). Para obter mais informações sobre como configurar o NTP, consulte a documentação do usuário para o seu sistema operacional.

9. Assegure-se de que um cliente SSH esteja instalado em cada nó.

O que fazer a seguir

Instale seu cluster, consulte Instalando o IBM Cloud Private.

Configurando o Docker para o IBM Cloud Private

O IBM Cloud Private requer o Docker. Deve-se instalar manualmente o Docker no nó de inicialização. É possível instalar manualmente o Docker no restante de seus nós do cluster ou o instalador pode instalar automaticamente o Docker em seus nós de gerenciamento principais, do trabalhador, de proxy e opcionais e nós do Vulnerability Advisor (VA) configurados corretamente.

Deve-se instalar manualmente uma versão do Docker que seja suportada pelo IBM Cloud Private em seu nó de inicialização.

Também é possível instalar o Docker manualmente em todos os seus nós do cluster ou você pode permitir que o instalador do IBM Cloud Private configure o Docker em seus nós do cluster. Se desejar que o instalador configure o Docker em seus nós do cluster, será possível definir essa configuração durante a instalação de seu cluster. Veja Configurando os nós do cluster para a instalação automática do Docker.

Para obter uma lista de versões do Docker que são suportadas pelo IBM Cloud Private, consulte Versões do Docker suportadas.

Instalando o Docker manualmente

É possível instalar o pacote do Docker do IBM Cloud Private fornecido ou instalar uma versão suportada do Docker do IBM Cloud Private por meio do website do Docker.

Instalando o Docker manualmente usando o pacote do Docker do IBM Cloud Private fornecido

Conclua as etapas a seguir em cada nó em que você deseja instalar manualmente o Docker.

1. Em seu nó, assegure-se de que seu gerenciador de pacotes esteja configurado para permitir atualizações do pacote. Os gerenciadores de pacotes incluem `RPM` para Red Hat Enterprise Linux (RHEL) e `Apt` para Ubuntu.
2. Faça download do pacote do Docker para seu nó. Consulte IBM Cloud Private Pacotes do Docker.
3. Instale o Docker.
   - Para o Linux®, execute este comando:

     ```
     chmod +x icp-docker-18.06.2_x86_64.bin
     sudo ./icp-docker-18.06.2_x86_64.bin --install
     ```

   - Para o Linux® on Power® (ppc64le), execute este comando:

     ```
     chmod +x icp-docker-18.06.2_ppc64le.bin
     sudo ./icp-docker-18.06.2_ppc64le.bin --install
     ```

   - Para o Linux® on IBM® Z and LinuxONE, execute este comando:

     ```
     chmod +x icp-docker-18.06.2_s390x.bin
     sudo ./icp-docker-18.06.2_s390x.bin --install
     ```

     Nota: para desinstalar este pacote do Docker, substitua a opção `--install` por `--uninstall` no comando.
4. Verifique sua instalação; consulte Verificando sua instalação.

Instalando o Docker manualmente por meio do website do Docker

Conclua as etapas a seguir em cada nó em que você deseja instalar manualmente o Docker.

1. Instale o Docker. **Nota:** talvez seja necessário registrar-se para fazer download do pacote Docker.
   - Para Ubuntu, consulte a Documentação do Docker.
   - Para nós do SLES, é possível instalar manualmente o Docker usando as instruções de instalação do Docker na documentação do SLES ou usando o pacote do Docker IBM Cloud Private fornecido.
2. Verifique sua instalação; consulte Verificando sua instalação.

Verificando Sua Instalação

1. Assegure-se de que o mecanismo de Docker esteja iniciado. Execute o comando a seguir:

   ```
   sudo systemctl start docker
   sudo systemctl status docker
   ```

2. Configure seu mecanismo de Docker, consulte Configurando seu mecanismo de Docker.

Configurando o mecanismo de Docker

1. Se você deseja mudar o local do diretório de armazenamento padrão do Docker, deve-se configurar uma montagem bind para o novo diretório antes de instalar o IBM Cloud Private. Veja Especificando um diretório de armazenamento padrão do Docker para o Docker instalado manualmente.

2. Configure a rotação do log do Docker. Isso reduz os problemas de disco que são causados pela retenção de informações de log em excesso. Para configurar uma rotação de log, conclua as etapas a seguir:

    1. Configure a rotação de log usando o arquivo `/lib/systemd/system/docker.service`. Para o parâmetro ExecStart, inclua a seguinte opção:

        ```
        -- log-opt max-size=10m -- log-opt max-file=10
        ```

    2. Recarregue e reinicie o Docker:

        ```
        sudo systemctl daemon-reload
        sudo systemctl restart docker
        ```

3. Para visualizar os logs de contêineres e serviços do Docker na console de gerenciamento do IBM Cloud Private, deve-se configurar o driver de criação de log padrão para `json-file`.

    1. Localize o driver de criação de log padrão para o daemon do Docker:

        ```
        sudo docker info|grep "Logging Driver"
        ```

        A saída se assemelha ao código a seguir:

        ```
        Logging Driver: journald
        ```

    2. Configure o driver de criação de log do Docker para `json-file`. Consulte a documentação do Docker.

    3. Verifique se o driver de criação de log padrão foi atualizado para `json-file`:

        ```
        sudo systemctl daemon-reload
        sudo systemctl start docker
        sudo docker info|grep "Logging Driver"
        ```

        A saída se assemelha ao código a seguir:

        ```
        Logging Driver: json-file
        ```

4. Se você estiver usando um arquivo `docker.service` customizado, assegure que o parâmetro **MountFlags** esteja configurado para compartilhamento ou remova o parâmetro MountFlags do arquivo `docker.service`. Para obter informações adicionais sobre como criar manualmente o arquivo de unidade `systemd`, `docker.service`, consulte a documentação do Docker.
5. Para sistemas RHEL, revise as configurações do driver de armazenamento. Consulte Arquiteturas e drivers de armazenamento.

Especificando um diretório de armazenamento do Docker padrão usando montagem bind

Se você deseja usar um diretório de armazenamento do Docker padrão diferente, deve-se mudá-lo antes de instalar o IBM® Cloud Private usando uma montagem bind.

O IBM Cloud Private requer o Docker. É possível instalar o Docker em cada nó ou configurar seus nós para que o processo de instalação do IBM Cloud Private inclua a instalação do Docker. Consulte Instalando o Docker no IBM Cloud Private.

- Especificando um diretório de armazenamento padrão do Docker para o Docker instalado manualmente
- Especificando um diretório de armazenamento padrão do Docker para o Docker instalado automaticamente

Especificando um diretório de armazenamento padrão do Docker para o Docker instalado manualmente

Se você instalou o Docker em seus nós, execute as etapas a seguir:

1. Remova todos os contêineres e imagens do Docker.

   ```
   sudo docker rm -f $(docker ps -aq); docker rmi -f $(docker images -q)
   ```

2. Pare o serviço do Docker.

   ```
   sudo systemctl parar docker
   ```

3. Remova o diretório de armazenamento do Docker.

   ```
   sudo rm -rf /var/lib/docker
   ```

4. Crie um novo diretório de armazenamento `/var/lib/docker`.

   ```
   sudo mkdir /var/lib/docker
   ```

   Nota: um diretório `/var/lib/docker` com menos de 50 GB de espaço em disco não é suportado.

5. Use a montagem bind para configurar o novo local. Por exemplo, para configurar o novo local como `/mnt/docker` execute os comandos a seguir:

   ```
   sudo mkdir /mnt/docker sudo mount --rbind /mnt/docker /var/lib/docker
   ```

6. Inicie o serviço do Docker.

   ```
   sudo systemctl start docker
   ```

Especificando um diretório de armazenamento padrão do Docker para o Docker instalado automaticamente

Se o processo de instalação do IBM Cloud Private inclui a instalação do Docker, execute as etapas a seguir:

1. Crie um novo diretório de armazenamento `/var/lib/docker`.

   ```
   sudo mkdir /var/lib/docker
   ```

2. Use a montagem bind para configurar o novo local. Por exemplo, para configurar o novo local como `/mnt/docker` execute os comandos a seguir:

   ```
   sudo mkdir /mnt/docker sudo mount --rbind /mnt/docker /var/lib/docker
   ```

Especificando outros diretórios de armazenamento padrão usando a montagem bind

Se você deseja usar diferentes diretórios de armazenamento padrão para os serviços principais do IBM® Cloud Private, será necessário mudá-los antes de instalar o IBM Cloud Private usando uma montagem bind.

Para evitar problemas de espaço em disco, monte os diretórios de armazenamento padrão em caminhos separados que tenham capacidades do disco maiores.

É possível usar uma montagem bind para montar os diretórios a seguir:

- Etcd - `/var/lib/etcd`
- Registro de imagem privado - `/var/lib/registry`
- Serviços de gerenciamento - `/opt/ibm/cfc`
- VA - `/var/lib/icp`
- Serviço Kubelet - `/var/lib/kubelet`

Para configurar um novo local padrão, execute os comandos a seguir:

1. Crie um novo diretório de armazenamento.

   ```
   mkdir <new storage directory name>
   ```

2. Use a montagem bind para configurar o novo local.

   ```
   mount --rbind <new storage directory> <old storage directory>
   ```

3. Persista essa mudança em reinicializações do sistema.

```
echo "<new storage directory> <old storage directory> none defaults,bind 0 0" >> /etc/fstab
```

Por exemplo, para configurar o novo local como `/mnt/etcd` para o diretório `/var/lib/etcd`, você executaria os comandos a seguir:

```
mkdir /mnt/etcd
mount --rbind /mnt/etcd /var/lib/etcd
echo "/mnt/etcd /var/lib/etcd none defaults,bind 0 0" >> /etc/fstab
```

IBM Cloud Private instalação atrás de um proxy HTTP

O IBM Cloud Private requer o Docker. Deve-se instalar manualmente o Docker no nó de inicialização. É possível instalar manualmente o Docker no restante de seus nós do cluster ou o instalador pode instalar o Docker automaticamente.

- Instalação manual do Docker

  Se você instalou o Docker manualmente, siga as etapas para instalar o IBM Cloud Private por trás de um proxy HTTP.

  1. Crie a pasta `docker.service.d/` . Em todos os nós (nós de inicialização, gerenciamento, proxy, trabalho, VA e principais), execute os seguintes comandos:

     ```
     sudo mkdir -p /etc/systemd/system/docker.service.d
     ```

  2. Crie o arquivo `docker.service.d/http-proxy.conf` e inclua as variáveis a seguir: `HTTP_PROXY`, `HTTPS_PROXY` e `NO_PROXY`.

     ```
     sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf
     [Service]
     Environment="HTTP_PROXY=http://1.2.3.4:3128" "HTTPS_PROXY=http://1.2.3.4:3128"
     "NO_PROXY=localhost,127.0.0.1,<cluster_CA_domain>.icp,<ICP ip address/range>"
     ```

     Nota: A entrada `NO_PROXY` indica que nenhum proxy deve ser usado para o registro privado do Docker do IBM Cloud Private. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação. Mude `<ICPipaddress/range>` para o intervalo de endereço IP de nós do ICP, por exemplo, 192.168.1.0/24. Isso é para certificar-se de que o Docker não utilize o proxy para comunicações entre Docker.

  3. Reinicie o Docker utilizando os seguintes comandos:

     ```
     sudo systemctl daemon-reload
      sudo systemctl restart docker
     ```

  4. Customize o arquivo `config.yaml` do IBM Cloud Private e configure os parâmetros `tiller_http_proxy` e `tiller_https_proxy`. Isso definirá as configurações de proxy do daemon tiller do Helm para preencher o Catálogo de Apps do IBM Cloud Private.

     ```
     sudo vi /<installation_directory>/cluster/config.yaml

     # Licensed Materials - Property of IBM
     # @ Copyright IBM Corp. 2017 All Rights Reserved
     # US Government Users Restricted Rights - Use, duplication or disclosure restricted by
     GSA ADP Schedule Contract with IBM Corp.
     ---
     ## Network Settings network_type: calico ## Network in IPv4 CIDR format network_cidr:
     10.1.0.0/16 ## Kubernetes Settings service_cluster_ip_range: 10.0.0.1/24 ...
     tiller_http_proxy: http://1.2.3.4:3128 tiller_https_proxy: http://1.2.3.4:3128 ...
     ```

     Agora, continue o processo de instalação do IBM Cloud Private normalmente. No IBM Cloud Private console de gerenciamento, verifique **Catálogo** .

- Instalação Automática do Docker Usando IBM Cloud Private

  Se você instalou o Docker em seu nó de inicialização manualmente e não o instalou em seus outros nós do cluster, o Docker está sendo implementado automaticamente utilizando o instalador do IBM Cloud Private. Siga as etapas para instalar o IBM Cloud Private por trás de um proxy HTTP.

  1. Remova o comentário das variáveis de ambiente do Docker a seguir em seu arquivo `config.yaml`:

```
## Docker environment setup
docker_env:
- HTTP_PROXY=http://1.2.3.4:3128
- HTTPS_PROXY=http://1.2.3.4:3128
- NO_PROXY=localhost,127.0.0.1,{{ cluster_CA_domain }}
## Install/upgrade docker   version
```

2. Customize o arquivo `config.yaml` do IBM Cloud Private e configure os parâmetros `tiller_http_proxy` e `tiller_https_proxy`, conforme mostrado no comando a seguir:

```
sudo vi /<installation_directory>/cluster/config.yaml

# Licensed Materials - Property of IBM
# IBM Cloud private
# @ Copyright IBM Corp. 2017 All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.
---
## Network Settings network_type: calico ## Network in IPv4 CIDR format network_cidr:
10.1.0.0/16 ## Kubernetes Settings service_cluster_ip_range: 10.0.0.1/24 ...
tiller_http_proxy: http://1.2.3.4:3128 tiller_https_proxy: http://1.2.3.4:3128 ...
```

Agora, continue o processo de instalação do IBM Cloud Private normalmente. No IBM Cloud Private console de gerenciamento, verifique **Catálogo** .

Configuração do Proxy de Pós-instalação

Pós-instalação, é possível editar as configurações de proxy do IBM Cloud Private com as etapas a seguir:

1. No console de gerenciamento do IBM Cloud Private, acesse **Cargas de trabalho** > **Implementações**.
2. Na procura de **Implementações** para o helm-api.
3. Clique em **Editar**.
4. Procure as seguintes linhas:

```
{
"nome": "HTTP_PROXY"
},
{
"nome": "HTTPS_PROXY"
},
{
"name": "NO_PROXY",
"value": "<ICP cluster IP>,mycluster.icp,mongodb,platform-identity-
provider,localhost,127.0.0.1"
},
```

5. Edite HTTP_PROXY e HTTPS_PROXY conforme apropriado.

```
{
"nome": "HTTP_PROXY", "valor": "http://1.2.3.4:3128 "
},
{
"nome": "HTTPS_PROXY", "valor": "http://1.2.3.4:3128 "
},
{
"name": "NO_PROXY",
"value": "<ICP cluster IP>,mycluster.icp,mongodb,platform-identity-provider,icp-management-
ingress,iam-pap,localhost,127.0.0.1"
},
```

Nota: dependendo de seu ambiente, os valores NO_PROXY podem variar. Por exemplo, eles podem incluir os recursos Ingressos e Serviços do Kubernetes. É importante que o NO_PROXY seja configurado completamente para evitar comunicação do IBM Cloud Private pelo proxy.

6. Clique em **Enviar**.
7. Acesse **Catálogo** para verificar que os gráficos Helm são mostrados.

Isolando Ambientes de Rede e de Cálculo

Implemente nós do trabalhador dedicados para isolar os ambientes de rede e de cálculo em seu cluster do IBM® Cloud Private.

Seu cluster do IBM Cloud Private pode ser configurado com múltiplos namespaces. Se você deseja restringir a comunicação entre namespaces ou restringir a implementação do pod a um conjunto específico de nós, é possível atribuir os nós do trabalhador dedicados a cada namespace. Cada namespace é configurado em uma sub-rede diferente.

Também é possível atribuir nós de proxy separados a esses namespaces. Os nós do proxy especificam os controladores de ingresso que atendem um namespace específico.

Pré-requisitos

Planeje os requisitos de rede e de nó antes de implementar seu cluster do IBM Cloud Private.

- Planeje o número de nós do cluster de que você precisa com base no número de namespaces que deseja isolar.
- Certifique-se de que o parâmetro `network_type` que está no arquivo `<installation_folder>/cluster/config.yaml` esteja configurado como `calico`.
- Quando um grupo de hosts de proxy possui vários nós do proxy, configure um balanceador de carga na frente do grupo de hosts do proxy.

Isolando namespaces e proxies durante a instalação do IBM Cloud Private

Prepare seu cluster concluindo as etapas a seguir:

1. Para cada namespace que você deseja isolar, inclua um grupo de hosts customizados. Se necessário, inclua os grupos de hosts do nó do proxy. Para obter mais informações sobre como incluir grupos de hosts customizados, consulte Definindo grupos de hosts customizados.

2. Inclua a seguinte parte de código no arquivo `config.yaml` que está na pasta `<installation_directory>/cluster`.

```
isolated_namespaces: [{namespace: <namespace_name>, hostgroup: <hostgroup_name>}, {namespace:
<namespace_name>, hostgroup: <hostgroup_name>}]

isolated_proxies: [{namespace: <namespace_name>, hostgroup: <hostgroup_name>, lb_address:
<load_balancer_IP_address>}, {namespace: <namespace_name>, hostgroup: <hostgroup_name>, lb_address:
<load_balancer_IP_address>}]
```

Nota:

- Os namespaces que você especifica no parâmetro `isolated_namespaces` são criados durante a instalação do IBM Cloud Private.
- No parâmetro `isolated_proxies`, se você estiver especificando um namespace que não está definido no parâmetro `isolated_namespaces`, deve-se criar manualmente esse namespace depois de instalar o IBM Cloud Private.
- O parâmetro `lb_address` é opcional. Se você não especificar o parâmetro `lb_address`, o primeiro nó do grupo de hosts a partir do arquivo hosts será usado como o balanceador de carga padrão.

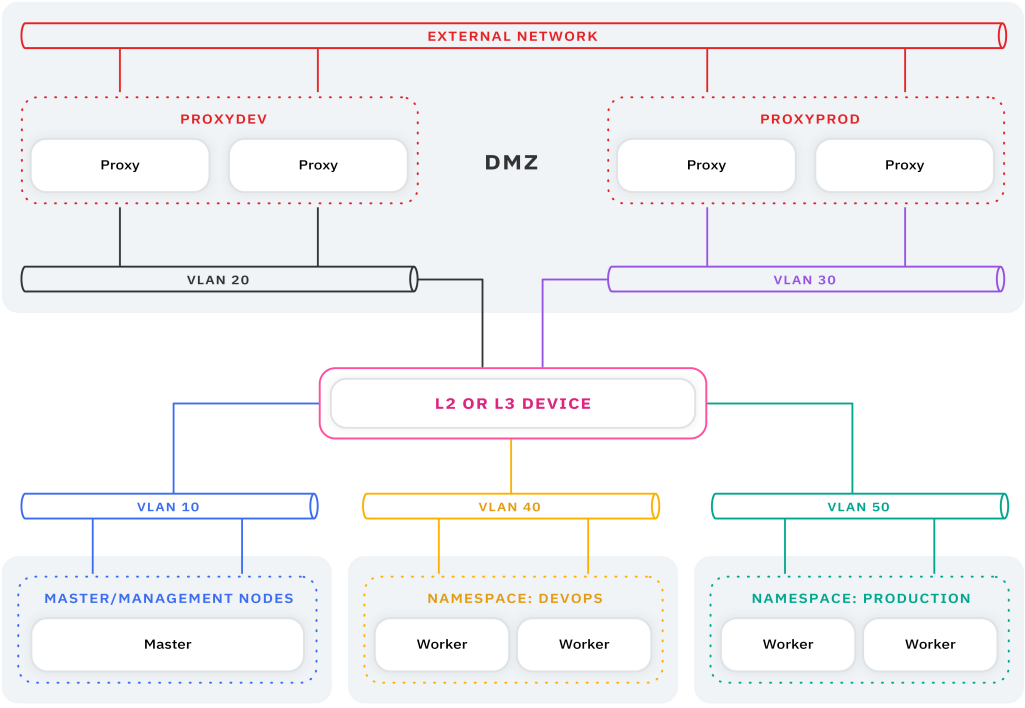
Em seguida, continue com a instalação do IBM Cloud Private.

Configuração de exemplo

Considere um cluster do IBM Cloud Private com a configuração a seguir:

- Os nós principais ou de gerenciamento estão na rede local virtual (VLAN) 10.
- Dois nós do trabalhador, dedicados ao namespace `devops`, estão na VLAN 40.
- Dois nós do trabalhador, dedicados ao namespace `production`, estão na VLAN 50.
- Dois nós de proxy, dedicados ao namespace `devops`, estão na VLAN 20.
- Dois nós de proxy, dedicados ao namespace `production`, estão na VLAN 30.

A configuração de rede do cluster é semelhante ao diagrama a seguir:



A seguir estão as etapas para preparar o cluster:

1. Inclua grupos de hosts customizados para cada namespace. Além disso, inclua os grupos de hosts do nó do proxy. O `<installation_directory>/cluster/hosts` é atualizado da seguinte forma:

Nota: cada grupo de hosts está em uma VLAN separada.

```
[ master ] 172.68.10.10

[ trabalhador ] 172.68.10.20

[proxy]
172.68.10.10

[ hostgroup-dev ] 172.68.40.10 172.68.40.11

[ hostgroup-prod ] 172.68.50.10 172.68.50.11

[ hostgroup-proxydev ] 172.68.20.10 172.68.20.11

[ hostgroup-proxyprod ] 172.68.30.10 172.68.30.11
```

1. Inclua a parte de código a seguir no arquivo `config.yaml`:

```
isolated_namespaces: [{namespace: devops, hostgroup: dev}, {namespace: production, hostgroup: prod}]

isolated_proxies: [{namespace: devops, hostgroup: proxydev, lb_address: 172.68.20.11}, {namespace:
production, hostgroup: proxyprod}]
```

Os namespaces `devops` e `production` são criados durante a instalação de seu cluster.

Para ativar o isolamento do ambiente após a instalação do IBM Cloud Private, consulte Ativando o namespace e o isolamento de proxy.

Configurando para um ambiente IBM Power

As definições de configuração a seguir são recomendadas ao instalar o IBM® Cloud Private em um ambiente do IBM Power.

Configurações de partição de troca do sistema operacional

Inclua uma pequena partição de troca e desative a contabilidade de troca para controlar pequenos picos no uso de memória do pod que excede o limite configurado e para evitar erros de falta de memória.

Nota: o espaço de troca não deve ser maior que 4 GB. O espaço de troca não deve ser compartilhado com unidades que gerenciam grandes quantias de atividade de E/S, como `/var/lib/docker` e `/var/log`.

- Ative uma partição de troca pequena (de 2 a 4 GB) em cada nó no cluster.
- Para Red Hat Enterprise Linux, inclua `swapaccount=0` na linha de comandos do kernel concluindo as seguintes etapas:

  1. Abra o arquivo `/etc/default/grub` .
  2. Inclua `swapaccount=0` nas opções existentes editando a opção *GRUB_CMDLINE_LINUX*.
  3. Execute o comando a seguir:

     ```
     grub2-mkconfig -o /boot/grub2/grub.cfg
     ```

  4. Reinicialize os nós antes de instalar o IBM Cloud Private.

  Nota: A contabilidade de troca é desativada por padrão no Ubuntu e no SUSE Linux Enterprise Server, portanto, não são necessárias mudanças na linha de comandos do kernel para esses sistemas operacionais.

IBM Cloud Private configurações

Algumas de suas configurações de ambiente do IBM Cloud Private são determinadas pelo modo com que você configura seu ambiente do IBM Power. As seções a seguir descrevem um ambiente de amostra e suas definições de configuração recomendadas.

Importante: as recomendações nesta seção são para clusters que possuam tanto nós principais quanto de gerenciamento do Power. Para um cluster combinado no qual os nós Power são apenas trabalhadores, apenas aplique as mudanças do sistema operacional na seção *Sistema operacional*.

Características da configuração:

- O número de CPUs é 32, ou maior (conforme identificado pelo sistema operacional).

  As CPUs que são identificadas pelo sistema operacional podem ser diferentes da CPU que está configurada no perfil de LPAR para hypervisors do PowerVM. Por exemplo, o sistema operacional de uma Partição Lógica (LPAR) do PowerVM que é configurada com 4 vCPUs relata 32 CPUs. Isso ocorre devido à configuração padrão de SMT=8. O método que o sistema operacional usa para determinar o número de CPUs que são identificadas é determinado pela fórmula a seguir:

  ```
  (vCPUs * SMT value)
  ```

  Este exemplo é resolvido como a fórmula a seguir:

  ```
  (4 * 8 = 32)
  ```

  Para determinar esse valor, é possível executar um dos comandos a seguir:

  ```
  /proc/cpuinfo
  ```

  ou

  ```
  lscpu
  ```

- O RAM é de 64 GB ou superior.
- Para o hypervisor do PowerVM, a autorização do processador deve ter um valor maior ou igual a 2 para o nó principal e os de gerenciamento.

Configurações de pré-instalação

Se você tiver 32 ou mais CPUs, deverá atualizar as configurações em seu `config.yaml` para o ambiente do Power. Há um arquivo chamado `power_config.yaml` no diretório xxx que contém as configurações atualizadas.

Substitua o arquivo `config.yaml` pelo arquivo `power.config.yaml`. Se estiver implementando seu cluster em um ambiente IBM Power, você deverá usar as configurações no arquivo `power.config.yaml`. Conclua as etapas a seguir para substituir o arquivo:

1. Insira o comando a seguir para renomear o arquivo `config.yaml` existente para `config.yaml.orig`:

   ```
   sudo mv /<installation_directory>/cluster/config.yaml
   /<installation_directory>/cluster/config.yaml.orig
   ```

   Substitua *installation_directory* pelo caminho para o diretório de instalação.

2. Insira o comando a seguir para renomear o arquivo `power.config.yaml` para `config.yaml`:

   ```
   sudo cp /<installation_directory>/cluster/power.config.yaml
   /<installation_directory>/cluster/config.yaml
   ```

   Substitua *installation_directory* pelo caminho para o diretório de instalação.

Configurações de pós-instalação

É possível, opcionalmente, incluir um alerta que o notifica se o uso de memória de um contêiner atinge 90% de sua memória disponível. Esse alerta indica a necessidade de revisitar o tamanho do pod. Conclua as etapas a seguir para criar o alerta:

1. Crie um arquivo denominado `pod-mem-usage-alert.yaml` com o conteúdo a seguir:

   ```
   apiVersion: monitoringcontroller.cloud.ibm.com/v1
   kind: AlertRule
   metadata:
    name: pod-mem-usage
   spec:
    enabled: true
    data: |-
      groups:
        - name: podMemUsage
          rules:
            - alert: podMemUsage
              expr:  (sum(container_memory_working_set_bytes) by (name, pod_name,
   namespace)/sum(container_spec_memory_limit_bytes) by (name, pod_name, namespace)) > 0.90 and
   (sum(container_memory_working_set_bytes) by (name, pod_name,
   namespace)/sum(container_spec_memory_limit_bytes) by (name, pod_name, namespace)) != Inf
              for:  30m
              annotations:
                description: 'Pod {{  $labels.pod_name  }} in namespace {{  $labels.namespace  }}
   is reaching memory limit threshold'
                summary: Memory Utilization of Pod is reaching limit
   ```

2. Implemente o novo alerta inserindo o comando a seguir:

   ```
   kubectl aplicar -f pod-mem-usage-alert.yaml
   ```

   Consulte a seção *Alertas* do Monitoramento de cluster do IBM Cloud Private para obter mais informações sobre alertas.
   Dica: consulte `https://icp-master-ip:8443/alertmanager` para visualizar seus alertas ativos.

Instalando as edições Cloud Native, Enterprise e Community do IBM Cloud Private

Siga as etapas nesse tópico para instalar o IBM Cloud Private Native ou Enterprise editions ou o IBM® Cloud Private-CE (Community Edition).

Para o IBM Cloud Private Cloud Native ou Enterprise editions, é possível instalar um cluster padrão ou de alta disponibilidade (HA). Para o IBM Cloud Private-CE, é possível configurar um nó de gerenciamento principal, do trabalhador, de proxy e nós opcionais em seu cluster.

É possível ter um cluster do IBM Cloud Private que suporte os sistemas Linux® x86_64, Linux® on Power® (ppc64le) e Linux on IBM Z and LinuxONE.

Antes de instalar:

- Você deve preparar seu cluster. Consulte Configurando o seu cluster.
- Se você deseja ativar o IBM Multicloud Manager, consulte Preparando-se para a instalação do IBM Multicloud Manager.
- Se seu nó principal ou do proxy usar um sistema operacional SUSE Linux Enterprise Server (SLES), durante a instalação, você deve desativar todos os firewalls em seu cluster.

A instalação pode ser concluída em seis etapas principais:

1. Instale o Docker para o seu nó de inicialização apenas
2. Configure o ambiente de instalação
3. Customizar seu cluster
4. Configure o Docker para os nós do cluster
5. Implementar o Ambiente
6. Verifique a instalação

Quando a instalação estiver concluída, será possível acessar seu cluster e concluir tarefas de pós-instalação.

Se encontrar erros durante a instalação, consulte Resolução de problemas de instalação.

Etapa 1: instalar o Docker somente para o nó de inicialização

O nó de inicialização é o nó que é usado para a instalação de seu cluster. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre o nó de inicialização, consulte Nó de inicialização.

É necessária uma versão do Docker que seja suportada por IBM Cloud Private instalada em seu nó de inicialização. Consulte Versões do Docker suportadas. Para instalar o Docker, consulte Instalando manualmente o Docker.

Etapa 2: configurar o ambiente de instalação

1. Efetue login no nó de inicialização como um usuário com permissões raiz.
2. Faça download do arquivo ou da imagem de instalação.
   - **Para a instalação do IBM Cloud Private Native ou Enterprise**: Faça download do arquivo ou arquivos corretos para o tipo de nós em seu cluster a partir do website do IBM Passport Advantage®.
     - Para um cluster do Linux x86_64, faça download do arquivo `ibm-cloud-private-x86_64-3.2.0.tar.gz`.
     - Para um cluster do Linux on Power (ppc64le), faça download do arquivo `ibm-cloud-private-ppc64le-3.2.0.tar.gz`.
     - Para um cluster do IBM® Z, faça download do arquivo `ibm-cloud-private-s390x-3.2.0.tar.gz`.
   - **Para a instalação do IBM Cloud Private-CE**: Faça download da imagem do CE a partir do Docker Hub executando o seguinte comando:

     ```
     docker pull ibmcom/icp-inception:3.2.0
     ```

     Nota: Esta imagem do instalador suporta o Linux® em sistemas x86_64, sistemas Linux on Power LE de 64 bits e sistemas Linux on IBM Z and LinuxONE.
3. **Somente para a instalação do IBM Cloud Private Native ou Enterprise**: Extraia as imagens e carregue-as no Docker. Extrair as imagens pode levar alguns minutos.
   - Para o Linux x86_64, execute este comando:

     ```
     tar xf ibm-cloud-private-x86_64-3.2.0.tar.gz -O | sudo docker load
     ```
   - Para o Linux on Power (ppc64le), execute este comando:

     ```
     tar xf ibm-cloud-private-ppc64le-3.2.0.tar.gz -O | sudo docker load
     ```
   - Para o Linux on IBM Z and LinuxONE, execute este comando:

     ```
     tar xf ibm-cloud-private-s390x-3.2.0.tar.gz -O | sudo docker load
     ```
4. Crie um diretório de instalação (`/<installation_directory>/`) para armazenar os arquivos de configuração do IBM Cloud Private ou do IBM Cloud Private-CE e mude para esse diretório. **Nota**: O diretório de instalação deve ter pelo menos

50 GB de espaço em disco disponível para a instalação e os arquivos de instalação. Por exemplo, para armazenar os arquivos de configuração do IBM Cloud Private em `/opt/ibm-cloud-private-3.2.0`, execute os seguintes comandos:

```
sudo mkdir /opt/ibm-cloud-private-3.2.0;
cd /opt/ibm-cloud-private-3.2.0
```

5. Extraia os arquivos de configuração da imagem do instalador.

   - **Para instalação do IBM Cloud Private Native ou Enterprise**:

     - Para o Linux x86_64, execute este comando:

       ```
       sudo docker run -v $(pwd):/data -e LICENSE=accept \
       ibmcom/icp-inception-amd64:3.2.0-ee \
       cp -r cluster /data
       ```

     - Para o Linux on Power (ppc64le), execute este comando:

       ```
       sudo docker run -v $(pwd):/data -e LICENSE=accept \ ibmcom/icp-inception-
       ppc64le:3.2.0-ee \ cp -r cluster /data
       ```

     - Para o Linux on IBM Z and LinuxONE, execute este comando:

       ```
       sudo docker run -v $(pwd):/data -e LICENSE=accept \
       ibmcom/icp-inception-s390x:3.2.0-ee \
       cp -r cluster /data
       ```

   - **Para instalação do IBM Cloud Private-CE**:

     ```
       sudo docker run -e LICENSE=accept \
     -v "$(pwd)":/data ibmcom/icp-inception:3.2.0 cp -r cluster /data
     ```

     Um diretório de `cluster` é criado dentro do seu diretório de instalação. Por exemplo, se o seu diretório de instalação for `/opt/ibm-cloud-private-3.2.0`, a pasta `/opt/ibm-cloud-private-3.2.0/cluster` será criada. Para obter uma visão geral da estrutura de diretório do cluster, consulte Estrutura de diretório do cluster.

     Nota: Por padrão, o diretório do cluster é de propriedade de `root`. Se for necessário que o diretório pertença a um usuário diferente, execute `chmod -R` no diretório.

6. Opcional: É possível visualizar o arquivo de licença para o IBM Cloud Private. Para uma lista de formatos de linguagem suportadas, consulte Linguagens suportadas.

   - **Para instalação do IBM Cloud Private Native ou Enterprise**:

     - Para o Linux x86_64, execute este comando:

       ```
       sudo docker run -e LICENSE=view -e LANG=$LANG ibmcom/icp-inception-amd64:3.2.0-ee
       ```

     - Para o Linux on Power (ppc64le), execute este comando:

       ```
       sudo docker run -e LICENSE=view -e LANG=$LANG ibmcom/icp-inception-ppc64le:3.2.0-ee
       ```

     - Para o Linux on IBM Z and LinuxONE, execute este comando:

       ```
        sudo docker run -e LICENSE=view -e LANG=$LANG ibmcom/icp-inception-s390x:3.2.0-ee
        cp -r cluster /data
       ```

   - **Para a instalação do IBM Cloud Private-CE**:

     ```
     sudo docker run -e LICENSE=view -e LANG=$LANG ibmcom/icp-inception:3.2.0
     ```

   Nota: O valor `$LANG` deve estar no idioma suportado. Por exemplo, para visualizar a licença em chinês simplificado usando o Linux x86_64 para uma instalação do IBM Cloud Private Cloud Native ou Enterprise, execute o seguinte comando:

   ```
   sudo docker run -e LICENSE=view -e LANG=zh_CN ibmcom/icp-inception-amd64:3.2.0-ee
   ```

7. Crie uma conexão segura entre o nó de inicialização e todos os outros nós em seu cluster. Conclua uma das seguintes configurações:

   - Configure o SSH em seu cluster. Consulte Compartilhando chaves SSH entre os nós do cluster.
   - Configure a autenticação de senha em seu cluster. Consulte Configurando a senha de autenticação para nós do cluster.

8. Inclua o endereço IP de cada nó no cluster no arquivo `/<installation_directory>/cluster/hosts`. Consulte Configurando as funções do nó nos arquivos host. Também é possível definir grupos de hosts customizados, consulte Definindo grupos de hosts customizados.

   Nota: Para a instalação do IBM Cloud Private Native ou Enterprise: Os nós do trabalhador podem suportar arquiteturas mistas. É possível incluir nós do trabalhador em um único cluster que são executados em plataformas Linux x86_64, Linux on Power (ppc64le) e IBM Z. Os nós que não são do trabalhador suportam somente um tipo de arquitetura.

   Nota: Para a instalação do IBM Cloud Private-CE: Os nós do trabalhador e do proxy podem suportar arquiteturas mistas. Não é necessário fazer download ou extrair pacotes específicos de plataforma para configurar um ambiente do trabalhador ou do proxy de arquitetura mista para o IBM Cloud Private-CE. Para incluir nós do trabalhador ou do proxy em um cluster que contém plataformas Linux x86_64, Linux on Power (ppc64le) e Linux on IBM Z and LinuxONE, é necessário incluir o endereço IP desses nós somente no arquivo `/<installation_directory>/cluster/hosts`.

9. **Somente para a instalação do IBM Cloud Private Native ou Enterprise**: Mova os arquivos de imagem de seu cluster para a pasta `/<installation_directory>/cluster/images`.

   1. Crie um diretório de imagens executando o seguinte comando:

      ```
      sudo mkdir -p /<installation_directory>/cluster/images
      ```

   2. Se seu cluster contiver quaisquer nós `x86_64`, coloque o pacote x86 no diretório de imagens. Inclua o caminho para seu arquivo de imagem de instalação no seguinte comando:

      ```
      sudo mv ibm-cloud-private-x86_64-3.2.0.tar.gz cluster/images/
      ```

   3. Se seu cluster contiver quaisquer nós `ppc64le`, coloque o pacote ppc64le no diretório de imagens. Inclua o caminho para seu arquivo de imagem de instalação no seguinte comando:

      ```
      sudo mv ibm-cloud-private-ppc64le-3.2.0.tar.gz cluster/images/
      ```

   4. Se seu cluster contiver quaisquer nós `s390x`, coloque o pacote s390x no diretório de imagens. Inclua o caminho para seu arquivo de imagem de instalação no seguinte comando:

      ```
      sudo mv ibm-cloud-private-s390x-3.2.0.tar.gz cluster/images/
      ```

Etapa 3: Customizar seu cluster

As configurações no arquivo `config.yaml`, localizado no diretório `/<installation_directory>/cluster/`, contêm todas as definições de configuração necessárias para implementar seu cluster.

1. **Apenas para o ambiente IBM Power** Substitua o arquivo `config.yaml` pelo arquivo `power.config.yaml`. Se estiver implementando seu cluster em um ambiente IBM Power, você deverá usar as configurações no arquivo `power.config.yaml`. Conclua as etapas a seguir para substituir o arquivo:

   1. Insira o comando a seguir para renomear o arquivo `config.yaml` existente para `config.yaml.orig`:

      ```
      sudo mv /<installation_directory>/cluster/config.yaml
      /<installation_directory>/cluster/config.yaml.orig
      ```

      Substitua *installation_directory* pelo caminho para o diretório de instalação.
   2. Insira o comando a seguir para renomear o arquivo `power.config.yaml` para `config.yaml`:

      ```
      sudo cp /<installation_directory>/cluster/power.config.yaml
      /<installation_directory>/cluster/config.yaml
      ```

      Substitua *installation_directory* pelo caminho para o diretório de instalação.

2. Configure uma senha padrão no arquivo `config.yaml` que atenda à regra de regra cumprimento de senha padrão `'^([a-zA-Z0-9\-]{32,})$'`. Também é possível definir um conjunto customizado de regras de senha.

   1. Abra o arquivo `/<installation_directory>/cluster/config.yaml` e configure o `default_admin_password`. A senha deve satisfazer todas as expressões regulares que são especificadas em `password_rules`.
   2. Opcional: É possível definir uma ou mais regras como expressões regulares em uma lista de matrizes que a senha deve transmitir. Por exemplo, uma regra pode declarar que a senha deve ser maior que um número especificado de caracteres e/ou que deve conter pelo menos um caractere especial. As regras são gravadas como expressões regulares que são suportadas pela linguagem de programação Go. Para definir um conjunto de regras de senha, inclua o seguinte parâmetro e valores no arquivo `config.yaml`:

```
password_rules:
- ' ^. {10, } '
- '.*[!@#\$%\^&\*].*'
```

Para desativar o `password_rule`, inclua `(. *)`

```
password_rules:
- '(.*)'
```

Nota: O `default_admin_password` deve corresponder a todas as regras definidas. Se `password_rules` não estiver definido, o `default_admin_password` deverá atender à regra de cumprimento de passaporte padrão `'^([a-zA-Z0-9\-]{32,})$'`.

3. Opcional: ative o IBM Multicloud Manager. No arquivo `config.yaml`, `multicluster-hub` fica ativado por padrão. Para ativar a funcionalidade completa do IBM Multicloud Manager, configure `single_cluster_mode` para `false`. Para obter mais informações, consulte Configurando o IBM Multicloud Manager durante a instalação do IBM Cloud Private.

4. Opcional: Customize seu cluster. Para revisar a lista completa de parâmetros que estão disponíveis para customização, consulte Customizando o cluster com o arquivo config.yaml. Para outros tipos de customizações que devem ser configuradas durante a instalação, como a configuração do serviço de monitoramento ou GlusterFS, revise Customizando sua instalação.

5. Em um ambiente que tem múltiplas interfaces de rede (NICs), como OpenStack e AWS, deve-se incluir o código a seguir no arquivo `config.yaml`:

   - **Para instalação do IBM Cloud Private Native ou Enterprise**:

     ```
     cluster_lb_address: <external address>
     proxy_lb_address: <external address>
     ```

   - **Para instalação do IBM Cloud Private-CE**:

     ```
     cluster_lb_address: <external IP address>
     ```

   O valor `<external address>` é o endereço IP, o nome completo do domínio ou o endereço IP flutuante do OpenStack que gerencia a comunicação com serviços externos. Configurar o parâmetro `proxy_lb_address` é necessário apenas para ambientes de HA do proxy.

6. Para ambientes de HA, há várias opções de instalação de HA. Consulte Configurações de HA.

Etapa 4: configurar o Docker para os nós do cluster

Os nós do cluster são os nós principais, do trabalhador, do proxy e de gerenciamento. Para saber mais, consulte Arquitetura.

Você precisa de uma versão do Docker que seja suportada pelo IBM Cloud Private instalado em seu nó do cluster. Consulte Versões do Docker suportadas. Se você não tiver uma versão suportada do Docker que está instalada em seus nós do cluster, o IBM Cloud Private poderá instalar automaticamente o Docker em seus nós do cluster durante a instalação.

Para preparar seus nós do cluster para instalação automática do Docker, consulte Configurando nós do cluster para instalação automática do Docker.

Etapa 5: Implementar o ambiente

1. Mude para a pasta `cluster` em seu diretório de instalação executando o seguinte comando:

   ```
   cd ./cluster
   ```

2. **Opcional somente para a instalação do IBM Cloud Private Native ou Enterprise**: Dependendo de suas opções, pode ser necessário incluir mais parâmetros no comando de implementação. Se você especificou o parâmetro `offline_pkg_copy_path` no arquivo `config.yaml`; no comando de implementação, inclua a opção `-e ANSIBLE_REMOTE_TEMP=<offline_pkg_copy_path>`, em que `<offline_pkg_copy_path>` é o valor do parâmetro `offline_pkg_copy_path` configurado no arquivo `config.yaml`.

   Nota: Por padrão, o comando para implementar seu ambiente está configurado para implementar 15 nós de cada vez. Se o seu cluster tiver mais de 15 nós, a implementação poderá levar um tempo mais longo para ser concluída. Se você desejar acelerar a implementação, será possível especificar um número mais alto de nós a ser implementado por vez. Use o argumento `-f <number of nodes to deploy>` com o comando.

3. **Opcional**: Para verificar e validar se seu ambiente está disponível para instalar o IBM Cloud Private, execute o comando apropriado:

   - **Para instalação do IBM Cloud Private Native ou Enterprise**:

     - Para o Linux x86_64, execute este comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee check
       ```

     - Para o Linux on Power (ppc64le), execute este comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee check
       ```

     - Para o Linux on IBM Z and LinuxONE, execute este comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
         -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee check
       ```

   - **Para instalação do IBM Cloud Private-CE**:

     ```
     sudo docker run --net=host -t -e LICENSE=accept \
     -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 check
     ```

4. Implemente seu ambiente:

   - **Para instalação do IBM Cloud Private Native ou Enterprise**:

     - Para o Linux x86_64, execute este comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
        -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee install
       ```

     - Para o Linux on Power (ppc64le), execute este comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \ -v "$(pwd)":/installer/cluster
       ibmcom/icp-inception-ppc64le:3.2.0-ee install
       ```

     - Para o Linux on IBM Z and LinuxONE, execute este comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
         -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee install
       ```

   - **Para instalação do IBM Cloud Private-CE**:

     ```
     sudo docker run --net=host -t -e LICENSE=accept \
     -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 install
     ```

5. **Opcional somente para instalação do IBM Cloud Private Native ou Enterprise**: Se encontrar erros durante a implementação, execute novamente o comando de implementação com `-v` para coletar outras mensagens de erro. Se você continuar a receber erros durante a nova execução, execute o comando a seguir para coletar os arquivos de log:

   - Para o Linux x86_64, execute este comando:

     ```
      sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster
     ibmcom/icp-inception-amd64:3.2.0-ee healthcheck
     ```

   - Para o Linux on Power (ppc64le), execute este comando:

     ```
      sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster
     ibmcom/icp-inception-ppc64le:3.2.0-ee healthcheck
     ```

   - Para o Linux on IBM Z and LinuxONE, execute este comando:

     ```
      sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster
     ibmcom/icp-inception-s390x:3.2.0-ee healthcheck
     ```

     Os arquivos de log estão localizados no diretório `cluster/logs`.

Etapa 6: Verificar o status de sua instalação

Se a instalação for bem-sucedida, as informações de acesso para seu cluster serão exibidas. A URL é `https://<Cluster Master Host>:<Cluster Master API Port>`, em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal.

Acesse o seu cluster

Agora é possível acessar seu cluster. Em um navegador da web, navegue para a URL de seu cluster. Para obter uma lista de navegadores suportados, consulte Navegadores suportados.

- Para saber como acessar seu cluster usando o IBM Cloud Privateconsole de gerenciamento a partir de um navegador da web, consulte Acessando o cluster do IBM Cloud Private usando o console de gerenciamento.
- Para saber como acessar seu cluster usando a linha de comandos do Kubernetes (kubectl), consulte Acessando seu cluster do IBM Cloud Private usando a CLI kubectl.

Notas:

- Se não for possível efetuar login imediatamente após a conclusão da instalação, pode ser que os serviços de gerenciamento não estejam prontos. Aguarde alguns minutos e tente novamente.
- Você pode ver uma mensagem `502 Bad Gateway` quando abrir uma página no console de gerenciamento logo após a instalação. Se você vir, o serviço do NGINX não iniciou todos os componentes. As páginas são carregadas após todos os componentes serem iniciados.

Tarefas pós-instalação

1. Reinicie seu firewall.
2. Assegure-se de que todas as portas padrão do IBM Cloud Private estejam abertas. Para obter mais informações sobre as portas padrão do IBM Cloud Private, consulte Portas padrão.
3. Faça backup do nó de inicialização. Copie o seu diretório `/<installation_directory>/cluster` para um local seguro. Se você usar chaves SSH para proteger o seu cluster, assegure-se de que as chaves SSH no diretório de backup permaneçam em sincronia.
4. Mantenha o acesso ao nó de inicialização adequado. O nó de inicialização contém informações sobre autenticação que são usadas para a implementação inicial de 1 dia do IBM Cloud Private e atualizações de 2 dias para o IBM Cloud Private. O acesso ao nó de inicialização deve ser limitado somente os usuários com uma necessidade real de negócios e esse acesso deve ser controlado usando as ferramentas de gerenciamento de identidade corporativa para aprovação, recertificação periódica, revogação de acesso na demissão do funcionário ou mudanças de cargo. Somente usuários que têm acesso ao nó de inicialização devem ser aqueles que têm a função `clusteradmin` para o IBM Cloud Private.
5. Instale outro software do seu pacote configurável. Consulte Instalando o software IBM no IBM Cloud Private.
6. **Opcional**: revise o Contrato de Licença do Programa Internacional (IPLA) para o IBM Cloud Private e o IBM Multicloud Manager:
    1. Abra o seguinte link: https://www-03.ibm.com/software/sla/sladb.nsf/search?OpenForm
    2. Procure um dos seguintes números de Informações sobre Licença:
        - L-TKAO-BA3Q8F - IBM Cloud Private 3.2.0
        - L-TKAO-BA3Q3J - IBM Cloud Private Foundation 3.2.0
        - L-ECUN-BALP9Z - IBM Multicloud Manager Enterprise Edition 3.2.0
7. **Opcional**: revise o arquivo de Aviso e o arquivo de licença não IBM para o IBM Cloud Private e o IBM Multicloud Manager:
    1. Acesse o diretório `<installation_directory>/cfc/license`.
    2. Abra o arquivo `Stacked_License_for_ICP_ICP_Foundation_MCMEE_3.2.0.zip`.
    3. Acesse o diretório `RTF` e abra os arquivos `notices.rtf` e `non_ibm_license.rtf` para revisar os avisos e as informações sobre licença não IBM.

Instalando o software IBM no IBM Cloud Private

É possível instalar outro software IBM no IBM® Cloud Private para estender a funcionalidade da plataforma IBM Cloud Private ou incluir no IBM Cloud Private Catalog.

Existem duas opções para instalar o software IBM no IBM Cloud Private:

1. Incluindo o software IBM no IBM Cloud Private Catalog: É possível usar a CLI para instalar manualmente o software IBM e, em seguida, carregar o gráfico do Helm no IBM Cloud Private Catalog.

2. Instalando o software IBM na plataforma IBM Cloud Private: É possível usar o instalador do IBM Cloud Private para instalar e ativar o software IBM imediatamente no cluster.

É possível obter o software IBM que pode se integrar com o IBM Cloud Private por meio de pacotes configuráveis do IBM Cloud Private, dos IBM Cloud Paks e dos IBM Certified Containers. Cada pacote configurável do IBM Cloud Private e o IBM Cloud Pak contêm um software autorizado diferente do qual você faz download a partir do IBM Passport Advantage. A lista de complementos que estão disponíveis no Passport Advantage muda frequentemente, à medida que novos são incluídos. Para obter mais informações, consulte Pacotes configuráveis do IBM Cloud Private, Identificando o IBM Cloud Paks e Identificando os IBM Certified Containers.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Incluindo o software IBM no IBM Cloud PrivateCatalog

Você deve instalar manualmente os gráficos do Helm para outros produtos que podem se integrar com o IBM® Cloud Private.

Antes de carregar um gráfico, conclua os pré-requisitos a seguir:

- Instale o IBM Cloud Private. Consulte Instalando as edições do IBM Cloud Private Cloud Native, Enterprise e Community.
- Instale a CLI do IBM Cloud Private e efetue login em seu cluster. Consulte Instalando a CLI do IBM Cloud Private.
- Configure a autenticação de seu computador para o host do registro de imagem privado do Docker e efetue login no registro privado. Consulte Configurando a autenticação para a CLI do Docker.
- Se você não for um usuário raiz, assegure-se de que sua conta seja parte do grupo `docker`. Consulte Etapas pós-instalação para o Linux® na documentação do Docker.
- Assegure-se de que você tenha uma conexão de rede estável entre seu computador e o cluster.

Para instalar o software IBM e carregar um gráfico do Helm no Catalog:

1. Faça download do arquivo compactado a partir do IBM Passport Advantage®.

2. Assegure-se de que você tenha espaço em disco suficiente para carregar as imagens nos arquivos compactados em seu computador.

    1. Verifique o uso do disco do Docker executando o seguinte comando:

        ```
        docker system df
        ```

        Para mais opções de comando, consulte docker system df na documentação do Docker.

    2. Se precisar de mais espaço em disco, execute uma das seguintes ações:
        - Remova as imagens antigas do Docker.
        - Aumente a quantia de armazenamento que o daemon do Docker usa. Para aumentar a quantia de armazenamento que o daemon do Docker usa, consulte a entrada para `dm.basesize` na documentação do Docker dockerd.

3. Efetue login em seu cluster a partir da CLI do IBM Cloud Private e efetue login no registro de imagem privado do Docker:

    ```
    cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
    docker login <cluster_CA_domain>:8500
    ```

    Em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal. O `cluster_CA_domain` é o domínio da autoridade de certificação (CA). Se você não especificou um domínio de CA, o valor padrão é `mycluster.icp`. Consulte Especificando sua própria autoridade de certificação (CA) para serviços IBM Cloud Private.

4. Instale o arquivo do Passport Advantage:

    - Para o Linux ou o Windows, execute o seguinte comando:

        ```
        cloudctl catalog load-archive --archive <compressed_file_name> --registry
        mycluster.icp:8500/namespace
        ```

        O parâmetro `compressed_file_name` é o nome do arquivo que você transferiu por download do Passport Advantage, `--registry` é o registro no qual a imagem do Docker é enviada por push e `namespace` é o namespace do Docker que hospeda a imagem do Docker.

- Para o macOS, execute o comando a seguir:

  ```
  cloudctl catalog load-archive --archive <compressed_file_name> --username user --password
  pass --registry mycluster.icp:8500/namespace
  ```

Nota: o parâmetro `--registry` não é necessário. Se o parâmetro `--registry` não for especificado, as imagens serão transferidas por upload para o registro associado ao domínio de CA de cluster padrão e ao namespace de destino atual.

5. Visualize os gráficos no IBM Cloud PrivateCatalog:

   Na IBM Cloud Private do IBM Cloud Private , clique em **Catálogo** . Os novos gráficos de Helm são carregados no Catalog e é possível instalá-los em seu cluster.

   Nota: é possível carregar os gráficos de Helm usando apenas o Catalog. Não é possível carregar os gráficos usando a CLI de Helm.

Instalando o software IBM na plataforma IBM Cloud Private

É possível usar o instalador do IBM Cloud Private para instalar e ativar o software IBM imediatamente no cluster.

Antes de iniciar, você deve ter um cluster do IBM Cloud Private ou do IBM Cloud Private-CE instalado que esteja na versão 3.2.0 ou mais recente.

Para instalar o software IBM no IBM Cloud Private:

1. Faça download do arquivo compactado a partir do IBM Passport Advantage®.
2. Opcional: prepare o arquivo de recursos yaml de dependência de archive.
3. Opcional: crie um diretório chamado `resources` em seu diretório `cluster` e coloque o arquivo de recursos yaml nele.

   ```
   mkdir -p cluster/resources
   mv your-resource.yaml cluster/resources/
   ```

4. Crie um diretório chamado `addon` em seu diretório `cluster` executando o seguinte comando:

   ```
   mkdir -p cluster/addon
   ```

5. Mova o pacote de archive para o diretório recém-criado. Por exemplo:

   ```
   mv software-version.tgz cluster/addon/
   ```

6. Atualize o arquivo `cluster/config.yaml` incluindo o seguinte conteúdo. Por exemplo:

   ```
   archive_addons:
     software_name:
       namespace: default
       repo: local-charts
       path: addon/software-version.tgz

       charts:
         - name: software
           values:
             service:
             name: software-service
   ```

   A lista a seguir descreve o conteúdo no exemplo:

   - *namespace*: O namespace onde você deseja fazer upload de imagens do IBM Passport Advantage e instalar os gráficos do IBM Passport Advantage.
   - *repo*: O repositório do Helm onde você deseja fazer upload dos gráficos do IBM Passport Advantage. Isso pode ser `local-charts` ou `mgmt-charts`.
   - *path*: o caminho do pacote do IBM Passport Advantage relativo ao diretório `cluster`.
   - *charts*: uma lista de gráficos que você deseja instalar. Se nenhum gráfico estiver listado, nenhum gráfico está instalado.
   - *charts[].name*: O gráfico do Passport Advantage que é instalado com a instalação do IBM Cloud Private.
   - *charts[].values*: os valores customizados do gráfico do IBM Passport Advantage.

7. Execute o comando de instalação do IBM Cloud Private. O instalador faz upload dos gráficos e imagens especificados para o repositório e registro do Helm.

Opções de configuração durante a instalação

Esta seção fornece as opções de configuração para o IBM® Cloud Private.

Compartilhando chaves SSH entre nós do cluster

As chaves de shell seguro (SSH) são usadas para permitir conexões seguras entre hosts em um cluster do IBM® Cloud Private.

Antes de instalar um cluster do IBM Cloud Private, configure a autenticação entre os nós de configuração. É possível gerar um par de chaves SSH em seu nó de inicialização e compartilhar essa chave com os outros nós do cluster. Para compartilhar a chave com os nós do cluster, será necessário ter o acesso a uma conta com acesso raiz para cada nó em seu cluster.

Nota: por padrão, uma vez que a instalação de um cluster do IBM Cloud Private é executada com a conta raiz, a conta raiz deve ser ativada para login e para login por meio de ssh. Se você deseja instalar o IBM Cloud Private com uma conta de usuário não raiz que tenha privilégios sudo, no arquivo `config.yaml`, esse usuário deve ser especificado na seção "Configurações do usuário" para todos os parâmetros possíveis, conforme descrito em Configuração de usuário.

Para configurar a autenticação sem compartilhar chaves SSH, configure a autenticação de senha para os nós do cluster. Consulte Configurando a senha de autenticação para nós do cluster.

1. Efetue login no nó de inicialização com uma conta com acesso raiz.
2. Gerar uma chave SSH.

   ```
   ssh-keygen -b 4096 -f ~/.ssh/id_rsa -N ""
   ```

3. Inclua a chave em cada nó do cluster. Os nós do cluster são os nós principais, de trabalhador, de proxy, de gerenciamento e do Vulnerability Advisor (VA). Conclua a etapa a seguir para cada nó do cluster.

   No nó de inicialização, inclua a chave pública SSH no nó do cluster.

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub <user>@<node_ip_address>
   ```

   Em que `<user>` é o nome de usuário para o nó e `<node_ip_address>` é o endereço IP do nó do cluster.

4. Na pasta `/<installation_directory>/cluster`, substitua o arquivo `ssh_key` pelo arquivo de chave privado que é usado para se comunicar com os outros nós do cluster. Para obter informações adicionais, consulte Compartilhando chaves SSH entre nós do cluster. Execute o comando a seguir:

   ```
   sudo cp ~/.ssh/id_rsa./cluster/ssh_key
   ```

Nesse exemplo, `~/.ssh/id_rsa` é o local e o nome do arquivo de chave privado.

Configurando a autenticação de senha para os nós do cluster

Use senhas em vez de chaves SSH para permitir conexões seguras entre hosts em um cluster do IBM® Cloud Private.

Antes de instalar um cluster do IBM Cloud Private, deve-se configurar a autenticação entre os nós.

Nota: É possível configurar a autenticação de senha para um usuário em cada nó, usando o arquivo `/<installation_directory>/cluster/config.yaml` ou `/<installation_directory>/cluster/hosts`. Não configure a autenticação de senha usando ambos os arquivos.

Deve-se fornecer senhas para o usuário raiz ou para nomes de usuários que têm acesso raiz.

Para configurar a autenticação sem fornecer a senha para cada nó, gere um par de chaves SSH em seu nó de inicialização e compartilhe essa chave com os outros nós do cluster. Consulte Compartilhando chaves SSH entre os nós do cluster.

Configurando a autenticação de senha usando o arquivo config.yaml

O arquivo `config.yaml` pode ser usado para configurar a autenticação de senha apenas para nós que possuem a mesma senha. Se cada nó possuir uma senha diferente, configure a autenticação de senha usando o arquivos `hosts`.

Inclua os parâmetros de configuração para seu caso de uso no arquivo `/<installation_directory>/cluster/config.yaml`:

Tipo de usuário	Parâmetros de configuração

Raiz	`ansible_user: root`
	`ansible_ssh_pass: SHARED_PASSWORD`
	`ansible_ssh_common_args: "-oPubkeyAuthentication=no"`
	
	Em que
	
	`SHARED_PASSWORD`
	
	é a senha para cada usuário raiz.

Não raiz	`ansible_user: non_root`
	`ansible_ssh_pass: SHARED_PASSWORD`
	`ansible_become: true`
	`ansible_become_password: "{{ ansible_ssh_pass }}"`
	`ansible_ssh_common_args: "-oPubkeyAuthentication=no"`
	
	Em que
	
	`SHARED_PASSWORD`
	
	é a senha para cada usuário.

Configurando a autenticação de senha usando o arquivo `hosts`

O arquivo `hosts` pode ser usado para configurar a autenticação de senha para nós que usam as mesmas senhas ou senhas diferentes.

Inclua os parâmetros de configuração para seu caso de uso no arquivo `/<installation_directory>/cluster/hosts`:

- Para o usuário raiz, anexe cada endereço IP com os parâmetros `ansible_user`, `ansible_ssh_pass`, `ansible_ssh_common_args` e `ansible_port` opcional para o usuário raiz nesse nó, conforme mostrado no código a seguir:

```
[master]
<master_node_IP_address> ansible_user="root" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_ssh_common_args="-oPubkeyAuthentication=no" ansible_port="<PORT_VALUE>"
```

```
[worker]
<worker_node_IP_address> ansible_user="root" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_ssh_common_args="-oPubkeyAuthentication=no" ansible_port="<PORT_VALUE>"

[proxy]
<proxy_node_IP_address> ansible_user="root" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_ssh_common_args="-oPubkeyAuthentication=no" ansible_port="<PORT_VALUE>"

[management]
<management_node_IP_address> ansible_user="root" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_ssh_common_args="-oPubkeyAuthentication=no" ansible_port="<PORT_VALUE>"

[va]
<va_node_IP_address> ansible_user="root" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_ssh_common_args="-oPubkeyAuthentication=no" ansible_port="<PORT_VALUE>"
```

Em que `<NODE_PASSWORD>` é a senha para o usuário raiz nesse nó e `<PORT_VALUE>` é a sua porta SSH customizada.

- Para usuários não raiz, anexe cada endereço IP com os parâmetros `ansible_user`, `ansible_ssh_pass`, `ansible_ssh_common_args`, `ansible_become`, `ansible_become_password` e `ansible_port` opcional, conforme mostrado no seguinte código:

```
[master]
<master_node_IP_address> ansible_user="<USER>" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_become=true ansible_become_password="<NODE_PASSWORD>" ansible_port="<PORT_VALUE>"
ansible_ssh_common_args="-oPubkeyAuthentication=no"

[worker]
<worker_node_IP_address> ansible_user="<USER>" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_become=true ansible_become_password="<NODE_PASSWORD>" ansible_port="<PORT_VALUE>"
ansible_ssh_common_args="-oPubkeyAuthentication=no"

[proxy]
<proxy_node_IP_address> ansible_user="<USER>" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_become=true ansible_become_password="<NODE_PASSWORD>" ansible_port="<PORT_VALUE>"
ansible_ssh_common_args="-oPubkeyAuthentication=no"

[management]
<management_node_IP_address> ansible_user="<USER>" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_become=true ansible_become_password="<NODE_PASSWORD>" ansible_port="<PORT_VALUE>"
ansible_ssh_common_args="-oPubkeyAuthentication=no"

[va]
<va_node_IP_address> ansible_user="<USER>" ansible_ssh_pass="<NODE_PASSWORD>"
ansible_become=true ansible_become_password="<NODE_PASSWORD>" ansible_ssh_common_args="-
oPubkeyAuthentication=no"
```

Em que `<USER>` é o usuário não raiz que tem permissão raiz nesse nó, `<NODE_PASSWORD>` é a senha para esse usuário não raiz e `<PORT_VALUE>` é a sua porta SSH customizada.

Configurando as funções do nó nos arquivos host

O arquivo `hosts` contém o endereço IP dos nós de gerenciamento principais, do trabalhador, de proxy e opcionais e nós do Consultor de Vulnerabilidade em seu cluster.

Para obter mais informações sobre os tipos de nó do IBM® Cloud Private, consulte Arquitetura.

Este arquivo hosts está na pasta `/<installation_directory>/cluster`.

Durante a instalação do IBM Cloud Private, você inclui o endereço IP para os nós principais, do trabalhador e do proxy nesse arquivo. Opcionalmente, é possível especificar um nó de gerenciamento. Depois de instalar o IBM Cloud Private, não é possível modificar os nós principais, de proxy ou de gerenciamento em seu cluster.

Importante: não inclua nomes de host nesse arquivo.

1. Abra o arquivo `/<installation_directory>/cluster/hosts`.
2. Inclua os endereços IP para os diferentes tipos de nó nas diferentes seções do arquivo.
    - Para um ambiente padrão ou do Community Edition, é possível ter apenas um nó na seção principal.

- Para um ambiente de alta disponibilidade (HA), especifique vários hosts nas seções principais e de proxy. É possível especificar qualquer número de nós do proxy, mas devem ser especificados 3 ou 5 nós principais. É possível configurar alta disponibilidade para apenas os nós principais, apenas os nós do proxy ou para ambos os tipos de nó. Consulte Entendendo nós de alta disponibilidade e do proxy para obter informações adicionais.
- É possível ativar o nó de gerenciamento opcional.
- Se você usar um único computador como vários nós em seu cluster, será necessário especificar seu endereço IP em cada seção do nó aplicável. Por exemplo, se você usar o mesmo nó como um principal e proxy, insira seu endereço IP nas seções `principal` e `proxy`.
- É possível ativar o nó etcd opcional. Se você inclui um nó etcd, o `etcd` é instalado nesse nó. Além disso, o `etcd` é instalado no nó principal.

O arquivo `hosts` para um ambiente padrão ou do Community Edition é semelhante ao texto a seguir:

```
[master]
<master_node_IP_address>

[worker]
<worker_node_1_IP_address>
....
<worker_node_n_IP_address>

[proxy]
<proxy_node_IP_address>

[management]
<management_node_1_IP_address>
....
<management_node_n_IP_address>

[va]
<va_node_IP_address>

[etcd]
<etcd_node_IP_address>
```

Nota: se você deseja ativar nós de gerenciamento, será necessário remover o `#` do cabeçalho `[management]`.

O arquivo `hosts` para um ambiente de alta disponibilidade é semelhante ao texto a seguir:

```
[master]
<master_node_1_IP_address>
<master_node_2_IP_address>
<master_node_3_IP_address>

[worker]
<worker_node_1_IP_address>
....
<worker_node_n_IP_address>

[proxy]
<proxy_node_1_IP_address>
<proxy_node_2_IP_address>
<proxy_node_3_IP_address>

[management]
<management_node_1_IP_address>
....
<management_node_n_IP_address>

[va]
<va_node_IP_address>

[etcd]
<etcd_node_IP_address>
```

Nota: se você deseja ativar o Vulnerability Advisor e os nós de gerenciamento, deve-se remover o `#` dos cabeçalhos `[va]` e `[management]`.

1. Em ambientes de HA, também é possível configurar valores para parâmetros específicos de nó nos arquivos host. Por exemplo, é possível configurar os valores de parâmetro `vip_iface`, conforme mostrado no código a seguir:

```
[master]
<master_node_1_IP_address> vip_iface=eth0
<master_node_2_IP_address> vip_iface=ens192
<master_node_3_IP_address> vip_iface=ens160
```

Os valores de parâmetros no arquivo `config.yaml` têm a prioridade mais alta durante uma instalação. Para configurar um valor de parâmetro no arquivo hosts, deve-se remover o parâmetro do arquivo `config.yaml`. Consulte Instalando as edições do IBM® Cloud Private Cloud Native, Enterprise e Community.

Definindo grupos de hosts customizados

Também é possível criar grupos de hosts que podem ser reservados para aplicativos ou processos específicos. Grupos de hosts podem ser definidos durante a instalação ou após a instalação.

Para definir um grupo de hosts customizado:

1. Crie um nome para o grupo de hosts. O nome do grupo de hosts deve estar no formato `hostgroup-customname`.
2. Inclua o IPs para o host que pertencem ao grupo de hosts customizado. Por exemplo, para criar um grupo de hosts customizado que pode ser usado apenas por processos do DB2. Inclua o seguinte no arquivo `hosts`.

   ```
   ......

   [hostgroup-db2]
   <hostgroup_node_1_IP_address>
   <hostgroup_node_2_IP_address>
   <hostgroup_node_3_IP_address>
   ```

3. Implemente o grupo de hosts.
   - Se você estiver criando um grupo de hosts customizado durante a instalação, continue com o procedimento de instalação.
   - Se você estiver criando um grupo de hosts customizado após a instalação, consulte Incluindo nós do cluster.

Após a implementação, os nós do grupo de hosts são designados ao rótulo `customname=true` e contaminações `dedicated=customname:NoSchedule`. Por exemplo, os nós `hostgroup-db2` são designados ao rótulo `db2=true` e contaminações `dedicated=db2:NoSchedule`.

Customizando sua instalação

É possível concluir a maior parte de sua customização de cluster no arquivo `/<installation_directory>/cluster/config.yaml`. Essas customizações devem ser feitas durante a instalação de seu cluster.

Para revisar uma lista completa de parâmetros que estão disponíveis para customizar, consulte Customizando o cluster com o arquivo config.yaml.

Também é possível configurar valores de parâmetros específicos do nó no arquivo `/<installation_directory>/cluster/hosts`. No entanto, os valores de parâmetro que são configurados no arquivo `config.yaml` têm prioridade mais alta durante uma instalação. Para configurar um valor de parâmetro no arquivo `hosts`, deve-se remover o parâmetro do arquivo `config.yaml`. Para obter mais informações sobre a configuração dos valores de parâmetro específicos do nó nos arquivos host, consulte Configurando as funções do nó nos arquivos host.

- Opções gerais de instalação
- Opções de instalação de HA
- Estrutura de diretório do cluster

Opções gerais de instalação

Para customizar seu cluster, as seguir há alguns dos recursos opcionais que podem ser configurados durante a instalação.

1. Configure o serviço de monitoramento. Consulte Configurando o serviço de monitoramento. **Importante:** se você usar uma configuração de alta disponibilidade, siga as etapas de configuração de serviço de monitoramento para permitir que o serviço de monitoramento use um provedor de armazenamento compartilhado de rede.
2. Especifique uma autoridade de certificação (CA) para seu cluster. Consulte Especificando sua própria autoridade de certificação (CA) para serviços IBM Cloud Private.

3. Ative o Vulnerability Advisor. Consulte Ativando o Vulnerability Advisor. Esse recurso não está disponível para o IBM Cloud Private-CE (Community Edition).
4. Configure o armazenamento do GlusterFS. Consulte Configurando GlusterFS durante a IBM Cloud Private instalação .
5. Configure o vSphere Cloud Provider. Consulte Configurando um vSphere Cloud Provider.
6. Configure o AWS Cloud Provider. Consulte Configurações do AWS Cloud Provider.
7. Crie uma ou mais classes de armazenamento para os fornecedores de armazenamento em seu ambiente. Consulte Fornecimento de armazenamento dinâmico.
8. Criptografe o tráfego de rede de dados do cluster com IPsec. Consulte Criptografando o tráfego de rede de dados do cluster com IPsec.
9. Criptografe volumes. Consulte Criptografando volumes usando dm-crypt .
10. Integre o VMware NSX-T 2.4 com os nós do cluster do IBM Cloud Private. Consulte Integrando o VMware NSX-T 2.4 com o IBM Cloud Private.
11. Especifique um diretório de armazenamento do Docker padrão. Consulte Especificando um diretório de armazenamento do Docker padrão usando montagem bind.
12. Especifique outros diretórios de armazenamento padrão. Consulte Especificando outros diretórios de armazenamento padrão usando a montagem bind.
13. Configure um balanceador de carga externo. Veja Configurando um balanceador de carga externo.

Opções de instalação de HA

Assegure-se de revisar o tópico Clusters de alta disponibilidade do IBM® Cloud Private. A HA é suportada apenas pelas edições IBM Cloud Private Cloud Native e Enterprise.

Para clusters de HA, configure os parâmetros de HA:

1. Para configurar a HA para seus nós principais, atualize a seção `Configurações de HA` no arquivo `config.yaml`. Para o valor de parâmetro **vip_iface**, forneça o nome da interface de seu ambiente. Para o valor de parâmetro **cluster_vip**, forneça um endereço IP disponível, de preferência um do mesmo intervalo de IPs usado pelos nós do cluster. Para os nós principais, o IP virtual deve estar na mesma sub-rede. A configuração se assemelha ao código a seguir:

   ```
   # HA settings
   vip_iface:  eth0
   cluster_vip:  5.5.5.1
   ```

2. Para configurar a HA para os nós do proxy, atualize a seção `Configurações de proxy` no arquivo `config.yaml`. Para o valor de parâmetro **proxy_vip_iface**, forneça o nome da interface de seu ambiente. Para o valor de parâmetro **proxy_vip**, forneça um endereço IP disponível, de preferência um do mesmo intervalo IP usado pelos nós do cluster. A configuração se assemelha ao código a seguir:

   ```
   # Proxy settings
   proxy_vip_iface:  eth0
   proxy_vip:  5.5.5.2
   ```

   Nota: deve-se usar endereços IP diferentes para os valores de parâmetro **cluster_vip** e **proxy_vip**.

   Nota: em um ambiente de HA do IBM Cloud Private, não use um endereço IP que termine com dígito 0 para **proxy_vip**, por exemplo, x.y.z.0, quando **vip_manager** é configurado como **keepalived** ou **ucarp**.

   Nota: também é possível configurar esses parâmetros específicos do nó nos arquivos host. Por exemplo, é possível configurar os valores de parâmetro **vip_iface**, conforme mostrado no código a seguir:

   ```
   [master]
   <master_node_1_IP_address> vip_iface=eth0
   <master_node_2_IP_address> vip_iface=ens192
   <master_node_3_IP_address> vip_iface=ens160
   ```

Os valores de parâmetros no arquivo `config.yaml` têm prioridade mais alta durante uma instalação. Para configurar um valor de parâmetro no arquivo hosts, deve-se remover o parâmetro do arquivo `config.yaml`.

Estrutura de diretório do cluster

O diretório do cluster contém os arquivos a seguir:

- **config.yaml**: as definições de configuração que são usadas para instalar o IBM Cloud Private em seu cluster.
- **hosts**: a definição dos nós em seu cluster.

- **misc/storage_class**: uma pasta que contém as definições de classe de armazenamento dinâmico para seu cluster.
- **ssh_key**: um arquivo de item temporário para a chave privada SSH que é usada para se comunicar com outros nós no cluster.
- **runtime-engine**: Contém os pacotes do Docker do IBM Cloud Private que podem ser usados para instalar o Docker nos nós do cluster.

Customizando o cluster com o arquivo config.yaml

O arquivo `config.yaml` contém todas as definições de configuração que são necessárias para implementar seu cluster.

No arquivo `config.yaml`, é possível customizar sua instalação usando vários parâmetros.

O `config.yaml` também contém uma lista de imagens do Docker que são obtidas do Docker Hub pelo instalador durante o processo de instalação do IBM® Cloud Private-CE (Community Edition). Para uma instalação do IBM Cloud Private-CE (Community Edition), também é possível armazenar essas imagens de instalação em um registro de imagem privado em vez de obter diretamente do Docker Hub. Se as imagens estiverem armazenadas em um registro de imagem privado, atualize o arquivo `config.yaml` para apontar para as imagens de instalação em seu registro de imagem privado. Para uma instalação do IBM Cloud Private, essas imagens do Docker do instalador são comentadas pois as imagens estão disponíveis no pacote do instalador transferido por download.

Nota: antes de atualizar qualquer seção do arquivo `config.yaml`, revise os comentários sequenciais nessa seção.

É possível configurar ou atualizar os parâmetros a seguir modificando o arquivo `config.yaml`.

1. Abra o arquivo `/<installation_directory>/cluster/config.yaml`.
2. Inclua ou modifique os parâmetros e valores. O formato para definir um parâmetro e valores é `<parameter_name>: <value>`.

Configurações gerais

Parâmetro	Descrição	Valor Padrão
`offline_pkg_copy_path`	O diretório para manter os arquivos de instalação temporários durante a instalação off-line. Esse local deve ter pelo menos 50 GB de espaço em disco disponível. Se o diretório `/tmp` tem menos de 50 GB de espaço, deve-se configurar esse parâmetro para um local que tenha o requisito de espaço em disco disponível.	`/tmp`

Parâmetro	Descrição	Valor Padrão
`firewall_enabled`	Se configurado como `true`, o instalador mantém o firewall do sistema operacional local em execução e abre as portas requeridas pelo IBM Cloud Private no firewall.	false
`wait_for_timeout`	Esse parâmetro especifica o valor de tempo limite padrão para operações. Uma configuração de 3600 é ideal na maioria dos ambientes.	600
`router_http_port`	A porta http que é usada pelo ingresso de gerenciamento do	

IBM Cloud Private, que age como um proxy para todos os serviços de gerenciamento do IBM Cloud Private.|8080| | `router_https_port`|A porta https que é usada pelo ingresso de gerenciamento do IBM Cloud Private, que age como um proxy para todos os serviços de gerenciamento do IBM Cloud Private.|8443| | `ingresso: http_port`|A porta http que é usada pelo controlador de ingresso NGINX, que age como um proxy para todos os serviços do usuário.|80| | `ingresso: https_port`|A porta https que é usada pelo controlador de ingresso NGINX, que age como um proxy para todos os serviços do usuário.|443| | `loopback_dns`|Configure esse parâmetro como `true` se o nó de inicialização usar um IP de loopback, por exemplo, um endereço IP que começa com 127, como o servidor DNS.|false| | `fips_enabled`|Configure esse parâmetro como `true` para ativar a conformidade do Federal Information Processing Standard (FIPS) 140-2 para ingresso de gerenciamento do IBM Cloud Private (console de gerenciamento), controlador de ingresso NGINX (serviço de ingresso), gerenciador de imagem e configuração de autorização do provedor de identidade.|`false`| | `ansible_python_interpreter`|Configure esse parâmetro como `/usr/bin/python3` se você usar o python3 em seus nós do cluster. |`/usr/bin/python`| | `bootstrap_token_ttl`|Duração para a qual o token de autoinicialização é válido.|`"24h0m0s"`| | `upload_chart_enabled`|Configure este parâmetro como `true` para ativar o upload do gráfico para o repositório do Helm quando o serviço de gerenciamento `mgmt-repo` estiver ativado.|`true`|

Configurações do Kubernetes

Parâmetro	Descrição	Valor Padrão
`kube_apiserver_extra_args`	Define configurações de apiserver extras para Kubernetes. Aceita uma lista de argumentos de apiserver que são fornecidos no formato `--key=value`. **Nota**: ao configurar o plug-in de controle de admissão	

`AlwaysPullImages`, os usuários devem extrair todas as imagens de seus próprios registros. |Nenhum| | `kube_apiserver_secure_port`|Configura a porta segura do apiserver do Kubernetes.|8001| | `kube_controller_manager_extra_args`|Define configurações extras do controlador para Kubernetes. Aceita uma lista de argumentos do controlador que são fornecidos no formato `--key=value`.|Nenhum| | `kube_proxy_extra_args`|Define configurações extras de proxy para Kubernetes. Aceita uma lista de argumentos de proxy que são fornecidos no formato `--key=value`.|Nenhum(a)| | `kube_scheduler_extra_args`|Define configurações extras do planejador para Kubernetes. Aceita uma lista de argumentos do planejador que são fornecidos no formato `--key=value`.|Nenhum| | `kubelet_extra_args`|Define configurações extras para kubelet. Aceita uma lista de argumentos de kubelet que são fornecidos no formato `--key=value`. Por exemplo, para configurar o número máximo de pods que podem ser executados em um kubelet, defina a configuração a seguir:`kubelet_extra_args: ["--max-pods=110"]`|Nenhum(a)| | `auditlog_enabled`| Ativa o log de auditoria do Kubernetes, que registra a sequência cronológica de atividades por usuários individuais, administradores ou outros componentes do sistema que modificaram o sistema. Configure esse parâmetro como `true` para ativar o log de auditoria.|false| | `journal_path`|Configura o caminho padrão para armazenar dados do log de auditoria.|Caminho para o diretório: `/run/log/journal`|

Configurações de log

Tabela 3. Configurações de log

Parâmetro	Descrição	Valores	Valor Padrão
`metrics_max_age`	Configura o número máximo de dias para armazenar as métricas do sistema e do aplicativo. As métricas mais antigas do que esse número especificado de dias são removidas. As métricas são removidas às 23h59min no dia especificado.	Número de dias	1
`logs_maxage`	Configura o número máximo de dias para armazenar logs no Elasticsearch.	Número de dias	1
`docker_config: log-opts:`	Opções de logs do Docker.		

Parâmetro	Descrição	Valores	Valor Padrão
`max-size`	Configure o tamanho máximo dos arquivos de log do Docker.		"100m"
`max-file`	Configure o número máximo de arquivos de log do Docker.		"10"

Configurações de rede

Parâmetro	Descrição	Valores	Valor Padrão
`network_type`	Tipo de gerenciamento de rede em seu cluster.	`calico` ou `nsx-t`	`calico`
`network_helm_chart_path`	Caminho absoluto para o gráfico helm da rede.	Nenhum	Nenhum
`calico_ipip_mode`	Permite que o Calico seja executado no modo IP sobre IP. Essa configuração é necessária quando os nós do trabalhador estão em sub-rede diferente e o BGP não está ativado em roteadores entre os nós do trabalhador. Essa configuração também é necessária em alguns ambientes de nuvem, como o OpenStack, em que as máquinas virtuais não podem trabalhar como roteadores.	• `Always`: o IP na malha de túnel de IP é criado nos nós do cluster para

comunicação entre pods.

- `Never`: o IP na malha de túnel IP não é criado nos nós do cluster para comunicação entre pods.
- `CrossSubnet`: o IP na malha de túnel IP é criado para os nós que estejam em sub-redes diferentes.

|`Sempre`| | `calico_tunnel_mtu`|O IPIP para Calico tem uma MTU padrão de 1430. Se a interface principal de seu host tiver uma MTU que seja menor que 1450, o IPIP do Calico terá desempenho fraco. Configure a MTU de modo que a MTU da interface principal do host menos a MTU padrão do túnel IPIP do Calico seja maior ou igual a 20. |Números inteiros positivos|1430| | `network_cidr`|A rede IPv4 a ser usada para a rede inteira. Esse valor deve estar no formato CIDR. Quando você criar um network_cidr, assegure-se de selecionar um intervalo de IPs que não entre em conflito com a rede do host existente ou com o service_cluster_ip_range. Na maioria dos ambientes, é possível usar o valor padrão.
Importante: Não é possível modificar esse parâmetro após a instalação, pois todos os serviços de gerenciamento de plataforma irão usar esses intervalos de IP e o Kube-proxy também está ciente desse cidr de rede.|Endereço IP no formato do CIDR|10.1.0.0/16| | `calico_ip_autodetection_method`|É possível configurar o nó Calico para detectar automaticamente o endereço IP que é usado para rotear entre nós. É possível usar um dos seguintes métodos:

- `calico_ip_autodetection_method: first-found`: esse método usa o primeiro endereço IP válido em uma interface válida que é localizada primeiro.
- `calico_ip_autodetection_method: interface`: esse parâmetro aceita uma lista separada por vírgula de nomes de expressão regular como valor. Ele usa o primeiro endereço IP que está localizado na interface especificada.

Exemplos:

- `calico_ip_autodetection_method: interface=eth0`
- `calico_ip_autodetection_method: interface=eth.*`
- `calico_ip_autodetection_method: interface=eth.*,ens.*`

- `calico_ip_autodetection_method: can-reach=<remote IP address or host name>`: O método `can-reach` usa seu roteamento local para determinar o endereço IP que é usado para acessar o destino especificado. Esse parâmetro aceita um endereço IP remoto ou o nome de domínio como valor.

Nota:

- Em um ambiente com múltiplas interfaces de rede (NICs), use o método `can-reach` para especificar a rede a ser usada para suas cargas de trabalho. No IBM Cloud Private, é possível configurar `calico_ip_autodetection_method: can-reach=<Master node IP address>`.

- Os nomes de interface de rede não podem conter as sequências a seguir: `"docker.*", "cbr.*", "dummy.*", "virbr.*", "lxcbr.*", "veth.*", "lo", "cali.*", "tunl.*" ou "flannel.*"`.
- Alguns IPs não são reconhecidos pelo Calico. Assegure-se de que suas interfaces não tenham IPs nos intervalos a seguir:
  - `10.0.2.0/24` - este intervalo de IPs é o intervalo de endereços de interface NAT vagrant/virtualbox padrão.
  - `192.168.122.0/24` - esse intervalo de IPs é o intervalo de endereços da interface da MV libvirt padrão.

|

- `first-found`

- `interface=INTERFACE-REGEX`

- `can-reach=<remote IP address or domain name>`

|`can-reach={{ groups['master'][0] }}`|| `service_cluster_ip_range`|O intervalo de IPs do cluster de serviço do Kubernetes. Essa configuração aloca um bloco de IPs para serviços. Ao criar um service_cluster_ip_range, assegure-se de selecionar um intervalo de IPs que não entre em conflito com a rede do host existente ou com o network_cidr. Service_cluster_ip_range é uma rede virtual. Na maioria dos ambientes, é possível manter o valor padrão. |Endereço IP no formato do CIDR|10.0.0.0/16|

Configurações de acesso ao cluster

Parâmetro	Descrição	Valores	Valor Padrão
`cluster_name`	O nome de seu cluster. Em um ambiente com múltiplos clusters, especifique um nome distinto para cada cluster. `cluster_name` deve consistir apenas em caracteres alfanuméricos minúsculos. **Nota:** seu `cluster_name` ativo aparece apenas no cabeçalho da		

Plataforma IBM Cloud após a instalação do IBM Multicloud Manager.

Configurações do Cluster

Tabela 6. Configurações de cluster

Parâmetro	Descrição	Valor Padrão
`single_cluster_mode:`	Se for configurado como false, o instalador implementará o hub IBM Multicloud Manager inteiro e o cluster poderá utilizar os recursos completos do IBM Multicloud Manager e gerenciar outros clusters.	`true`

Configurações do Docker

Nota: essas configurações podem ser definidas somente para os pacotes do Docker do IBM Cloud Private fornecidos. Consulte IBM Cloud Private Pacotes do Docker.

Parâmetro	Descrição	Formato	Valor Padrão
`docker_version`	Especifique a versão do Docker que você deseja instalar. Um pacote para a versão necessária deve estar disponível no diretório `/<installation_directory>/cluster/runtime-engine`.	sequência	18.06.2
`install_docker`	Permite que o instalador instale automaticamente o Docker em seus nós do cluster.	true ou false	true
`docker_env`	Configura o ambiente para o Docker. Por exemplo, é possível configurar um		

local `https_proxy`, se o Docker for executado atrás de um firewall. O local do ambiente é armazenado no arquivo `/etc/systemd/system/docker.service.d/docker-env-icp.conf`.|`"HTTP_PROXY=http://proxy-server:port/", "HTTPS_PROXY=http://proxy-server:port", "NO_PROXY=localhost,127.0.0.1,{{ cluster_CA_domain }}"` |Nenhum|

Configurações de HA do proxy

Tabela 8. Configurações de AD do Proxy

Parâmetro	Descrição	Valores	Padrão
`proxy_vip_iface`	Configura a interface IP virtual para um ambiente de HA do nó do proxy.	`eth0`	N/D
`proxy_vip`	Configura o endereço IP virtual para um ambiente de HA do nó do proxy.	`127.0.1.1` Não especifique a sub-rede no endereço IP.	N/D
`vip_manager`	Especifique o gerenciador de VIP a ser usado para o nó principal ou do proxy em um ambiente de AD.	É possível especificar `etcd` ou `keepalived` como o `vip_manager`.	`etcd`

Principais configurações de HA

Tabela 9. Principais configurações de HA

Parâmetro	Descrição	Valores	Padrão
`cluster_vip`	Configura o endereço IP virtual para o ambiente de HA do IBM Cloud Private.	`127.0.1.1` Não especifique a sub-rede no endereço IP.	N/D
`vip_iface`	Configura a interface IP virtual para o ambiente de HA do IBM Cloud Private.	`eth0`	N/D
`vip_manager`	Especifique o gerenciador de VIP a ser usado para o nó principal ou do proxy em um ambiente de AD.	É possível especificar `etcd` ou `keepalived` como o `vip_manager`.	`etcd`

Configurações do Usuário

Tabela 10. Configurações do Usuário

Parâmetro	Descrição	Padrão

`ansible_user`
`ansible_become`
`ansible_become_password`
	IBM Cloud Private usa o valor de parâmetro `ansible_user` para acessar seus nós de cluster durante a instalação. Se você usar uma conta não de administrador que tenha privilégios `sudo` para se conectar a um nó principal ou do trabalhador, configure o `ansible_user` com o nome de usuário e `ansible_become` como `true`.
	Se você executa `sudo` com uma senha, deve-se configurar o parâmetro `ansible_become_password` para o valor de sua senha não raiz (usuário sudo). Essa variável é opcional se você configura `NOPASSWD` no arquivo `/etc/sudoers`.
	Para configurar esses valores de parâmetros, consulte Configurando a autenticação de senha para nós do cluster.
		Nenhum

`default_admin_user`	Configura um nome do usuário administrador do cluster customizado. Se o LDAP estiver ativado e você tiver um usuário LDAP com o nome de usuário `admin`, mude o valor de parâmetro `default_admin_user` para algo diferente para evitar um conflito de nome do administrador do cluster e de nome do usuário LDAP.	usuário
`default_admin_password`	Este parâmetro configura a senha do administrador de cluster padrão necessária. A senha deve satisfazer todas as expressões regulares que são especificadas em `password_rules`. Se o LDAP estiver ativado, esse parâmetro configurará a senha do administrador LDAP.	N/D

Parâmetro	Descrição	Padrão
password_rules	Define uma lista de uma ou mais regras de senha como expressões regulares que o default_admin_password deve transmitir. Se não estiver definido, o default_admin_password deverá atender à regra de cumprimento de senha padrão '^([a-zA-Z0-9\-]{32,})$': a senha deve ter no mínimo 32 caracteres de comprimento e pode conter apenas caracteres alfanuméricos minúsculos e maiúsculos e traços. As regras devem ser definidas em uma lista de matrizes. Para obter informações adicionais e um exemplo, consulte Customizar seu cluster.	- '^([a-zA-Z0-9\-]{32,})$'

Configurações do GlusterFS

Tabela 11. Configurações do GlusterFS

Parâmetro	Descrição
storage-glusterfs	Armazenamento de provisão em nós dedicados. Para configurar o GlusterFS, deve-se configurar vários parâmetros. Consulte, Configurando o GlusterFS durante a IBM Cloud Private instalação .

Configurações do Cloud Provider

Parâmetro	Valores	Descrição
cloud_provider	vsphere	**Configurações do vSphere Cloud Provider** Configura o vSphere Cloud Provider. Para configurar o vSphere Cloud Provider, deve-se configurar vários parâmetros. Consulte Configurando um vSphere Cloud Provider

para obter mais informações.

Criptografando o tráfego de rede de dados do cluster com o IPsec

Tabela 13. Criptografando o tráfego de rede de dados do cluster com o IPsec

Parâmetro	Descrição
ipsec_mesh:	Ativa o IPsec. Para configurar o IPsec, deve-se configurar vários parâmetros. Consulte Criptografando o tráfego de rede de dados do cluster com IPsec.

Configurações do serviço de gerenciamento

Parâmetro	Descrição	Valor Padrão
management_services	Use esse parâmetro para desativar os serviços de gerenciamento. É	

possível desativar os seguintes serviços: key-management, audit-logging, custom-metrics-adapter, image-security-enforcement, istio, metering, logging, monitoring, service-catalog, storage-minio, storage-glusterfs, vulnerability-advisor, node-problem-detector-draino, and multicluster-endpoint. Para obter informações adicionais sobre serviços de gerenciamento, consulte Ativando e desativando serviços de gerenciamento do IBM Cloud Private.

Integrando o VMware NSX-T 2.4 ao IBM Cloud Private

Tabela 15. Integrando o VMware NSX-T 2.4 ao IBM Cloud Private

Parâmetro	Descrição
network_type:	Configure o valor como nsx-t.
nsx_t:	Deve-se configurar vários parâmetros para integrar o VMware NSX-T 2.4 ao IBM Cloud Private. Consulte Integrando o VMware NSX-T 2.4 com o IBM Cloud Private.

Isolamento do

Parâmetro	Descrição
`isolated_namespaces: [ ]`	Use esse parâmetro se desejar isolar um namespace. Inclua um

namespace e atribua um grupo de hosts com nós de trabalhador dedicados ao namespace. A sintaxe é `[{namespace: <namespace_name>, hostgroup: <hostgroup_name>}]`. Por exemplo, `[{namespace: devops, hostgroup: dev}, {namespace: production, hostgroup: prod}]`. Os namespaces são criados durante a instalação do IBM Cloud Private . | | `isolated_proxies: [ ]`|Use esse parâmetro se desejar atribuir grupos de hosts do nó do proxy a um namespace. Inclua um namespace e atribua um grupo de hosts com nós de proxy dedicados ao namespace. A sintaxe é `[{namespace: <namespace_name>, hostgroup: <hostgroup_name>, lb_address: <load_balancer_IP_address>}]`. Por exemplo, `[{namespace: devops, hostgroup: proxydev, lb_address: 172.68.20.11}, {namespace: production, hostgroup: proxyprod}]`. Se você estiver especificando um namespace que não esteja definido no parâmetro `isolated_namespaces`, deverá criar manualmente o namespace após instalar o IBM Cloud Private.|

configurações etcd

Parâmetro	Descrição	Valor Padrão
`etcd_extra_args`	Configura configurações de etcd extras. Aceita uma lista de argumentos que são fornecidos no formato --key=value.	["--grpc-keepalive-timeout= 0", "--grpc-keepalive-keepalive-interval= 0", "--snapshot-count=10000"]
`etcd_data_dir`	Diretório de dados etcd. Por exemplo, `/var/lib/etcd`.	`/var/lib/etcd`
`etcd_wal_dir`	Diretório etcd wal. Por exemplo, `/var/lib/etcd-wal`. Seria possível configurar o diretório para um diretório de log remoto centralizado para criação de log persistente. O	

valor padrão que o etcd usa é `-- max-wals= 5`.| `/var/lib/etcd-wal`|

Configurações de segurança de Istio add-ons

Tabela 18. Configurações de segurança de Istio add-ons

Parâmetro	Descrição	Valor Padrão
`istio_addon`	Configure as configurações de segurança do Istio add-ons.	
`grafana`	Configure o complemento Grafana Istio para visualizar dados de tráfego de malha de serviço. Configure o nome de usuário e a senha para acessar a interface.	Nenhum
`kiali`	Configure o complemento Kiali Istio para visualizar dados de microsserviços na malha de serviço Istio. Configure o nome de usuário e a senha para acessar a interface.	Nenhum

Controlador de ingresso NGINX

Parâmetro	Descrição
`nginx-ingress: ingresso`	Customize a configuração do controlador de ingresso NGINX. O controlador de ingresso é ativado por padrão. Para obter uma lista de parâmetros configuráveis, consulte

Opções de configuração do Kubernetes nginx-ingresso.| | `config:` | | `disable-access-log`|Desativa o log de acesso. Configure o valor como `true` ou `false`.| | `keep-alive-requests`|Configura o número máximo de solicitações que podem ser entregues por meio de uma conexão keep-alive. Por exemplo, 10000.| | `upstream-keepalive-conexões`|Ativa o cache para conexões com servidores de envio de dados. O parâmetro de conexões configura o número máximo de conexões keepalive inativas para os servidores de envio de dados que são preservados no cache de cada processo do trabalhador. Quando esse número é excedido, as conexões usadas menos recentemente são encerradas. Por exemplo, 64.| | `worker-processes`|Configura o número de processos do trabalhador. Por exemplo, 2.| | `extraArgs:` | | `publish-status-address`|Especifique o endereço que o controlador de ingresso deve usar. Por exemplo, "{{proxy_external_address }}".| | `enable-ssl-passthrough`|Envie as conexões TLS diretamente para o pod em vez de permitir que NGINX decriptografe a comunicação. Configure o valor como `true` ou `false`.|

Configurações de Minio

Tabela 20. Configurações de Minio

Parâmetro	Descrição
`storage-minio`	Para configurar o Minio, deve-se configurar vários parâmetros. Consulte, Configurando o Minio durante a IBM Cloud Private instalação .

Configurações de segurança MongoDB

Tabela 21. Configurações de segurança MongoDB

Parâmetro	Descrição	Valor Padrão
`mongodb_credentials:`	Configure credenciais de acesso MongoDB.	
`admin_user`	Configure o nome do usuário administrador.	`admin`
`admin_password`	Configure a senha do administrador. Se você não especificar uma senha, o instalador gerará uma sequência aleatória e a configurará como a senha.	Sequência gerada aleatoriamente

Configurações de terminal multicluster

Parâmetro	Descrição	Valor Padrão
`multicluster-endpoint:`	Configure parâmetros do terminal multicluster.	
`global:`	Configure parâmetros globais multicluster.	
`clusterName`	O nome do cluster exibido no multicluster-hub. O parâmetro `cluster_name` ou `local-cluster` ao implementar em um multicluster-hub.| "`{{ cluster_name }}`"| `clusterNamespace`|O namespace ao qual o cluster pertencerá no multicluster-hub. O parâmetro `cluster_name` ou `local-cluster` ao implementar em um multicluster-hub.| "`{{ cluster_name }}`"| `clusterLabels:`| Forneça rótulos de cluster.|| `environment:`|Rótulo do ambiente em cluster.|`Dispositivo`| `região:`|Região na qual o cluster é configurado.|`CEP: 22296-903`| `datacenter:`|Local do data center no qual o cluster é configurado.|`Toronto`| `owner:`|Proprietário do cluster.|`marketing`| `operator:`|Configuração do operador Klusterlet.|| `bootstrapConfig:`|Informações para registro com multicluster-hubs.|| `hub0:`|Forneça informações do hub0.|| `name:`| Forneça o nome do hub0.|`hub0`| `secret:`|Forneça o segredo de autoinicialização que contém o kubeconfig para conexão com o hub0.|`kube-system/klusterlet-bootstrap`| `hub1:`|Forneça informações hub1.|| `name:`| Forneça o nome do hub1.|Nenhum| `secret:`| Forneça o segredo de autoinicialização que contém o kubeconfig para conectar-se ao hub1.|Nenhum| `klusterlet:`| Forneça os parâmetros de configuração do Klusterlet.|| `host:`| Forneça o nome do host para o ingresso ou rota do serviço Klusterlet.|Nenhum| `prometheusIntegration:`||| `enabled:`| Ative ou desative o Prometheus.|`true`| `policy:`||| `cemIntegration:`| Ative ou desative a integração do CEM.|`false`| `topology:`||| `enabled:`| Ative ou desative a topologia.|`true`| `serviceRegistry:`||| `enabled:`| Ative ou desative um registro de serviço.|`false`| `dnsSuffix:`| Forneça o sufixo DNS.|`mcm.svc`| `plugins:`| Forneça os plug-ins.|`kube-service`|	

Configurando os nós do cluster para a instalação automática do Docker

Configure o IBM Cloud Private para instalar o Docker automaticamente em seus nós do cluster durante a instalação. Os nós do cluster são os nós principal, proxy, trabalhador, consultor de vulnerabilidade e gerenciamento.

Durante a instalação, o Docker é instalado automaticamente nos nós do cluster do Red Hat Enterprise Linux (RHEL) ou do Ubuntu. O pacote do Docker que é usado para instalação nos nós do cluster está localizado na pasta `/<installation_directory>/cluster/runtime-engine` no nó de inicialização. Esse processo instala o Docker em nós que ainda não têm uma versão do Docker instalada. Se você tiver uma versão do Docker que é suportada pelo IBM Cloud Private instalada em seu nó, será possível ignorar esse procedimento.

Para os nós do SUSE Linux Enterprise Server (SLES), deve-se instalar manualmente o Docker usando as instruções de instalação do Docker na documentação do SLES.

Para permitir a instalação automática do Docker em nós do cluster:

1. Em seus nós principais, de proxy, do trabalhador, de gerenciamento e do VA, assegure-se de que seu gerenciador de pacote esteja configurado para permitir atualizações do pacote. Os gerenciadores de pacotes incluem `RPM` para RHEL e `Apt` para Ubuntu.

2. Se você deseja mudar o local do diretório de armazenamento padrão do Docker, deve-se configurar uma montagem bind para o novo diretório antes de instalar o IBM Cloud Private. Veja Especificando um diretório de armazenamento padrão do Docker para Docker instalado automaticamente.
3. (Opcional) Determine quais opções do Docker seus nós principais, de proxy, do trabalhador, de gerenciamento e do VA requerem. Você especifica essas opções no arquivo config.yaml.

Clusters de alta disponibilidade do IBM® Cloud Private

É possível configurar a alta disponibilidade (HA) para os nós principal e de proxy do IBM Cloud Private.

É possível configurar a HA apenas para os nós principais, apenas para os nós de proxy ou para os dois tipos de nó. Para reduzir os requisitos de infraestrutura do seu cluster, é possível designar funções de proxy e principal aos nós de HA. Para os nós principais, o gerenciador de IP virtual deve estar na mesma sub-rede.

Nota: para assegurar a disponibilidade, configure mais de um nó do proxy e 3 ou 5 nós principais.

Deve-se configurar o armazenamento compartilhado em seus nós principais. O IBM Cloud Private requer armazenamento compartilhado para o registro do Docker. O armazenamento deve ser um sistema de arquivo compartilhado compatível com POSIX que está localizado fora de seu cluster do IBM Cloud Private. Seus nós principais devem ter acesso de leitura/gravação ao sistema de arquivos. O sistema de arquivos deve ser montado como hostPath local em seus nós principais. O diretório a seguir deve ser montado em seu armazenamento compartilhado:

- `/var/lib/registry` - O diretório `/var/lib/registry` armazena imagens no registro de imagem privado e mantém as imagens que estão sincronizadas em todos os nós principais.

Nota: deve-se configurar o parâmetro `file` como `0755` para o diretório.

Requisitos para HA principal

Estes requisitos são apenas para HA principal. A HA do proxy não é afetada pelo número de nós principais.

Para N número de principais em um cluster, o cluster pode tolerar até (N-1)/2 falhas permanentes. Por exemplo, em um cluster que possui três principais, se um principal falhar, a tolerância a falhas será como (3-1)/2=1. Você deve apontar para uma tolerância a falhas de uma ou mais.

Deve-se ter um número ímpar de principais em seu cluster. A inclusão de nós principais adicionais fornece uma maior tolerância à falhas. É possível revisar como a tolerância a falhas em um cluster é afetada pelo número de nós principais na *Tabela 1: Tolerância a falhas em clusters de HA*.

Tabela 1. Tolerância a falhas em clusters de HA

Número de nós principais	Tolerância a falhas
1	0
3	1
5	2
7	3

Designação e comunicação do nó em clusters HA

Em clusters do IBM® Cloud Private de HA, o gerenciador de IP virtual controla a designação de nó do proxy e principal.

O gerenciador de IP virtual controla quais nós atendem as funções principal e de proxy designando endereços IP virtuais a esses nós. O gerenciador de IP virtual no IBM Cloud Private facilita a comunicação entre nós por meio do controlador de interface de rede (NIC). O gerenciador de IP virtual designa o endereço IP `cluster_vip` a um nó principal disponível e designa o endereço IP `proxy_vip` a um nó do proxy disponível. Esses nós agem como o nó principal inicial e do proxy. O endereço IP `cluster_vip` deve estar no NIC que você especifica no parâmetro `vip_iface`. Da mesma forma, os endereços IP `proxy_vip` devem estar no NIC que você especifica no parâmetro `proxy_vip_iface`.

O gerenciador de IP virtual monitora o funcionamento dos nós principais e do proxy do cluster. Se o nó principal ou do proxy líder não estiver mais disponível, o gerenciador de IP virtual selecionará um nó disponível e o designará ao endereço IP virtual correto.

Embora o IBM Cloud Private gerencie a HA através do gerenciador de IP virtual, também é possível usar um balanceador de carga externo para distribuir a carga dos nós de proxy e principal e facilitar a comunicação externa. Para usar um balanceador de carga,

durante a instalação, especifique seu endereço IP como os valores de parâmetro `cluster_lb_address` e `proxy_lb_address` no arquivo `config.yaml`.

Nota: para um ambiente de HA, deve-se configurar pelo menos um dos parâmetros a seguir: `cluster_vip`, `cluster_lb_address`.

Para obter mais detalhes sobre como configurar HA durante a instalação, consulte Opções de instalação de HA.

Configurando o serviço de monitoramento

É possível customizar o serviço de monitoramento durante a instalação do IBM® Cloud Private.

Inclua as seguintes linhas de código no arquivo `config.yaml` que está localizado na pasta `/<installation_directory>/cluster`. Customize os parâmetros conforme necessário. Consulte Customizar os Parâmetros . Em seguida, salve e saia do arquivo.

```
monitoring:
  prometheus:
    scrapeInterval: 1m
    evaluationInterval: 1m
    retention: 24h
    persistentVolume:
      enabled: false
      storageClass: "-"
    resources:
      limits:
        cpu: 500m
        memory: 2048Mi
      requests:
        cpu: 100m
        memory: 128Mi
  alertmanager:
    persistentVolume:
      enabled: false
      storageClass: "-"
    resources:
      limits:
        cpu: 200m
        memory: 256Mi
      requests:
        cpu: 10m
        memory: 64Mi
  grafana:
    persistentVolume:
      enabled: false
      storageClass: "-"
    resources:
      limits:
        cpu: 500m
        memory: 512Mi
      requests:
        cpu: 100m
        memory: 128Mi
```

Customizar os parâmetros

É possível customizar os valores dos parâmetros, conforme necessário.

- A seção `monitoring.prometheus` tem os parâmetros a seguir:
  - `prometheus.scrapeInterval` é a frequência para extrair destinos no Prometheus.
  - `prometheus.evaluationInterval` é a frequência para avaliar as regras no Prometheus.
  - `prometheus.retention` é a duração de tempo para reter os dados de monitoramento.
  - `prometheus.persistentVolume.enabled` é uma sinalização que você configura para usar um volume persistente para o Prometheus. A sinalização `false` significa que você não usa um volume persistente.
  - `prometheus.persistentVolume.storageClass` é a classe de armazenamento a ser usada pelo Prometheus. Consulte Parâmetro de classe de armazenamento.
  - `prometheus.resources.limits.cpu` é o limite de CPU que você configurou para o contêiner Prometheus. O valor padrão é de 500 millicpu.

    - `prometheus.resources.limits.memory` é o limite de memória que você configurou para o contêiner Prometheus. O valor padrão é de 512 milhões de bytes.
- A seção `monitoring.alertmanager` tem os parâmetros a seguir:
    - `alertmanager.persistentVolume.enabled` é uma sinalização que você configura para usar um volume persistente para Alertmanager. A sinalização `false` significa que você não usa um volume persistente.
    - `alertmanager.persistentVolume.storageClass` é a classe de armazenamento a ser usada pelo Alertmanager. Consulte Parâmetro de classe de armazenamento.
    - `alertmanager.resources.limits.cpu` é o limite de CPU que você configurou para o contêiner Alertmanager. O valor padrão é 200 millicpu.
    - `alertmanager.resources.limits.memory` é o limite de memória que você configurou para o contêiner Alertmanager. O valor padrão é 256 milhões de bytes.
- A seção `monitoring.grafana` tem os parâmetros a seguir:
    - `grafana.user` é o nome do usuário que você usa para acessar o Grafana.
    - `grafana.password` é a senha do usuário que está especificada no parâmetro `grafana.user`.
    - `grafana.persistentVolume.enabled` é uma sinalização que você configura para usar um volume persistente para Grafana. A sinalização `false` significa que você não usa um volume persistente.
    - `grafana.persistentVolume.storageClass` é a classe de armazenamento a ser usada pelo Grafana. Consulte Parâmetro de classe de armazenamento.
    - `grafana.resources.limits.cpu` é o limite de CPU que você configurou para o contêiner Grafana. O valor padrão é de 500 millicpu.
    - `grafana.resources.limits.memory` é o limite de memória que você configurou para o contêiner Grafana. O valor padrão é de 512 milhões de bytes.

Para todos os parâmetros disponíveis, consulte Parâmetros.

Parâmetro de classe de armazenamento

O valor de parâmetro `storageClass` é o nome da classe de armazenamento que o serviço de monitoramento usa.

- Insira – para não usar uma classe de armazenamento. Os dados são armazenados dentro do sistema de arquivos do contêiner e todos os dados são perdidos se o contêiner trava.
- Insira o nome de uma classe de armazenamento, como `glusterfs`, para usar o armazenamento compartilhado. Se você usar o armazenamento compartilhado, seus dados serão preservados se o contêiner travar. Para usar essa opção, deve-se configurar o provedor de armazenamento de rede. Consulte Armazenamento.

É possível especificar qualquer classe de armazenamento do Kubernetes válida. Veja Classes de armazenamento na documentação do Kubernetes. **Importante:** para instalações de alta disponibilidade do IBM Cloud Private, configure o armazenamento compartilhado. Consulte Alta Disponibilidade IBM® Cloud Private clusters. Se você não configurar o armazenamento compartilhado, o serviço de monitoramento poderá se tornar inacessível se o principal falhar.

NOTA: para ativar volumes persistentes para o serviço de monitoramento durante a instalação do IBM Cloud Private, deve-se usar um provedor de armazenamento, como o GlusterFS, que suporta fornecimento de armazenamento dinâmico. Se você escolher um provedor, como o NFS, que não suporta fornecimento de armazenamento dinâmico, deverá instalar seu serviço de monitoramento depois de instalar o IBM Cloud Private. Para obter mais informações, consulte Instalando o serviço de monitoramento no IBM Cloud Private.

Parâmetros

A tabela a seguir lista os parâmetros do Prometheus e seus valores padrão. É possível configurar esses parâmetros, conforme necessário.

Tabela 1. Parâmetro

Parâmetro	Descrição	Valor Padrão
`environment`	Ambiente de destino da implementação. As opções válidas são `openshift` e `non-openshift`.	`non-openshift`
`mode`	Modo de implementação. As opções válidas são `managed` e `standard`.	`standard`
`tls.enabled`	Ativar segurança para o Gráfico	`false`
`tls.issuer`	Nome do emissor do certificado	`icp-ca-issuer`
`tls.issuerKind`	Tipo de emissor de certificado. As opções válidas são `Issuer` e `ClusterIssuer`.	`ClusterIssuer`
`tls.ca.secretName`	Segredo do certificado de autoridade de certificação	`cluster-ca-cert`

Parâmetro	Descrição	Valor Padrão
tls.ca.certFieldName	Nome do certificado de autoridade de certificação que é usado no segredo	tls.crt
tls.server.existingSecretName	Segredo existente do certificado do servidor	""
tls.server.certFieldName	Nome do certificado do servidor que é usado no segredo	tls.crt
tls.server.keyFieldName	Nome da chave do servidor no segredo	tls.key
tls.exporter.existingSecretName	O segredo existente do certificado exportador	""
tls.exporter.certFieldName	Nome do certificado do exportador que é usado no segredo	tls.crt
tls.exporter.keyFieldName	Nome da chave do exportador que é usada no segredo	tls.key
tls.client.existingSecretName	Segredo existente do certificado de cliente	""
tls.client.certFieldName	Nome do certificado de cliente que é usado no segredo	tls.crt
tls.client.keyFieldName	Nome da chave do cliente que é usada no segredo	tls.key
imagePullPolicy	Política para puxar as imagens implementadas	IfNotPresent
imagePullSecrets	Segredo da imagem que é usado para extrair imagens de um repositório privado	""
clusterAddress	Endereço IP ou nome de DNS que é usado para acessar o cluster	127.0.0.1
clusterPort	Porta que é usada para acessar o cluster	8443
clusterDomain	Nome de domínio do cluster	cluster.local
clusterName	Nome do cluster de destino	mycluster
prometheus.image.repository	Nome da imagem do contêiner do servidor Prometheus	ibmcom/prometheus
prometheus.image.tag	Tag de imagem do contêiner do servidor Prometheus	v2.0.0
prometheus.port	Número da porta do serviço do servidor Prometheus	80
prometheus.scrapeInterval	Intervalo para raspar métricas	1m
prometheus.evaluationInterval	Intervalo de avaliação para regras de alerta	1m
prometheus.retention	Tempo de retenção de armazenamento do Prometheus	24h
prometheus.args	Argumentos para o contêiner prometheus	{}
prometheus.persistentVolume.enabled	Configure como true se desejar criar um volume para armazenar dados	false
prometheus.persistentVolume.useDynamicProvisioning	Configure como true se você desejar provisionar dinamicamente o volume persistente	true
prometheus.persistentVolume.size	Capacidade da solicitação de volume persistente	10Gi
prometheus.persistentVolume.storageClass	Classe de armazenamento para o volume persistente do Prometheus	""
prometheus.persistentVolume.existingClaimName	Especifique o nome se você desejar usar uma solicitação de volume persistente existente	""
prometheus.persistentVolume.selector.label	Se você desejar usar um determinado volume, especifique o nome do rótulo	""
prometheus.persistentVolume.selector.value	Se você desejar usar um determinado volume, especifique o valor do rótulo	""
prometheus.probe.enabled	Configure como true se desejar ativar a análise de funcionamento para Prometheus	true
prometheus.probe.readiness.args	Argumentos para análise de prontidão	{}
prometheus.probe.liveness.args	Argumentos para Análise de Atividade	{}
prometheus.resources.limits.cpu	Limites de CPU do Prometheus	500m
prometheus.resources.limits.memory	Limites de memória do Prometheus	512Mi
prometheus.resources.requests.cpu	Solicitações de CPU Prometheus	100m
prometheus.resources.requests.memory	Solicitações de memória do Prometheus	128Mi
prometheus.alertRuleFiles	Modelo de regras de alerta do Prometheus	alertRules
prometheus.configFiles	Modelo de configurações de Prometheus	prometheusConfig

Parâmetro	Descrição	Valor Padrão
prometheus.rbacRoleCreation	Configure como true se desejar criar a função de controle de acesso baseado na função (RBAC) e a ligação de função	true
prometheus.ingress.enabled	Configure como true se desejar criar ingresso do Prometheus	false
prometheus.ingress.annotations	Anotação para ingresso de Prometheus	{}
prometheus.service.type	Tipo de serviço do Prometheus	NodePort
prometheus.etcdTarget.enabled	Inclua o destino de extração etcd na configuração do Prometheus, se configurada como true	false
prometheus.etcdTarget.etcdAddress	lista de servidores etcd	["127.0.0.1"]
prometheus.etcdTarget.etcdPort	Porta do servidor etcd	4001
prometheus.etcdTarget.secret	Segredo que é usado para acessar o terminal de métricas etcd	etcd-secret
prometheus.etcdTarget.tlsConfig	Configuração de TLS para configuração de scrape etcd	{}
alertmanager.image.repository	Nome da imagem do contêiner do Alertmanager	ibmcom/alertmanager
alertmanager.image.tag	Tag de imagem do contêiner Alertmanager	v0.13.0
alertmanager.port	Porta de serviço do Alertmanager	80
alertmanager.persistentVolume.enabled	Cria um volume para armazenar dados, se configurado como true	false
alertmanager.persistentVolume.useDynamicProvisioning	Provisiona dinamicamente um volume persistente, se configurado como true	true
alertmanager.persistentVolume.size	Tamanho da solicitação de volume persistente	1Gi
alertmanager.persistentVolume.storageClass	Classe de armazenamento para o volume persistente do Alertmanager	""
alertmanager.persistentVolume.existingClaimName	Especifique o nome se você desejar usar uma solicitação de volume persistente existente	""
alertmanager.persistentVolume.selector.label	Se você desejar usar um determinado volume, especifique o nome do rótulo	""
alertmanager.persistentVolume.selector.value	Se você desejar usar um determinado volume, especifique o valor do rótulo	""
alertmanager.probe.enabled	Ativa a análise de funcionamento para Alertmanager, se configurado como true	true
alertmanager.probe.readiness.args	Argumentos para análise de prontidão	{}
alertmanager.probe.liveness.args	Argumentos para Análise de Atividade	{}
alertmanager.resources.limits.cpu	Limites de CPU do Alertmanager	200m
alertmanager.resources.limits.memory	Limites de Memória do Alertmanager	256Mi
alertmanager.resources.requests.cpu	Solicitações de CPU do Alertmanager	10m
alertmanager.resources.requests.memory	Pedidos de memória do Alertmanager	64Mi
alertmanager.configFiles	Nome do arquivo de configurações do Alertmanager	alermanagerConfig
alertmanager.ingress.enabled	Cria um ingresso do Alertmanager, se configurado como true	false
alertmanager.ingress.annotations	Anotação para ingresso do Alertmanager	{}
alertmanager.service.type	Tipo de serviço Alertmanager	NodePort
kubeStateMetrics.enabled	Instala o exportador de métricas do Kubernetes, se configurado como true	false
kubeStateMetrics.image.repository	kube-state-metrics nome da imagem do contêiner	ibmcom/kube-state-metrics
kubeStateMetrics.image.tag	tag de imagem de contêiner kube-state-metrics	v1.2.0
kubeStateMetrics.port	kube-state-metrics service port	80

Parâmetro	Descrição	Valor Padrão
kubeStateMetrics.probe.enabled	Ativa a análise de funcionamento para kubeStateMetrics, se configurado como true	true
kubeStateMetrics.probe.readiness.args	Argumentos para análise de prontidão	{}
kubeStateMetrics.probe.liveness.args	Argumentos para Análise de Atividade	{}
nodeExporter.enabled	Instala o exportador de nó, se configurado como true	false
nodeExporter.image.repository	node-nome da imagem do contêiner exportador	ibmcom/node-exporter
nodeExporter.image.tag	tag image do contêiner do nó exportadora	v0.15.2
nodeExporter.port	node-porta de serviço do exportador	9100
nodeExporter.probe.enabled	Ativa a análise de funcionamento para nodeExporter, se configurado como true	true
nodeExporter.probe.readiness.args	Argumentos para análise de prontidão	{}
nodeExporter.probe.liveness.args	Argumentos para Análise de Atividade	{}
grafana.image.repository	Nome da imagem do Docker Grafana	ibmcom/grafana
grafana.image.tag	Tag de Imagem do Docker Grafana	4.6.3
grafana.port	Porta exposta do contêiner Grafana	3000
grafana.user	Nome do usuário do Grafana	"admin"
grafana.password	Senha do usuário do Grafana	""
grafana.persistentVolume.enabled	Cria um volume para armazenar dados, se configurado como true	false
grafana.persistentVolume.useDynamicProvisioning	Provisiona dinamicamente um volume persistente, se configurado como true	true
grafana.persistentVolume.size	Tamanho da solicitação de volume persistente	1Gi
grafana.persistentVolume.storageClass	Classe de armazenamento para volume persistente	""
grafana.persistentVolume.existingClaimName	Especifique o nome se você desejar usar uma solicitação de volume persistente existente	""
grafana.persistentVolume.selector.label	Se você desejar usar um determinado volume, especifique o nome do rótulo	""
grafana.persistentVolume.selector.value	Se você desejar usar um determinado volume, especifique o valor do rótulo	""
grafana.probe.enabled	Ativa a análise de funcionamento para Grafana, se configurada como true	true
grafana.probe.readiness.args	Argumentos para análise de prontidão	{}
grafana.probe.liveness.args	Argumentos para Análise de Atividade	{}
grafana.resources.limits.cpu	Limites de CPU de Grafana	500m
grafana.resources.limits.memory	Limites de memória do Grafana	512Mi
grafana.resources.requests.cpu	Solicitações de CPU Grafana	100m
grafana.resources.requests.memory	Solicitações de memória de Grafana	128Mi
grafana.configFiles	Arquivo de configurações Grafana	grafanaConfig
grafana.ingress.enabled	Cria um ingresso de Grafana, se configurado como true	false
grafana.ingress.annotations	Anotação para ingresso de Grafana	{}
grafana.service.type	Tipo de serviço Grafana	NodePort
grafana.elasticsearchDash.enabled	Inclui um painel Elasticsearch, se configurado como true	false
collectdExporter.enabled	Instala o exportador collectd, se configurado como true	false
collectdExporter.image.repository	Nome da imagem do exportador Collectd	ibmcom/collectd-exporter
collectdExporter.image.tag	Tag de imagem do exportador Collectd	0.3.1
collectdExporter.service.serviceMetricsPort	Porta exposta do serviço de métricas	9103

Parâmetro	Descrição	Valor Padrão
collectdExporter.service.serviceCollectorPort	Porta Exposta Serviço do Coletor	25826
collectdExporter.probe.enabled	Ativa a análise de funcionamento para o exportador collectd, se configurado como true	true
collectdExporter.probe.readiness.args	Argumentos para análise de prontidão	{}
collectdExporter.probe.liveness.args	Argumentos para Análise de Atividade	{}
configmapReload.image.repository	Nome da imagem configmapReload Docker	ibmcom/configmap-reload
configmapReload.image.tag	Tag de imagem do Docker configmapReload	v0.1
router.image.repository	Nome da imagem do Docker do roteador	ibmcom/icp-router
router.image.tag	Tag de imagem do Docker do roteador	2.2.0
router.subjectAlt	DNC alternativo do assunto ou endereço IP para a chave SSL	127.0.0.1
elasticsearchExporter.enabled	Instala o exportador Elasticsearch, se configurado como true	false
elasticsearchExporter.image.repository	Nome da imagem do Docker exportador do Elasticsearch	ibmcom/elasticsearch_exporter
elasticsearchExporter.image.tag	Tag de imagem do Docker exportadora do Elasticsearch	1.0.2
elasticsearchExporter.esUri	URL Elasticsearch	https://elasticsearch:9200
elasticsearchExporter.tls.enabled	Ativa TLS para exportador para solicitar o terminal Elasticsearch	true
elasticsearchExporter.tls.ca.secretName	Segredo do certificado CS	cluster-ca-cert
elasticsearchExporter.tls.ca.certFieldName	Nome do campo para o certificado de autoridade de certificação no segredo	tls.crt
elasticsearchExporter.tls.client.existingSecretName	Segredo existente para o certificado de cliente	""
elasticsearchExporter.tls.client.certFieldName	Nome do campo para o certificado de cliente no segredo	tls.crt
elasticsearchExporter.tls.client.keyFieldName	Nome do campo para a chave do cliente no segredo	tls.key
elasticsearchExporter.port	Porta exportadora do Elasticsearch exposta	9108
elasticsearchExporter.probe.enabled	Ativa a análise de funcionamento para o exportador Elasticsearch, se configurada como true	true
elasticsearchExporter.probe.readiness.args	Argumentos para análise de prontidão	{}
elasticsearchExporter.probe.liveness.args	Argumentos para Análise de Atividade	{}
curl.image.repository	Nome da imagem do Docker curl	ibmcom/curl
curl.image.tag	tag de imagem do Docker curl	4.0.0
certGen.image.repository	Nome da imagem do Docker para gerar o certificado	ibmcom/icp-cert-gen
certGen.image.tag	Tag de imagem do Docker para gerar o certificado	1.0.0
init.image.repository	Nome da imagem do Docker init	ibmcom/icp-cert-gen
init.image.tag	tag de imagem do Docker init	1.0.0

Configurando o Armazenamento

Requisito de armazenamento no IBM® Cloud Private.

O IBM Cloud Private requer dois tipos de armazenamento:

- **Armazenamento da plataforma:** o armazenamento necessário para instalar o sistema operacional e seu cluster do IBM Cloud Private. O armazenamento de plataforma também inclui o armazenamento necessário para os componentes do IBM Cloud Private, que incluem etcd, image-manager, audit log e outros componentes. Para obter mais informações, consulte Armazenamento da plataforma.

- **Armazenamento do aplicativo:** armazenamento necessário para cargas de trabalho do aplicativo que são hospedadas no IBM Cloud Private. Como os pods são efêmeros, os aplicativos precisam de um volume persistente, caso eles desejam persistir seus dados na reinicialização. Há muitas opções no IBM Cloud Private para provisionar armazenamento persistente para as cargas de trabalho do seu aplicativo. Para obter mais informações, consulte Armazenamento do aplicativo.

As opções de armazenamento a seguir estão disponíveis para aplicativos que são implementados em seu cluster do IBM Cloud Private. É possível concluir a configuração durante ou após a instalação do IBM Cloud Private. No entanto, nem todas as opções de armazenamento estão disponíveis para configuração juntamente com a instalação do IBM Cloud Private.

Tabela 1. Provedores de armazenamento suportados por plataforma

Provedores de Armazenamento	Linux®	Linux® on Power® (ppc64le)	Linux® on IBM® Z and LinuxONE
GlusterFS	S	S	N
vSphere Cloud Provider	S	N	N
IBM Spectrum Scale	S	S	N
O Ceph bloqueia o armazenamento de bloco usando o Rook	S	N	N
Minio	S	S	N

Provedores de armazenamento disponíveis para configuração durante ou após a instalação do

IBM Cloud Private

GlusterFS

Consulte GlusterFS .

vSphere Cloud Provider

Consulte Configurando um vSphere Cloud Provider.

Minio

Consulte Minio .

Provedores de armazenamento disponíveis para configuração somente após o

IBM Cloud Private ser instalado

IBM Spectrum Scale

Consulte Usando o IBM Spectrum Scale para armazenamento em seu cluster do IBM® Cloud Privatecluster.

Interface de Armazenamento de Contêiner (CSI)

Consulte Interface de Armazenamento do Contêiner .

O Ceph bloqueia o armazenamento de bloco usando o Rook

Veja Armazenamento de bloco do Ceph usando Rook.

hostPath

Consulte hostPath .

Network File System

Consulte Network File System .

RBD externo do Ceph

Consulte External Ceph RBD .

Gráficos do Helm da comunidade

Consulte Opções de armazenamento disponíveis como gráficos do Helm da comunidade.

Depois que o provedor de armazenamento for configurado, será possível criar classes de armazenamento, volumes persistentes e solicitações de volume persistente.

Utilizando o armazenamento

Usando PersistentVolumes e PersistentVolumeClaims para consumo de armazenamento

Consulte PersistentVolume e PersistentVolumeClaims .

Fornecimento de armazenamento dinâmico usando classes de armazenamento

Consulte Fornecimento de armazenamento dinâmico.

Criptografando o tráfego de rede de dados do cluster com o IPsec

Criptografe todo o tráfego de rede do plano de dados entre nós em seu cluster do IBM® Cloud Private.

Pré-requisito

- Cada nó no cluster deve ter pelo menos duas interfaces de rede. Uma é uma interface de gerenciamento e a outra interface fornece redes seguras para os pods. A rede de gerenciamento é uma rede separada que é usada pelo Ansible para a instalação do IBM® Cloud Private.
- Seu arquivo host que é usado para configurar os nós do IBM Cloud Private deve conter endereços IP na rede de gerenciamento.

  Determinados pods de gerenciamento do Kubernetes que são executados na rede do host no cluster do IBM Cloud Private usam a rede de gerenciamento. A segunda interface que é usada para a comunicação entre pods é o que é protegida pelo IPsec.

Nota: os nomes de interface de rede não podem conter as sequências a seguir: `docker.*, cbr.*, dummy.*, virbr.*, lxcbr.*, veth.*, lo.*, cali.*, tunl.* ou flannel.*`.

- As redes Calico devem ser ativadas no modo IP sobre IP.
- Certifique-se de instalar o pacote `libreswan` em todos os nós no cluster que têm o sistema operacional Red Hat Enterprise Linux (RHEL). Nos nós que têm outros sistemas operacionais, instale o pacote `strongswan`. O pacote é necessário para ativar a criptografia do tráfego de rede de dados com IPsec. Assegure-se de que o serviço `strongswan` ou `libreswan` esteja configurado para iniciar após reinicialização do nó.
- Todos os nós em seu cluster devem executar o mesmo sistema operacional.
- Para atender aos requisitos do Federal Information Processing Standards (FIPS), siga as diretrizes para seu sistema operacional:
  - Para o RHEL, consulte Federal Standards and Regulations
  - Para Ubuntu, consulte Certificação
  - Para o SUSE Linux Enterprise Server (SLES), consulte Certificações e Verificações de Segurança

Ative a criptografia de tráfego de rede de dados do cluster

Por padrão, a criptografia é desativada no cluster do IBM Cloud Private.

Para ativar a criptografia, conclua as tarefas a seguir:

1. Verifique se os parâmetros a seguir existem no arquivo `config.yaml`. Para obter mais informações sobre esses parâmetros, consulte Configurações de rede:
   - `network_type: calico`
   - `calico_ipip_mode: Sempre`
   - `calico_ip_autodetection_method: interface=<data network interface>`

- `calico_tunnel_mtu`: 1390

  Configure `calico_tunnel_mtu` para um valor que possa acomodar os cabeçalhos do túnel Calico. O valor de MTU do túnel do Calico deve ter pelo menos 60 bytes a menos em relação ao tamanho de MTU da interface de rede de dados fornecida.

2. Inclua os dados de configuração a seguir no arquivo `config.yaml`:

```
# IPsec mesh configuration
# If user wants to configure IPsec mesh, the following parameters
# should be configured through config.yaml
ipsec_mesh:
 # To enable IPsec feature
 enable: true
 # List of subnets for which the IPsec should be enabled
 subnets: []
 # List of IPs to be excluded from IPsec subnet
 exclude_ips: []
 # List of ESP encryption/authentication algorithms to be used
 # cipher_suite: "aes128gcm16!"
```

   Visualize as descrições de parâmetros a seguir:

   - `enable` é usado para ativar ou desativar a criptografia do tráfego de plano de dados. O valor padrão é `false`.
   - `subnets` é o intervalo de endereços de rede das interfaces que são configuradas nos nós do cluster para comunicação entre pods. O endereço de sub-rede deve ser especificado no formato CIDR: `[a.b.c.d/n]`. Se necessário, liste várias sub-redes separadas por uma vírgula: `[a.b.c.d/n, l.m.n.o/p]`
   - `exclude_ips` são os endereços IP que são excluídos da sub-rede IPsec. O tráfego desses endereços IP não está criptografado. Esse parâmetro é opcional. Exemplo: [1.1.1.1/32,2.2.2.0/28,3.3.3.3.3....]. Se você fornecer o endereço IP, uma máscara de rede de `/32` será anexada automaticamente à configuração do IPsec.

   - `cipher_suite` é o algoritmo de criptografia e de autenticação do Encapsulating Security Payload (ESP) a ser usado. Se necessário, liste vários algoritmos separados por vírgulas. Esse parâmetro é opcional.

     - O conjunto de cifras padrão para o `stongswan ipsec` é `aes128gcm16!`.
     - O conjunto de cifras padrão para o `libreswan ipsec` é `aes_gcm_c128`.

     Assegure-se de que esse módulo esteja disponível e carregado em todos os hosts. Também é possível mudar o conjunto de cifras.

Em seguida, continue com a instalação do IBM Cloud Private.

Ativando a Mesh de IPsec após a instalação do IBM Cloud Private

1. Conclua as etapas em Ativar criptografia de tráfego de rede de dados do cluster.
2. Execute o comando a seguir:

```
sudo docker run --net=host -t -e LICENSE=accept -v \
"$(pwd)":/installer/cluster ibmcom/icp-inception:3.1.2 ipsec-mesh
```

Certificado IPsec e chave

A implementação do IPsec no IBM Cloud Private usa troca de chave da Internet (IKE) para autenticação mútua entre dois nós em seu cluster.

Para autenticação, cada nó possui um certificado digital assinado por uma autoridade confiável e uma chave privada para esse certificado digital.

Os certificados de nó e as chaves são gerados durante a instalação do IBM Cloud Private.

Usando seu próprio certificado e chave

É possível substituir os certificados e as chaves padrão depois de concluir a instalação do IBM Cloud Private. Assegure-se de que você tenha os arquivos a seguir criados:

- Arquivo da autoridade de certificação (CA). Nome do arquivo de exemplo: `example-ca.crt`
- Arquivo de certificado. Nome do arquivo de exemplo: `example-cert.crt`
- Arquivo de chave privado. Nome do arquivo de exemplo: `example-private.key`

Substituindo certificados e chaves no RHEL

Faça download do pacote OpenSSL para gerar o arquivo de pacote configurável `PKCS #12`. Para obter mais informações sobre os pacotes no OpenSSL, consulte OpenSSL.

1. Em qualquer nó do RHEL, gere o arquivo `PKCS #12`. Forneça o arquivo de certificado, o arquivo-chave privado e o arquivo de certificado de CA. Execute o comando a seguir:

   ```
   openssl pkcs12 -export -in <example-cert.crt> -inkey <example-private.key> -certfile <example-ca.crt> -out <ipsec-libreswan-example>.p12 -name <ipsec-libreswan-example> -password pass:
   ```

   Nota: se você não deseja configurar uma senha para o pacote configurável `PKCS #12` que está sendo criado, não especifique um valor no parâmetro `pass:`.

2. Copie o arquivo de pacote configurável `PKCS #12` que foi gerado para todos os outros nós do RHEL no cluster.

3. Em todos os nós do RHEL, importe o arquivo de pacote configurável `PKCS #12` para o banco de dados Network Security Services (nssdb). Execute o comando a seguir, mas a criação de uma senha é opcional. Deve-se usar aspas simples se você não criar uma senha:

   ```
   pk12util -i <ipsec-libreswan-example>.p12 -d sql:/etc/ipsec.d -W 'your.password'
   ```

4. Substitua o nome do certificado e o nome comum do certificado que estão no arquivo `/etc/ipsec.d/ipsec-libreswan.conf` em todos os nós do RHEL. Seu arquivo pode ser semelhante ao conteúdo a seguir:

   ```
   leftcert=ipsec-libreswan-example
   leftid="CN=ipsec-mesh-example"
   rightid="CN=ipsec-mesh-example"
   ```

   1. Para reiniciar o serviço IPsec nos nós do RHEL, execute o seguinte comando:

      ```
      service ipsec restart
      ```

Substituindo certificados e chaves em outros sistemas operacionais

1. Coloque o arquivo CA na pasta `/etc/ipsec.d/cacerts/`.

   ```
   cp de exemplo-ca.crt /etc/ipsec.d/cacerts/
   ```

2. Coloque o arquivo de certificado na pasta `/etc/ipsec.d/certs/`.

   ```
   Exemplo de cp-cert.crt /etc/ipsec.d/certs/
   ```

3. Coloque o arquivo de chave privado na pasta `/etc/ipsec.d/private/`.

   ```
   cp exemplo-private.key /etc/ipsec.d/private/
   ```

4. Substitua o nome do arquivo de certificado no arquivo `/etc/ipsec.conf`. Sua entrada pode ser semelhante ao texto a seguir:

   ```
   leftcert="example-cert.crt "
   ```

5. Substitua o nome do arquivo de chave privado no arquivo `/etc/ipsec.secrets` pelo seguinte nome do arquivo-chave:

   ```
   exemplo-private.key
   ```

6. Reinicie o serviço strongSwan:

   ```
   reinício do serviço strongswan
   ```

Especificando sua própria autoridade de certificação para os serviços do IBM Cloud Private

Todos os certificados requeridos pelos serviços que são executados no IBM Cloud Private são criados durante a instalação do IBM Cloud Private. Os certificados são criados e gerenciados pelo instalador do IBM Cloud Private ou pelo Gerenciador de certificados (cert-manager) do IBM Cloud Private. Esses certificados são assinados por uma autoridade de certificação (CA) também criada durante a instalação.

Antes de instalar o IBM Cloud Private, é possível fornecer sua própria autoridade de certificação (CA) para assinar certificados usados pelos serviços do IBM Cloud Private.

Use sua própria autoridade de certificação (CA) (BYOK)

É possível BYOK (Bring Your Own Key) para usar dentro de seu cluster do IBM Cloud Private. Sua chave de certificado BYOK deve ser exportada no formato PEM(OpenSSL).

1. Crie o diretório `cfc-certs/root-ca` dentro do diretório do cluster.

   ```
   mkdir -p <installation_dir>/cluster/cfc-certs/root-ca
   ```

2. Renomeie sua chave de CA existente para `ca.key` e copie-a para o diretório de instalação.

   ```
   cp <BYOK> <installation_dir>/cluster/cfc-certs/root-ca/ca.key
   ```

3. Renomeie seu certificado de autoridade de certificação existente para `ca.crt` e copie-o para o diretório de instalação.

   ```
   cp <BYOK_cert> <installation_dir>/cluster/cfc-certs/root-ca/ca.crt
   ```

4. Instale o seu cluster.

Acessando a CA Raiz do ICP

Depois de trazer sua própria CA Raiz do ICP, o Certificado de CA Raiz do ICP poderá ser acessado por meio do Segredo do Kubernetes `ibmcloud-cluster-ca-cert` no namespace `kube-public`.

Para recuperar e decodificar o certificado, execute o comando a seguir:

```
kubectl get secret -n kube-public ibmcloud-cluster-ca-cert -o jsonpath='{.data.ca\.crt}' | base64 --decode
```

Upgrade do IBM Cloud Private 3.1.2 e anterior

No IBM Cloud Private versão 3.1.2 e anterior, o certificado que é usado pelo ingresso de gerenciamento e pelo gerenciador de imagem não era gerenciado pelo serviço IBM Cloud Private Certificate Manager. O certificado foi criado inicialmente pelo instalador, e a maneira de trazer seu próprio certificado para os serviços de ingresso de gerenciamento e gerenciador de imagem era colocar seu próprio certificado no diretório correto.

Para IBM Cloud Private versão 3.2.0 e superior, a maneira de trazer seu próprio certificado é trazer sua própria Autoridade de Certificação.

Esta Autoridade de Certificação (sua própria autoridade de certificação raiz) assinará o certificado usado pelo ingresso de gerenciamento ou gerenciador de imagem.

Os cenários a seguir podem existir durante o upgrade:

1. Você não trouxe seu próprio certificado para o ingresso de gerenciamento e o gerenciador de imagem. Neste cenário, o ingresso de gerenciamento e o gerenciador de imagem selecionam o novo certificado que é gerado pelo cert-manager durante um upgrade. Se quiser trazer o seu próprio, consulte as etapas em Substituindo a autoridade de certificação raiz após fazer o upgrade do IBM Cloud Private.

2. Você trouxe seu próprio certificado para o ingresso de gerenciamento e gerenciador de imagem. Neste cenário, o ingresso de gerenciamento e o gerenciador de imagem não usam mais seu certificado após o upgrade. Deve-se seguir as etapas em Substituindo a autoridade de certificação raiz após o upgrade do IBM Cloud Private.

NOTA: se você substituir a autoridade de certificação raiz, após o upgrade, você deverá aceitar e confiar no novo certificado no navegador ao navegar para o console de gerenciamento do IBM Cloud Private.

Criptografando volumes usando dm-crypt

Criptografe volumes com `dm-crypt`.

O `dm-crypt` fornece criptografia transparente de dispositivos de bloco. É possível acessar os dados imediatamente após montar o dispositivo. Para obter mais informações sobre `dm-crypt`, consulte dm-crypt.

É possível usar `dm-crypt` para criptografar volumes e uma passphrase ou um arquivo-chave para decriptografar o volume. É possível especificar o arquivo-chave ao reiniciar o sistema.

Diretórios de armazenamento no IBM® Cloud Private

No IBM Cloud Private, é possível usar `dm-crypt` para criptografar dados em repouso que são armazenados nos locais a seguir:

- Kubernetes armazenamento de valor da chave- `/var/lib/etcd`
- Kubernetes de auditoria- `/var/lib/icp/audit`
- MongoDB - `/var/lib/icp/mongodb`
- Repositório Helm- `/var/lib/icp/helmrepo`
- Orientador de Vulnerabilidade (VA)- `/var/lib/icp/va`
- Volume persistente para o gerenciador de imagem - `/var/lib/registry`
- Certificados e chaves- `/etc/cfc`
- Licenças- `/opt/ibm/cfc/license`
- ID do Software Tags- `/opt/ibm/cfc/swidtag`
- Volume persistente para criação de log - `/var/lib/icp/logging/elasticsearch`
- Volume persistente para Prometheus - `<installation_directory>/dirforPrometheusServer`
- Volume persistente para AlertManager - `<installation_directory>/dirforAlertManager`
- Volume persistente para Grafana - `<installation_directory>/dirforGrafana`
- Configuração do IBM Cloud Private - `<installation_directory>/cluster`

É possível criptografar os diretórios a seguir para cobrir os locais de armazenamento na lista anterior:

- `/var/lib/etcd`
- `/var/lib/icp`
- `/var/lib/registry`
- `/etc/cfc`
- `/opt/ibm`
- `<installation_directory>`

  Nota: é possível usar `/opt/ibm` como o diretório de instalação.

Para obter mais informações sobre os requisitos de espaço em disco para esses diretórios, consulte Requisitos de espaço em disco.

Requisitos do FIPS

Consulte as diretrizes para atender aos requisitos do Federal Information Processing Standards (FIPS):

- Para o Red Hat Enterprise Linux (RHEL), consulte as seguintes diretrizes:
  - Padrões e Regulamentações Federais
  - Problema de FIPS do RedHat
- Para Ubuntu, consulte Certificação
- Para o SUSE Linux Enterprise Server (SLES), consulte Considerações sobre segurança

Criptografando um diretório

Deve-se concluir as etapas a seguir para criptografar o diretório `/var/lib/etcd`. É possível criptografar qualquer diretório de sua escolha.

Pré-requisitos

Assegure-se de que os pacotes a seguir estejam instalados em todos os nós de seu cluster do IBM Cloud Private:

- No RHEL, devem ser instalados os seguintes pacotes:
  - `cryptsetup`

- device-mapper
- util-linux

  Se os pacotes não estiverem instalados, execute os comandos a seguir como um usuário raiz para instalá-los:

  ```
  yum install cryptsetup-luks
  ```

- No Ubuntu, deve-se instalar os pacotes a seguir:
  - cryptsetup
  - libdevmapper1
  - util-linux

  Se os pacotes não estiverem instalados, execute estes comandos como um usuário raiz para instalá-los:

  ```
  apt-get install cryptsetup
  ```

Criptografando um volume

Conclua as etapas a seguir como um usuário raiz para criptografar um volume:

1. Configure o gerenciamento de volume lógico (LVM) para armazenar os dados criptografados:
   1. Execute o comando a seguir para criar um volume físico.

      ```
      pvcreate <full path and name of the physical volume>
      ```

      A saída pode ser semelhante ao conteúdo a seguir:

      ```
      pvcreate /dev/sda1
      Physical volume "/dev/sda1" successfully created.
      ```

   2. Crie um grupo de volumes. Execute o comando a seguir:

      ```
      vgcreate <name of the volume group> <full path to the physical volume>
      ```

      A saída pode ser semelhante ao conteúdo a seguir:

      ```
      vgcreate etcdvg /dev/sda1
      Volume group "etcdvg" successfully created.
      ```

   3. Execute o comando a seguir para criar um volume lógico.

      ```
      lvcreate -L <amount of space required> <name of the volume group> -n <name of the logical
      volume>
      ```

      A saída pode ser semelhante ao conteúdo a seguir:

      ```
      lvcreate -L4G etcdvg -n etcd
      Logical volume "etcd" created.
      ```

2. Crie um Contêiner LUKS de dm-crypt no volume executando o comando a seguir:
   ```
   cryptsetup -y luksFormat <full path to the logical volume>
   ```

   É possível especificar uma passphrase para decriptografia.

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   cryptsetup -y luksFormat /dev/etcdvg/etcd

   WARNING!
   ========
   This will overwrite data on /dev/etcdvg/etcd irrevocably.

   Are you sure? (Type uppercase yes): YES
   Enter passphrase:
   Verify passphrase:
   ```

3. Abra o contêiner LUKS e mapeie o volume lógico para seu caminho. Execute o comando a seguir:

   ```
   cryptsetup  luksOpen <full path to the logical volume> <name of the logical volume>
   ```

Sua saída pode ser semelhante ao texto a seguir:

```
cryptsetup  luksOpen /dev/etcdvg/etcd etcd

Enter passphrase for /dev/etcdvg/etcd:
```

4. Crie um sistema de arquivos no volume lógico. É possível usar qualquer sistema de arquivos. Execute o comando a seguir para criar um sistema de arquivos XFS:

   ```
   mkfs.xfs /dev/mapper/<name of the logical volume>
   ```

   Sua saída pode ser semelhante à saída a seguir:

   ```
   mkfs.xfs /dev/mapper/etcd

   meta-data=/dev/mapper/etcd       isize=512    agcount=4, agsize=262016 blks
            =                       sectsz=512   attr=2, projid32bit=1
            =                       crc=1        finobt=0, sparse=0
   data     =                       bsize=4096   blocks=1048064, imaxpct=25
            =                       sunit=0      swidth=0 blks
   naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
   log      =internal log           bsize=4096   blocks=2560, version=2
            =                       sectsz=512   sunit=0 blks, lazy-count=1
   realtime =none                   extsz=4096   blocks=0, rtextents=0
   ```

5. Crie um local de montagem para montar o sistema de arquivos. Execute o comando a seguir:

   ```
   mkdir <mount location>
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   mkdir /var/lib/etcd
   ```

6. Execute o comando a seguir para montar o sistema de arquivos.

   ```
   mount /dev/mapper/<name of the logical volume> <mount location>
   ```

   O exemplo de comando seria conforme a seguir:

   ```
   mount /dev/mapper/etcd /var/lib/etcd
   ```

   Para montar automaticamente entre reinicializações do sistema, inclua as linhas a seguir nos arquivos `/etc/crypttab` e `/etc/fstab`:

   - Inclua a linha a seguir no arquivo `/etc/crypttab`:

     ```
     <name of the volume group> <full path to the logical volume> {none|
     <absolute_path_to_keyfile>} luks
     ```

     Se você usou uma passphrase para decriptografia, inclua `none`. Se você usou um arquivo-chave para decriptografia, inclua o caminho completo para o arquivo-chave. Execute o exemplo de comando a seguir:

     ```
     etcd /dev/etcdvg/etcd none luks
     ```

   - Inclua a linha a seguir no arquivo `/etc/fstab`:

     ```
     /dev/mapper/<name of the logical volume> <full path to the volume group> xfs defaults 0 2
     ```

     Execute o exemplo de comando a seguir:

     ```
     /dev/mapper/etcd /var/lib/etcd xfs defaults 0 2
     ```

Em seguida, continue a instalar o IBM Cloud Private.

Criptografando volumes vSphere

Criptografe volumes do vSphere.

Em um ambiente do VMware vSphere, é possível implementar máquinas virtuais (VMs) com volumes de armazenamento criptografados para segurança incluída. É possível ativar a criptografia nos volumes de armazenamento que você cria nos

armazenamentos de dados da rede de área de armazenamento virtual (vSAN), do sistema de arquivos da máquina virtual (VMFS) e do Network File System (NFS).

Para criptografar volumes do vSphere, deve-se configurar um Key Management Server (KMS) no vSphere 6.5 ou 6.7. Para obter mais informações sobre como configurar o KMS, consulte os documentos do VMware a seguir:

- Para o vSphere 6.5, consulte Configurar o cluster do Key Management Server.
- Para o vSphere 6.7, consulte Configurar o cluster do Key Management Server.

Depois de configurar o KMS, crie uma política de armazenamento para ativar a criptografia. Para obter mais informações, consulte os documentos do VMware a seguir:

- Para o vSphere 6.5, consulte Criar uma Política de Armazenamento de Criptografia.
- Para o vSphere 6.7, consulte Criar uma Política de Armazenamento de Criptografia.

Também é possível ativar a criptografia para MVs existentes. Deve-se editar a política de armazenamento existente que é aplicada às MVs. Para obter mais informações, consulte os documentos do VMware a seguir:

- Para o vSphere 6.5, consulte Criptografar uma máquina virtual ou um disco virtual existente.
- Para o vSphere 6.7, consulte Criptografar uma máquina virtual ou um disco virtual existente.

  Nota: depois de atualizar a política de armazenamento, as MVs que usam essa política são encerradas para aplicar a política de armazenamento atualizada. As MVs são reiniciadas após a criptografia ser aplicada.

Para criar um volume criptografado, deve-se definir uma classe de armazenamento e especificar o nome da política de armazenamento que você criou para ativar a criptografia. Consulte Criando uma classe de armazenamento para o volume do vSphere.

Os pods em seu cluster do IBM® Cloud Private agora podem usar uma solicitação de volume persistente (PVC) para solicitar os volumes criptografados.

Integrando o VMware NSX-T 2.4 ao IBM Cloud Private

VMware NSX-T 2.4 *(NSX-T)*, fornece recursos de rede para um cluster IBM® Cloud Private.

Nota: se você configurou NSX-T em um cluster do IBM Cloud Private, a remoção de nós do trabalhador de seu cluster não removerá as portas e os fluxos da ponte Open vSwitch (OVS).

- Antes de incluir o nó do trabalhador no cluster novamente, deve-se limpar as portas e fluxos da ponte.
- É possível limpar as portas e os fluxos excluindo a ponte `br-int` e incluindo a ponte novamente ao integrar o NSX-T com os nós de cluster do IBM Cloud Private.

Importante: se você desinstalar o IBM Cloud Private, deverá remover as entradas relacionadas ao cluster, como regras de firewall, comutadores e roteadores da Camada 1 do seu gerenciador NSX-T.

Sistemas operacionais suportados

A integração do NSX-T com o Kubernetes é suportada nos seguintes sistemas operacionais:

- Ubuntu 16.04
- Red Hat Enterprise Linux (RHEL) 7.5 e 7.6 apenas

Integre o NSX-T com os nós do cluster do IBM Cloud Private

1. Instale o NSX-T no ambiente do VMware vSphere. Para obter mais informações sobre o NSX-T, consulte o Guia de instalação do NSX-T Data Center.
2. Configure recursos do NSX-T para p cluster do IBM Cloud Private. Para obter mais informações, consulte a Documentação do VMware.

   Nota: crie recursos, como zona de transporte de sobreposição, roteador lógico de camada 0, bloqueios de IP e conjuntos de IP. Certifique-se de manter um registro do nome ou da UUID dos recursos. Inclua o nome ou UUID dos recursos no arquivo config.yaml.

Nota: Ao configurar a seção **Blocos de IP para pods do Kubernetes**, use o bloco de IP que foi configurado como o `network_cidr` no arquivo `<installation_directory>/cluster/config.yaml`.

3. Instale o pacote de plug-in Container Network Interface (CNI) do NSX-T em cada nó em seu cluster. Para obter mais informações, consulte a Documentação do VMware.

4. Instale e configure o Open vSwitch (OVS) em cada nó em seu cluster. Para obter mais informações, consulte a Documentação do VMware.

   Nota: se o `ofport` designado não for `1`, certifique-se de incluir `ovs_uplink_port` no arquivo `config.yaml` quando Preparar o arquivo de configuração do IBM Cloud Private.

5. Configure a rede NSX-T em cada nó em seu cluster. Para obter mais informações sobre como configurar o NSX-T nos nós, consulte a Documentação do VMware. Ao marcar a porta do comutador lógico, certifique-se de usar os seguintes valores de parâmetro:

   - **`{'ncp/node_name': '<node_name>'}`**: Se você configurou `kubelet_nodename: hostname` no arquivo `<installation_directory>/cluster/config.yaml`, inclua o nome do host do nó como o valor de parâmetro `<node_name>`. É possível obter o nome do host do nó executando o comando a seguir:

     ```
     hostname -s
     ```

     Se você não configurou `kubelet_nodename: hostname` no arquivo `<installation_directory>/cluster/config.yaml`, inclua o endereço IP do nó como o valor de parâmetro `<node_name>`. Obtenha p endereço IP do nó do cluster do IBM Cloud Private do arquivo `<installation_directory>/cluster/hosts`.

   - **`{'ncp/cluster': '<cluster_name>'}`**: Use o `cluster_name` configurado no arquivo `<installation_directory>/cluster/config.yaml`. O valor padrão é `mycluster`.

Prepare o arquivo de configuração do IBM Cloud Private

Conclua as etapas a seguir para preparar o arquivo de configuração:

1. Se ele não existir, crie um diretório com o nome `images` na pasta `<installation_directory>/cluster/`.
2. Faça download e copie o arquivo .tar do contêiner do Docker NSX-T para a pasta `<installation_directory>/cluster/images`.

3. Inclua o seguinte parâmetro no arquivo `<installation_directory>/cluster/config.yaml`:

   ```
   network_type: nsx-t
   ```

   Nota: apenas um tipo de rede pode ser ativado para um cluster do IBM Cloud Private. Ao ativar o `network_type: nsx-t`, assegure-se de remover a configuração `network_type: calico` padrão do arquivo `config.yaml`.

Configuração NSX-T

Para configurar o NSX-T, inclua os parâmetros a seguir no arquivo `config.yaml`:

```
nsx_t:
  managers: <IP address>[:<port>],<IP address>[:port]
  manager_user: <user name for NSX-T manager>
  manager_password: <password for NSX-T manager user>
  manager_ca_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDYzCCAkugAwIBAgIEcK9gWjANBgsedkiG9w0BAQsFADBiMQswCQYDVQQGEwJV
    ........................................
    ........................................
    ........................................
    hzYlaog68RTAQpkV0bwedxq8lizEBADCgderTw99OUgt+xVybTFtHume8JOd+1qt
    G3/WlLwiH9upSujL76cEG/ERkPR5SpGZhg37aK/ovLGTtCuAnQndtM5jVMKoNDl1
    /UOKWe1wrT==
    -----END CERTIFICATE-----
  client_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDUDCCAjigAwIBAgIBCDANBgkqhkiG9w0BAQsFADA6TR0wGwYDVQQDDBQxMjcu
    ........................................
    ........................................
```

```
    ....................................
    X9Kr61vjKeOpboUlz/oGRo7AFlqsCSderTtQH28DWumzutfj
    -----END CERTIFICATE-----
  client_private_key:  |
    -----BEGIN PRIVATE KEY-----
    MIIEvgIBADANBgkqhkiG9w0BAUYTRASCBKgwggSkAgEAAoIBAQC/Jz4WnaTmbfB7
    ....................................
    ....................................
    ....................................
    n8jakjGLolYe5yv0KyM4RTD5
    -----END PRIVATE KEY-----
  subnet_prefix: 24
  external_subnet_prefix: 24
  ingress_mode: <hostnetwork or nat>
  ncp_package: <name of the NSX-T Docker container file that is placed in
`<installation_directory>/cluster/images` folder>
  ncp_image: registry.local/ob-5667597/nsx-ncp
  ncp_image_tag: latest
  ovs_uplink_port: <name of the interface that is configured as an uplink port >
  ovs_bridge: <OVS bridge name that is used to configure container interface >
  tier0_router: <name or UUID of the tier0 router >
  overlay_TZ: <name or UUID of the NSX overlay transport zone >
  container_ip_blocks: <name or UUID of the container IP blocks >
  external_ip_pools: <name or UUID of the external IP pools >
  no_snat_ip_blocks: <name or UUID of the no-SNAT namespaces IP blocks >
  node_type: <type of container node. Allowed values are `HOSTVM` or `BAREMETAL`>
  enable_snat: true
  enable_nsx_err_crd: false
  loadbalancer_enabled: false
  lb_default_ingressclass_nsx: true
  lb_l4_auto_scaling: true
  lb_external_ip_pools: <name or UUID of the external IP pools for load balancer>
  lb_pool_algorithm: ROUND_ROBIN
  lb_service_size: SMALL
  lb_l4_persistence: source_ip
  lb_l7_persistence: <persistence type for ingress traffic through Layer 7 load balancer. Allowed
values are `source_ip` or `cookie`>
  lb_default_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDUDCCAjigAwIBAgIBCDANBgkqhkiG9w0BAQsFADA6TR0wGwYDVQQDDBQxMjcu
    ....................................
    ....................................
    ....................................
    X9Kr61vjKeOpboUlz/oGRo7AFlqsCSderTtQH28DWumzutfj
    -----END CERTIFICATE-----
  lb_default_private_key: |
    -----BEGIN PRIVATE KEY-----
    MIIEvgIBADANBgkqhkiG9w0BAUYTRASCBKgwggSkAgEAAoIBAQC/Jz4WnaTmbfB7
    ....................................
    ....................................
    ....................................
    n8jakjGLolYe5yv0KyM4RTD5
    -----END PRIVATE KEY-----

  apparmor_enabled: true
  apparmor_profile: <name of the AppArmor profile to be used>
  firewall_top_section_marker: <name of the firewall section under which the firewall rule for your
cluster is created>
  firewall_bottom_section_marker: <name of the firewall section above which the firewall rule for
your cluster is created>
```

Nota: `managers`, `subnet_prefix`, `ncp_package`, `ncp_image`, `ncp_image_tag`, `overlay_TZ`, `container_ip_blocks`, `external_ip_pools`, `tier0_router` são parâmetros obrigatórios.

Importante: o `manager_user` e o `manager_password` ou o `client_cert` e o `client_private_key` são obrigatórios.

Consulte as seguintes diretrizes e valores para os parâmetros:

- `network_type`: deve ser configurado para `nsx-t`.
- `managers`: endereço IP ou nome do host do gerenciador NSX-T.
  Nota: é possível especificar múltiplos gerenciadores NSX-T incluindo os endereços IP ou nomes de host usando vírgula para separá-los.
- `manager_user`: nome do usuário do usuário que tem acesso ao gerenciador NSX-T.

- `manager_password`: senha do usuário especificado no parâmetro `manager_user`.
- `manager_ca_cert`: conteúdo do arquivo de certificado de autoridade de certificação do gerenciador NSX-T para verificar o certificado do servidor gerenciador NSX-T.
- `client_cert`: conteúdo do arquivo de certificado de cliente do gerenciador NSX-T para autenticação com o gerenciador NSX-T.
- `client_private_key`: conteúdo do arquivo de chave privado do cliente do gerenciador NSX-T para autenticação com o gerenciador NSX-T.
- `subnet_prefix`: comprimento do prefixo de sub-rede do bloco de endereços IP para pods.
- `external_subnet_prefix`: comprimento do prefixo de sub-rede da Conversão de Endereço de Rede *(NAT)* do bloco de endereço IP externo. Se o comprimento não for especificado, o valor em `subnet_prefix` será o comprimento padrão.
- `ingresss_mode`: fornece a opção para expor o controlador de ingresso. Especificar `nat` utiliza o conjunto NAT do NSX-T e especificar `hostnetwork` utiliza o endereço IP do nó.
  Nota: o controlador de ingresso de gerenciamento do IBM Cloud Private padrão usa o endereço IP do nó para roteamento e é possível configurar os controladores de ingresso customizados para usar o conjunto NAT do NSX-T para roteamento.
- `ncp_package`: arquivo .tar do contêiner Docker NSX-T que está na pasta `<installation_directory>/cluster/images`.
- `ncp_image`: nome da imagem de contêiner Docker NSX-T.
- `ncp_image_tag`: tag para a imagem de contêiner Docker NSX-T, como `latest`.
- `ovs_uplink_port`: nome da interface que é configurada como uma porta uplink.
  Nota: inclua esse parâmetro somente se o valor de `ofport` não for `1`.
- `ovs_bridge`: nome da ponte OVS que é usada para configurar a interface do contêiner.
- `tier0_router`: nome ou UUID do roteador lógico da Camada 0. Os roteadores da Camada 0 usam portas de downlink para se conectar aos roteadores da Camada 1 e portas de uplink para se conectar a redes externas.
- `overlay_TZ`: nome ou UUID da zona de transporte de sobreposição de NSX que cria comutadores lógicos para rede de contêineres. Cada hypervisor que hospeda as MVs do nó Kubernetes deve se associar a essa zona de transporte.
- `container_ip_blocks`: nome ou UUID dos blocos de IP que criam sub-redes.
  Nota: se um nome for escolhido, ele deverá ser exclusivo.
- `external_ip_pools`: nome ou UUID dos conjuntos de IP externos que alocam endereços IP para conversão de IPs de contêiner utilizando as regras de Conversão de Endereço de Rede de Origem (SNAT).
- `no_snat_ip_blocks`: nome ou UUID dos blocos de IP que criam sub-redes para projetos não SNAT. É possível especificar que os projetos não SNAT utilizem esses blocos de IP.
  Nota: se o valor de `no_snat_ip_blocks` estiver vazio, o valor de `container_ip_blocks` será o padrão.
- `node_type`: Tipo de nó do contêiner. Os valores permitidos são `HOSTVM` ou `BAREMETAL`.
- `enable_snat`: configurando para ativar ou desativar o SNAT.
  Nota: O valor padrão é `true`.
- `enable_nsx_err_crd`: este parâmetro é usado para ativar ou desativar o relatório de erro através da definição de recurso customizado (CRD) NSXError. O valor padrão é `false`.
- `loadbalancer_enabled`: configuração para ativar ou desativar um balanceador de carga.
  Nota: O valor padrão é `false`.
- `lb_default_ingressclass_nsx`: a configuração para o comportamento do controlador de ingresso. Os balanceadores de carga NSX manipularão o ingresso se o valor de parâmetro for `true`. Os controladores de ingresso de terceiros manipulam o ingresso quando o valor do parâmetro é `false`. Um exemplo de ingresso de terceiro é o NGINX.
  Nota: O valor padrão é `true`.
- `lb_l4_auto_scaling`: ative ou desative o ajuste de escala automático do balanceador de carga da Camada 4. O valor-padrão é `true`.
- `lb_external_ip_pools`: nome ou UUID dos conjuntos IP externos para o balanceador de carga.
- `lb_pool_algorithm`: algoritmo de balanceamento de carga para o objeto do conjunto do balanceador de carga.
  Nota: suas opções são `ROUND_ROBIN`, `LEAST_CONNECTION`, `IP_HASH` ou `WEIGHTED_ROUND_ROBIN`. O valor padrão é `ROUND_ROBIN`.
- `lb_service_size`: Tamanho do serviço do Balanceador de Carga.
  Nota: suas opções são `SMALL`, `MEDIUM` ou `LARGE`. O valor padrão é `SMALL`.
  Importante:
  - O balanceador de carga `SMALL` suporta 10 servidores virtuais
  - O balanceador de carga `MEDIUM` suporta 100 servidores virtuais
  - O balanceador de carga `LARGE` suporta 1.000 servidores virtuais
- `lb_l4_persistence`: tipo de persistência para tráfego de ingresso por meio do balanceador de carga de Camada 4. O valor permitido é `source_ip`.
- `lb_l7_persistence`: tipo de persistência para tráfego de ingresso por meio do balanceador de carga de Camada 7. Os valores permitidos são `source_ip` ou `cookie`.
- `lb_default_cert`: insira o conteúdo do arquivo de certificado padrão para o balanceamento de carga HTTPS.
- `lb_default_private_key`: insira o conteúdo do arquivo de chave privado para o certificado padrão para o balanceamento de carga HTTPS.

- `apparmor_enabled`: especifica o status do serviço AppArmor no sistema. O valor-padrão é `true`.
  Importante: este parâmetro é aplicável apenas ao Ubuntu.
  Nota: Para o RHEL, o parâmetro deve ser configurado como `false`.
- `apparmor_profile`: Nome do perfil do AppArmor. O nome do perfil do AppArmor padrão é `node-agent-apparmor`. Se você estiver usando outro perfil, especifique o nome do perfil customizado como o valor de parâmetro.
- `firewall_top_section_marker`: nome da seção de firewall sob a qual a regra de firewall para seu cluster é criada.
- `firewall_bottom_section_marker`: nome da seção de firewall acima da qual a regra de firewall para seu cluster é criada.

Em seguida, continue com a instalação do IBM Cloud Private.

Ativando o Consultor de Vulnerabilidade

Ative o Orientador de vulnerabilidade no cluster durante a instalação do IBM® Cloud Private.

Nota: é possível ativar o Orientador de vulnerabilidade após a instalação do IBM Cloud Private. Para obter mais informações, consulte Vulnerability Advisor .

Para obter mais informações sobre o Vulnerability Advisor, consulte a seção *Sobre o Vulnerability Advisor* no IBM Cloud Docs.

O recurso do Consultor de vulnerabilidade é suportado apenas nas edições do Cloud Native e Enterprise do IBM Cloud Private.

Se você tiver o Consultor de Vulnerabilidade ativado, seu sistema poderá requerer CPU, memória e espaço em disco adicionais. Consulte Requisitos de hardware e recomendações.

1. Configure nós do VA dedicados. Deve-se ter 1, 3 ou 5 nós do VA dedicados. Para configurar os nós dedicados, durante a instalação, forneça os IPs do nó na seção `[va]` dos arquivos host. Deve-se remover o `#` do cabeçalho `[va]` em seu arquivo `hosts`.

2. Ative o orientador de vulnerabilidade. Na lista de serviços de gerenciamento, configure vulnerability-advisor como `enabled`, conforme mostrado no exemplo a seguir:

   ```
   management_services:
   istio: disabled
   vulnerability-advisor: enabled
   storage-glusterfs: disabled
   storage-minio: disabled
   ```

   Importante: o Vulnerability Advisor depende dos serviços de criação de log para operar. O parâmetro `logging` deve ser configurado como `enabled` na lista de parâmetros `management_services`. **Nota:** se você desejar implementar mais de um nó do VA, o diretório do VA Minio `/var/lib/icp/va/minio` em cada nó do VA deverá estar no armazenamento compartilhado. O caminho do diretório pode ser mudado ao usar a opção `va_minio_storage_dir` no `config.yaml`.

Configurando um balanceador de carga externo

Saiba como configurar um balanceador de carga externo para seus nós principais ou do proxy em um ambiente de alta disponibilidade.

Os nós principais e do proxy em um ambiente de alta disponibilidade do IBM® Cloud Private usam ucarp e etcd como um balanceador de carga. Com essa configuração, o endereço IP virtual (VIP) é ligado a um nó principal ou um nó do proxy.

Você também pode desejar usar um balanceador de carga externo como uma alternativa ou substituição para o VIP.

Para ativar um modo de balanceador de carga externo em um ambiente de alta disponibilidade do IBM Cloud Private, deve-se preparar um nó do balanceador de carga e instalar o HAProxy. Em seguida, configure o balanceador de carga configurando os parâmetros `cluster_lb_address` e `proxy_lb_address` no arquivo `config.yaml`.

O balanceador de carga externo do cluster é usado para balanceamento de carga dos serviços de gerenciamento do IBM Cloud Private. O balanceador de carga externo do proxy é usada para balanceamento de carga dos serviços de carga de trabalho do IBM Cloud Private.

1. Para configurar um balanceador de carga externo do cluster, assegure-se de que as portas a seguir sejam incluídas no nó do balanceador de carga e estejam abertas: 8001, 8443, 8500, 8600, 9443.

2. Para configurar um balanceador de carga externo do proxy, assegure-se de que as portas a seguir sejam incluídas no nó do balanceador de carga e estejam abertas: 80 e 443.

3. Configure o nó do balanceador de carga. Esse nó do balanceador de carga não deve ser compartilhado com outros nós do cluster, como nós principais, do trabalhador ou do proxy. Um nó dedicado é necessário para evitar conflitos de porta.

    1. Instale o HAproxy no nó do balanceador de carga.

       Para Ubuntu:

       ```
       Apt-get install haproxy
       ```

       Red Hat Enterprise Linux (RHEL):

       ```
       Haproxy yum install
       ```

    2. Configure haproxy. Configure o HAproxy no arquivo `/etc/haproxy/haproxy.cfg` no nó do balanceador de carga. Por exemplo:

       ```
         # Example configuration for a possible web application.  See the
       # full configuration options online.
         #
       #   http://haproxy.1wt.eu/download/1.4/doc/configuration.txt
       #
       # Global settings
       global
             # To view messages in the /var/log/haproxy.log you need to:
             #
             # 1) Configure syslog to accept network log events.  This is done
             #    by adding the '-r' option to the SYSLOGD_OPTIONS in
             #    /etc/sysconfig/syslog.
         #
             # 2) Configure local2 events to go to the /var/log/haproxy.log
             #   file. A line similar to the following can be added to
             #   /etc/sysconfig/syslog.
         #
             #    local2.*                       /var/log/haproxy.log
             #
             log         127.0.0.1 local2

         chroot      /var/lib/haproxy
             pidfile     /var/run/haproxy.pid
             maxconn     4000
             user        haproxy
             group       haproxy
             daemon

         # 3) Turn on stats unix socket
             stats socket /var/lib/haproxy/stats
       # Common defaults that all the 'listen' and 'backend' sections
       # use, if not designated in their block.
         defaults
         mode                    http
         log                     global
         option                  httplog
         option                  dontlognull
         option http-server-close
         option                  redispatch
         retries                 3
         timeout http-request    10s
         timeout queue           1m
         timeout connect         10s
         timeout client          2m
         timeout server          2m
         timeout http-keep-alive 10s
         timeout check           10s
         maxconn                 3000

         frontend k8s-api
         bind *:8001
         mode tcp
         option tcplog
         use_backend k8s-api
       ```

```
backend k8s-api
mode tcp
balance roundrobin
server server1 <master_node_1_IP_address>:8001 check
server server2 <master_node_2_IP_address>:8001 check
server server3 <master_node_3_IP_address>:8001 check

frontend dashboard
bind *:8443
mode tcp
option tcplog
use_backend dashboard

backend dashboard
mode tcp
balance roundrobin
server server1 <master_node_1_IP_address>:8443 check
server server2 <master_node_2_IP_address>:8443 check
server server3 <master_node_3_IP_address>:8443 check

frontend auth
bind *:9443
mode tcp
option tcplog
use_backend auth

backend auth
mode tcp
balance roundrobin
server server1 <master_node_1_IP_address>:9443 check
server server2 <master_node_2_IP_address>:9443 check
server server3 <master_node_3_IP_address>:9443 check

frontend registry
bind *:8500
mode tcp
option tcplog
use_backend registry

frontend image-manager
bind *:8600
mode tcp
option tcplog
use_backend image-manager

backend image-manager
mode tcp
balance roundrobin
server server1 <master_node_1_IP_address>:8600 check
server server2 <master_node_2_IP_address>:8600 check
server server3 <master_node_3_IP_address>:8600 check

backend registry
mode tcp
balance roundrobin
server server1 <master_node_1_IP_address>:8500 check
server server2 <master_node_2_IP_address>:8500 check
server server3 <master_node_3_IP_address>:8500 check

frontend proxy-http
bind *:80
mode tcp
option tcplog
use_backend proxy-http

backend proxy-http
mode tcp
balance roundrobin
server server1 <proxy_node_1_IP_address>:80 check
server server2 <proxy_node_2_IP_address>:80 check
server server3 <proxy_node_3_IP_address>:80 check

frontend proxy-https
bind *:443
mode tcp
```

```
  option tcplog
  use_backend proxy-https

  backend proxy-https
  mode tcp
  balance roundrobin
  server server1 <proxy_node_1_IP_address>:443 check
  server server2 <proxy_node_2_IP_address>:443 check
  server server3 <proxy_node_3_IP_address>:443 check

  # OPTIONAL: Enable the following Kubernetes NodePorts for applications that require
them:
  frontend proxy-nodeport
  bind *:30000-32767
  mode tcp
  option tcplog
  use_backend proxy-nodeport

  backend proxy-nodeport
  mode tcp
  balance roundrobin
  server server1 <proxy_node_1_IP_address>
  server server2 <proxy_node_2_IP_address>
  server server3 <proxy_node_3_IP_address>_
```

- Para configurar um balanceador de carga de cluster, substitua `<master_node_1_IP_address>`, `<master_node_2_IP_address>` e `<master_node_3_IP_address>` pelos endereços IP para os nós principais de HA.
- Para configurar um balanceador de carga de proxy, substitua `<proxy_node_1_IP_address>`, `<proxy_node_2_IP_address>` e `<proxy_node_3_IP_address>` pelos endereços IP para os nós do proxy de HA.

3. Inicie o `serviço haproxy` executando o comando a seguir no nó do balanceador de carga:

```
systemctl start haproxy
```

4. Atualize o `config.yaml` arquivo. Substitua o parâmetro `cluster_lb_address` ou `proxy_lb_address` pelo endereço IP para seu nó do balanceador de carga externo. Por exemplo:

```
 ## External loadbalancer IP or domain
## Or floating IP in OpenStack environment
cluster_lb_address: none

 ## External loadbalancer IP or domain
## Or floating IP in OpenStack environment
proxy_lb_address: none
```

Gerando logs de auditoria do Kubernetes

Logs de auditoria do Kubernetes em IBM® Cloud Private.

Os logs de auditoria do Kubernetes são usados para rastrear e armazenar dados que estão relacionados ao seu uso do IBM Cloud Private. As políticas de auditoria são usadas para definir as regras para o tipo de dados a serem salvos nos logs de auditoria. IBM Cloud Private usa o padrão Kubernetes política de auditoria. Para obter mais informações sobre a política de auditoria do Kubernetes padrão, consulte https://kubernetes.io/docs/tasks/debug-application-cluster/audit/.

Nota: para obter informações sobre tamanhos de dados de auditoria, consulte Estatísticas de dados de criação de log de auditoria.

Por padrão, os logs de auditoria do Kubernetes não estão disponíveis no IBM Cloud Private. Para gerar esses logs, durante a instalação, configure o parâmetro `auditlog_enabled` como `true` no arquivo `/<installation_directory>/cluster/config.yaml`. Para obter mais informações, consulte Configurações do Kubernetes .

Os arquivos de log são salvos na pasta `/var/log/k8saudit/`.

Ativando a auditoria do Kubernetes após a instalação

Conclua as etapas a seguir para ativar a auditoria do Kubernetes.

1. Efetue SSH para o nó principal como usuário `raiz`.

2. Copie o arquivo `master.json` no local `tmp`.

   ```
   cp /etc/cfc/pods/master.json /tmp/
   ```

3. Edite o arquivo `master.json` copiado usando qualquer editor. Por exemplo:

   ```
   vim /tmp/master.json
   ```

4. Inclua o caminho do `arquivo de política de auditoria` e o caminho do `arquivo de log de auditoria`. Os caminhos de arquivo devem estar sob a seção de configuração `apiserver` e após o último elemento na lista de comandos.

   ```
   "--audit-policy-file=/etc/cfc/conf/audit-policy.yaml",
   "--audit-log-path=/var/log/k8saudit/audit.log",
   "--audit-log-maxage=3",
   "--audit-log-maxbackup=10",
   "--audit-log-maxsize=10"
   ```

   Nota: inclua uma vírgula (`,`) após o último elemento de parâmetros de comando se ele for incluído no meio.

5. Substitua o `master.json` original por um atualizado.

   ```
   cp /tmp/master.json /etc/cfc/pods/master.json
   ```

6. O pod principal seleciona as mudanças e o `kube-apiserver` é reinicializado com a auditoria ativada.

Configurando um caminho de log systemd-journald

Configure o `systemd-journald` para armazenar dados do log.

Logs de auditoria gerados por vários serviços de plataforma do IBM Cloud Private são enviados para o `systemd-journald` no nó. Em seguida, um daemonset fluentd lê os dados de auditoria do log do diário e os envia para o Elasticsearch. Por padrão, o diário armazena dados do log no diretório `/run/log/journal`. Se o `systemd-journald` estiver configurado para armazenar dados do log em algum outro local, será necessário configurar o parâmetro *journal_path* para esse local. Configure todos os nós no cluster para usar o mesmo local para armazenar dados do log de diário.

Especificando cifras TLS para etcd e Kubernetes

Os conjuntos de cifras padrão que são captados por etcd, kube-apiserver e kubelet possuem cifras fracas `ECDHE-RSA-DES-CBC3-SHA`, que podem ter problemas de vulnerabilidade de segurança. Para evitar problemas, é possível configurar o etcd, o kube-apiserver e o kubelet para especificar os conjuntos de cifras que possuem forte proteção para o cluster do IBM® Cloud Private.

Nota: a ativação de HTTP2 pode complicar a ordenação de conjuntos de cifras. É necessário selecionar suas próprias cifras e especificar a ordem.

```
* [etcd](#etcd)
* [ kube-apiserver ](#kube-apiserver)
* [ kubelet ](#kubelet)
```

etcd

É possível especificar as cifras TLS suportadas a serem usadas na comunicação entre os servidores principais e etcd.

1. Em `config.yaml`, inclua a opção a seguir:

   ```
    etcd_extra_args: [ "-- cipher-suites=TLS_ECDHEDHE_RSA_WITH_AES_128_GCM_SHA256,
   TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" ]
   ```

   Para obter mais informações, consulte a Documentação da comunidade etcd.

2. Quando o cluster IBM® Cloud Private estiver em execução, será possível verificar se os conjuntos de criptografia foram aplicados. Por exemplo:

   ```
   # openssl s_client -connect 9.111.254.123:4001
   CONNECTED(00000003)
   depth=0 CN = demo.icp
   verify error:num=20:unable to get local issuer certificate
   verify return:1
   ```

```
 depth=0 CN = demo.icp
 verify error:num=21:unable to verify the first certificate
 verify return:1
 140175725818304:error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad
certificate:../ssl/record/rec_layer_s3.c:1399:SSL alert number 42
 ---
 Certificate chain
 0 s:/CN=demo.icp
 i:/CN=demo.icp
 ---
 Server certificate
 -----BEGIN CERTIFICATE-----
 MIIDbDCCAlSgAwIBAgIQFNCXgjR0zeZdoWqxKe7jHTANBgkqhkiG9w0BAQsFADAT
 MREwDwYDVQQDDAhkZW1vLmljcDAgFw0xODA5MjcxMTQ2NDlaGA8yMTE4MDkwMzEx
 NDY0OVowEzERMA8GA1UEAwwIZGVtby5pY3AwggEiMA0GCSqGSIb3DQEBAQUAA4IB
 DwAwggEKAoIBAQDmr1sxcPBHCOfIzcMZpQQGP2pHQ1R3s7mUgBNdjkPkyLhavkhE
 Zh6Wxg++7DMdf7hK/5aNjYUESK1JOasEGpYH3jlZ5fN9Ty3zj1n3EnBuN6y5RUKC
 UnWlWbBATaJ5FKxNzVLPdTLdk73+iQw3QERT5jIzIMz+00fuJCixGdSPHPu5BT85
 8+zcr48foENWPGn0Bjj4K6toKZCjof0JMSYHxHoxXFeTsj1uxlMkpZxzxYwXaevF
 4FrauwnpYQd50k7B7V+TvRJcGSmuB4oM5M+lVWG8fr1881c+zwy8ni3lzZZuuZjS
 6g2CCVx94Z2LgUYrZgjPd8NgYjTPN7rluqRBAgMBAAGjgbkwgbYwCQYDVR0TBAIw
 ADAdBgNVHQ4EFgQUAfQBsQCV103gEQMEhEc8utamfFowQwYDVR0jBDwwOoAU2oeq
 ruGU/ClldMAtX2FGI5rhomehF6QVMBMxETAPBgNVBAMMCGRlbW8uaWNwggkA0jui
 s4EcWZEwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMAsGA1UdDwQEAwIF
 oDAZBgNVHREEEjAQgghkZW1vLmljcIcECW/+ezANBgkqhkiG9w0BAQsFAAOCAQEA
 ltu1BfqxaaeYAQ/hwoJgWzRzAgfnfpynEYDfqE+DUne5uBRySMj3E2CJOZ3wPLOY
 KQQ/JKUSiNCtHvYkbGSys6YLjHb0VOTF0uCoo5nC4J4jAKQmOGZsoXS1XlqnC/HH
 o1nR4B493HKcJN/QkMWr7zy+2kSno2RSftNL6q/6zuMjN4DPm6+8fUJ/Vz89T/AL
 heQjVXZr3uZseFv6IkXVQWH7bhMYCcUoyk582N6h5UybbMCZwILJqdjLmzzH/99m
 JHRaoc0KFM5QR1gzfgnnIBes5AxxQfenkai7HA7rmJObDlbJq4TdNiQXXjpV0HVm
 Ay3Q5PFHNwepgtMNkB8FKg==
 -----END CERTIFICATE-----
 subject=/CN=demo.icp
 issuer=/CN=demo.icp
 ---
 Acceptable client certificate CA names
 /CN=demo.icp
 Client Certificate Types: RSA sign, ECDSA sign
 Requested Signature Algorithms:
RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA1:ECDSA+SHA1
 Shared Requested Signature Algorithms:
RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA1:ECDSA+SHA1
 Peer signing digest: SHA384
 Server Temp Key: X25519, 253 bits
 ---
 SSL handshake has read 1325 bytes and written 281 bytes
 Verification error: unable to verify the first certificate
 ---
 New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
 Server public key is 2048 bit
 Secure Renegotiation IS supported
 Compression: NONE
 Expansion: NONE
 No ALPN negotiated
 SSL-Session:
     Protocol  : TLSv1.2
     Cipher    : ECDHE-RSA-AES128-GCM-SHA256
     Session-ID:
     Session-ID-ctx:
     Master-Key:
0465F6532FBF62DBD971C9307EB86C9FAFCCD665A2E11C7B674AC78D7515B2DD6F7EE6F8C2D637AA7AD770C434A74C9
4
     PSK identity: None
     PSK identity hint: None
     SRP username: None
     Start Time: 1539238527
     Timeout   : 7200 (sec)
     Verify return code: 21 (unable to verify the first certificate)
     Extended master secret: no
 ---
```

Nota: deve-se substituir o IP `9.111.254.123` pelo seu próprio IP do host (etcd).

kube-apiserver

É possível especificar as cifras TLS suportadas para uso na comunicação entre o kube-apiserver e os aplicativos.

1. Em `config.yaml`, inclua a opção a seguir:

   ```
    kube_apiserver_extra_args: [ "--tls-cipher-suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
   TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" ]
   ```

   Possíveis conjuntos de cifras são:

   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
   - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
   - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
   - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305
   - TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
   - TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
   - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
   - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
   - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
   - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
   - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
   - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305
   - TLS_ECDHE_HEDHE_RSA_WITH_RC4_128_SHA
   - TLS_RSA_WITH_3DES_EDE_CBC_SHA
   - , TLS_RSA_WITH_AES_128_CBC_SHA
   - , TLS_RSA_WITH_AES_128_CBC_SHA256
   - TLS_RSA_WITH_AES_128_GCM_SHA256
   - TLS_RSA_WITH_AES_256_CBC_SHA
   - TLS_RSA_WITH_AES_256_GCM_SHA384
   - , TLS_RSA_WITH_RC4_128_SHA

     Para obter mais informações, consulte a Documentação do Kubernetes.

2. Quando o cluster IBM® Cloud Private estiver em execução, será possível verificar se os conjuntos de criptografia foram aplicados.

   ```
   # openssl s_client -connect 9.111.254.123:8001
   CONNECTED(00000003)
   depth=0 CN = kubernetes-master
   verify error:num=20:unable to get local issuer certificate
   verify return:1
   depth=0 CN = kubernetes-master
   verify error:num=21:unable to verify the first certificate
   verify return:1
   ---
   Certificate chain
   0 s:/CN=kubernetes-master
   i:/C=US/ST=New York/L=Armonk/O=IBM Cloud Private/CN=www.ibm.com
   ---
   Server certificate
   -----BEGIN CERTIFICATE-----
   MIIFjTCCA3WgAwIBAgIQZFPqfeJs0BCqmwejqkO96zANBgkqhkiG9w0BAQsFADBj
   MQswCQYDVQQGEwJVUzERMA8GA1UECAwITmV3IFlvcmsxDzANBgNVBAcMBkFybW9u
   azEaMBgGA1UECgwRSUJNIENsb3VkIFByaXZhdGUxFDASBgNVBAMMC3d3dy5pYm0u
   Y29tMCAXDTE4MDkyNzExNDY1NloYDzIxMTgwOTAzMTE0NjU2WjAcMRowGAYDVQQD
   DBFrdWJlcm5ldGVzLW1hc3RlcjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
   ggEBANhVrVCp3zb+3xGm+FTqXoTg7zPTZTsMkUOE0YV9iZF+AZhNYGUAgmkTqroL
   gsob/S60t+oBZfLrTrRq1/q3GPt6/2fS72dhfrcj/0ALNE9bVQJBf3c/A57qx+Io
   X/BACgnZOEqi2mn6+x1OUibdTyysFyrKoxAzDiO+kP1A4YcPGmPazGkHpETоJy1H
   tGoFzFw5u7+Q7FTrcmfid0hkyNX1AsGDyHMIr5SzG3zb60Hzp+flqKs4vj0qbrv3
   +aJOfd2RTGWRiXHghXRzpkWPI3dTqjvDrD0eGnAZCZZUZnuC5KFkRCtql4LVM6pP
   azt7ePb7exxlOBik0oLewYsjquUCAwEAAaOCAYAwggF8MAkGA1UdEwQCMAAwHQYD
   VR0OBBYEFOEOpZ4QWiaSf2ZWfNBf9x5rJpXMMIGVBgNVHSMEgY0wgYqAFN1iQB0O
   5t1rS5c0zJPEbOQTrgntoWekZTBjMQswCQYDVQQGEwJVUzERMA8GA1UECAwITmV3
   IFlvcmsxDzANBgNVBAcMBkFybW9uazEaMBgGA1UECgwRSUJNIENsb3VkIFByaXZh
   dGUxFDASBgNVBAMMC3d3dy5pYm0uY29tggkAwvWGvsSQvmowHQYDVR0lBBYwFAYI
   KwYBBQUHAwEGCCsGAQUFBwMCMAsGA1UdDwQEAwIFoDCBiwYDVR0RBIGDMIGAhwR/
   AAABhwR/AAABhwQKAAABhwQJb/57ggprdWJlcm5ldGVzghJrdWJlcm5ldGVzLmRl
   ```

```
 ZmF1bHSCFmt1YmVybmV0ZXMuZGVmYXVsdC5zdmOCJGt1YmVybmV0ZXMuZGVmYXVs
 dC5zdmMuY2x1c3Rlci5sb2NhbIIIZGVtby5pY3AwDQYJKoZIhvcNAQELBQADggIB
 AIN0Pv3Fj0d5ECLLLGcCd0l8KTcI7wtPX9JIm5ekhxXQp/rX0EOaIo4KLoZWujjA
 3jUq+qnNR7WfW5nPump8mfFkfwDgg00NXvejQM6C+ozugxjYMXOsg8iB2pLzdLoV
 NJgzUjDbRYbriypzIQqhxcmfmM8sHyg7RDdCEZHBbrUvOuWwyViaBsfCuEWRd0ML
 oWjIBAII4N+QosKraZCWkSVhuPIxIlWYQZRhYhahnbDfStnNnt+Du3NgBvcbvzs6
 v6AUmfyRzXPtaOUEUq+r6pdCpjyOpEiKUe2Qbudym+TP0XKR0jEYwRFeekMJEYBu
 KdGwkwHlxhSBgluCmNKqCHJkwdn/X+Txkhkeyhga2gFjtn4xglVe90WbFk2dzXOt
 tOmDRClPr1hgjAsiAkv0aRyZqvmtarXfPAIQXD74S8a3aTBcxxXSLW2SHKkMqFaF
 GbmI7LX761GCculY2mP32hCKjKDLXzWiOqxLUc2+2pie4Sj0gui++nn1oMUTdwt4
 eet2iMqQ89hEIhL6pbLoXnJP9asr/LU8lv/AT9ci++HNh6zr8AiGM377dFZ9NgwJ
 s4TdM0MyR4Qv45hGkQ1UNbJck//CD4FFJELoL2vYFAL2DZXA6u8g/lJVlWjro6Qz
 TtSr1bZrPgTK6AnO4qsVx5H3ctEOoBRTaYU5EcCxkB3n
 -----END CERTIFICATE-----
 subject=/CN=kubernetes-master
 issuer=/C=US/ST=New York/L=Armonk/O=IBM Cloud Private/CN=www.ibm.com
 ---
 Acceptable client certificate CA names
 /C=US/ST=New York/L=Armonk/O=IBM Cloud Private/CN=www.ibm.com
 /CN=demo.icp
 Client Certificate Types: RSA sign, ECDSA sign
 Requested Signature Algorithms:
RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA512:ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1
 Shared Requested Signature Algorithms:
RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA512:ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1
 Peer signing digest: SHA512
 Server Temp Key: X25519, 253 bits
 ---
 SSL handshake has read 2156 bytes and written 281 bytes
 Verification error: unable to verify the first certificate
 ---
 New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
 Server public key is 2048 bit
 Secure Renegotiation IS supported
 Compression: NONE
 Expansion: NONE
 No ALPN negotiated
 SSL-Session:
     Protocol  : TLSv1.2
     Cipher    : ECDHE-RSA-AES128-GCM-SHA256
     Session-ID: 0BC723C503CE047AECD13FEBC2AEA3A6C4B2B62F82BDF30B78A3E1EE099179CB
     Session-ID-ctx:
     Master-Key:
3844BC9E421A35462C71303631157D1C7D37EEBC419099ECA2924615953B6EFCEA79B8A87C4CE7B37ECF1C0B8BE9358
6
     PSK identity: None
     PSK identity hint: None
     SRP username: None
     TLS session ticket:
     0000 - 18 a2 6b 84 e3 6b 9c 8b-d9 d9 01 d7 68 ee 22 ad   ..k..k......h.".
     0010 - da 3d 12 03 26 82 fc 30-ca 8b 56 6e 8b 47 ff c7   .=.. & .. 0 .. Vn.G..
     0020 - 3b 01 b4 6a 8d b1 a9 a8-80 55 67 46 51 d3 2f b0   ;..j ..... UgFQ ./.
     0030 - f6 7d ff 9d 3f 29 c7 9d-35 3a a3 7a 4e 5f b7 0e   .}..?)..5: .zN_..
     0040 - 22 7e 05 35 e1 a4 46 4e-e7 ff 94 3b bd ca d0 7b   "~.5..FN...;...{
     0050 - 47 c0 85 2f ea c5 44 f1-b0 81 bf 30 7d 93 df af   G../..D.... 0 }...
     0060 - eb 61 89 33 dc 33 c6 1f-b2 e5 5b 3b bc c8 35 c2   .a.3.3....[;..5.
     0070 - c6 2d a9 47 a6 a8 53 40-                          .-.G..S@

     Start Time: 1539239373
     Timeout   : 7200 (sec)
     Verify return code: 21 (unable to verify the first certificate)
     Extended master secret: no
 ---
```

Nota: deve-se substituir o IP `9.111.254.123` pelo seu próprio IP do host principal.

kubelet

É possível especificar as cifras TLS suportadas para uso na comunicação entre o kubelet e os aplicativos, por exemplo, Prometheus.

1. Em `config.yaml`, inclua a opção a seguir:

```
 kubelet_extra_args: [ "--tls-cipher-suites=TLS_ECDHEDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" ]
```

Os valores possíveis são:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305
- TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305
- TLS_ECDHE_HEDHE_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- , TLS_RSA_WITH_AES_128_CBC_SHA
- , TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_GCM_SHA384
- , TLS_RSA_WITH_RC4_128_SHA

2. Quando o cluster IBM® Cloud Private estiver em execução, será possível verificar se os conjuntos de criptografia foram aplicados.

```
# openssl s_client -connect 9.111.255.33:10250
CONNECTED(00000003)
depth=1 CN = 9.111.255.33-ca@1538050035
verify error:num=19:self signed certificate in certificate chain
---
Certificate chain
0 s:/CN=9.111.255.33@1538050035
i:/CN=9.111.255.33-ca@1538050035
1 s:/CN=9.111.255.33-ca@1538050035
i:/CN=9.111.255.33-ca@1538050035
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDCDCCAfCgAwIBAgIBAjANBgkqhkiG9w0BAQsFADAlMSMwIQYDVQQDDBo5LjEx
MS4yNTUuMzMtY2FAMTUzODA1MDAzNTAeFw0xODA5MjcxMjA3MTVaFw0xOTA5Mjcx
MjA3MTVaMCIxIDAeBgNVBAMMFzkuMTExLjI1NS4zM0AxNTM4MDUwMDM1MIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvhpTqz26o/iAmQ2vvn/VbsqjJpno
P5DSOPaf4mCK0iClLj0hFPWplcPO4HmtuigfnC36ChTHQKKycdeUlLL6Fkth7F5K
dyYehMFA7jqUEppmf5DVit2EHusshg7mzGy0irUFGIpaV8loyKo9PE+pOpLaeLm0
j/Jq5qFVvT7lRoEP6/fmWuu2uUVsRMaluY8iVq2DMMsk4LvGH6a2qyzf0t2+TeYw
sCpz2z5s7b0L+66/dJibqlpJvO0SgjdLItjUPZSM9XQ2AzPInpZVKKjkrWH1fQNy
rlqzaJKm8dss2ZgGQ5dd8Nh0JWvMf0pV183S5o2fROzbfaBgTrQMRAhfEQIDAQAB
o0YwRDAOBgNVHQ8BAf8EBAMCBaAwEwYDVR0lBAwwCgYIKwYBBQUHAwEwDAYDVR0T
AQH/BAIwADAPBgNVHREECDAGhwQJb/8hMA0GCSqGSIb3DQEBCwUAA4IBAQBP+F4T
AGOetM9sFPwLga9HWAtG7ukgtNu4RNoc7WnBGrAOUkanTBVxNqnf382NeXoWVFay
WDYUsMVvMkfV5caGWtv0bxv0/zrDEu3S+l65pD7Tmofi7r0sjlCJ3q6PLPhmRNVm
4W7F+6lnVxLvgDQoMFNkRVFSSmM9WBBBvdsAk4YQ9ODG1fykObTBLHm45aADn/4Z
pdtQiqG1BSZKVN23jgqv4vmFfbpSCeLLZL5wqQn1gWJCnCMqk8XQFzRgn1Ye4jwP
eIgDDETuAhSoJFOlWmDoHWdWXMbsMYCNWxaSJA8oZjvqgzSJ+STgZEoIJrTQ+BOz
Ydindji7Vz6vovfV
-----END CERTIFICATE-----
subject=/CN=9.111.255.33@1538050035
issuer=/CN=9.111.255.33-ca@1538050035
---
Acceptable client certificate CA names
/C=US/ST=New York/L=Armonk/O=IBM Cloud Private/CN=www.ibm.com
Client Certificate Types: RSA sign, ECDSA sign
Requested Signature Algorithms:
RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA512:ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1
```

```
 Shared Requested Signature Algorithms:
RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA512:ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1
 Peer signing digest: SHA512
 Server Temp Key: X25519, 253 bits
 ---
 SSL handshake has read 2239 bytes and written 281 bytes
 Verification error: self signed certificate in certificate chain
 ---
 New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
 Server public key is 2048 bit
 Secure Renegotiation IS supported
 Compression: NONE
 Expansion: NONE
 No ALPN negotiated
 SSL-Session:
     Protocol  : TLSv1.2
     Cipher    : ECDHE-RSA-AES128-GCM-SHA256
     Session-ID: 43CE40B2F90AD58A04FBD25850D9C8B9444324ACB2E6FCE8AF5C5B51CB556069
     Session-ID-ctx:
     Master-Key:
6AE72C0F8E9CF2DAB8D07FE6885AE76E97FE0C2462E1B4FFD42A86825913D53A6518304CC37F61667365BEE543FEA86
9
     PSK identity: None
     PSK identity hint: None
     SRP username: None
     TLS session ticket:
     0000 - 7c 40 1d 7d b5 e9 67 a4-50 44 06 b3 f2 70 14 a2   |@.}..g.PD ... p..
     0010 - f2 43 ab 8e 1b 06 f4 b0-d8 99 71 c1 50 f2 88 c8   .C........q.P...
     0020 - 16 e5 4a 56 71 ca 65 c4-59 d8 51 ce 43 90 e7 84   ..JVq.e.Y.Q.C...
     0030 - 81 1f d0 dc 99 cd bd fd-8a b8 b3 7e 73 db 42 53   ...........~s.BS
     0040 - 3d f3 a8 68 45 0a 83 fb-a6 64 26 70 28 d4 3f 4d   =..hE....d&p(.?M
     0050 - b8 73 45 e9 0a 5d 6d db-09 e4 fd 8b 04 97 6e 53   .sE..]m.......nS
     0060 - 17 e4 f9 eb ea 12 05 4e-1d 6c cd 20 b5 ee ed 54   ....... N.l. ...T
     0070 - ac a0 d6 32 2d ab 42 12-                          ...2-.B.

     Start Time: 1539240039
     Timeout   : 7200 (sec)
     Verify return code: 19 (self signed certificate in certificate chain)
     Extended master secret: no
 ---
```

Nota: deve-se substituir o IP `9.111.255.33` pelo seu próprio IP do host do trabalhador.

Configurando o IBM Multicloud Manager durante a instalação do

IBM Cloud Private

É possível configurar o IBM Multicloud Manager durante a instalação do IBM Cloud Private customizando seu arquivo `config.yaml`. Consulte Customizando o cluster com o arquivo config.yaml para saber mais sobre as definições de configuração que estão disponíveis durante a instalação. Para obter mais tópicos do IBM Multicloud Manager e, para assegurar que seus clusters estejam preparados, consulte Preparando-se para a instalação do IBM Multicloud Manager.

- Configurando o hub-cluster do IBM Multicloud Manager com o arquivo config.yaml
- Configurando o managed-cluster do IBM Multicloud Manager com o arquivo config.yaml

Configurando o hub-cluster do IBM Multicloud Manager com o arquivo

`config.yaml`

Siga o processo para mudar suas configurações em seu arquivo `config.yaml`, que está localizado na pasta `/<installation_directory>/cluster`.

Por padrão, a opção `multicluster-hub` é `enabled` e a opção `single_cluster_mode` é `true`, mas não é possível usar o IBM Multicloud Manager com a configuração padrão `single_cluster_mode`.

1. Localize a opção `single_cluster_mode` no arquivo `config.yaml` e configure o valor como `false`, conforme exibido no exemplo a seguir:

   ```
   single_cluster_mode: false
   ```

2. Opcional: para usar o armazenamento persistente, é necessário configurar o volume de persistência local para o ETCD do IBM Multicloud Manager.

3. Crie a configuração a seguir para `multicluster-hub` incluindo a sub-rotina no arquivo `config.yaml`. É possível incluir o valor em qualquer lugar fora de sua seção `management_services`:

```
multicluster-hub:
 etcd:
   persistence: true
   localPath: /var/lib/etcd-mcm
```

4. Efetue login em seu nó de gerenciamento e crie o diretório `/var/lib/etcd-mcm`. Deve-se repetir esta etapa para todos os seus nós de gerenciamento.

Opcional: também é possível gerenciar seu hub-cluster com o procedimento a seguir, que ativa o `multicluster-endpoint`.

Configurando o managed-cluster do IBM Multicloud Manager com o arquivo

`config.yaml`

Continue com o procedimento para ativar o `multicluster-endpoint` em seu cluster.

1. No arquivo `config.yaml` para o novo cluster do IBM Cloud Private, que está localizado na pasta `/<installation_directory>/cluster`, ative o `multicluster-endpoint`, como no seguinte exemplo:

```
management_services:
 multicluster-endpoint: enabled
```

2. Continue para criar a sub-rotina com as seguintes configurações para `multicluster-endpoint`:

```
multicluster-endpoint:
 global:
   clusterName: "{{ cluster_name }}"
   clusterNamespace: "{{ cluster_name }}"
 clusterLabels:
   environment: "Dev"
   region: "US"
   datacenter: "toronto"
   owner: "marketing"
 operator:
   bootstrapConfig:
     hub0:
       name: hub0
       secret: kube-system/klusterlet-bootstrap
     hub1:
       name: null
       secret: null
 klusterlet:
   host: null
 prometheusIntegration:
   enabled: true
 policy:
   cemIntegration: false
 topology:
   enabled: true
 serviceRegistry:
   enabled: true
   dnsSuffix: "mcm.svc"
   plugins: "kube-service"
```

3. Salve e saia do arquivo. Conclua o procedimento de instalação.

Configurando a instalação do IBM Multicloud Manager IBM Cloud Private

Se você já instalou o IBM Cloud Private com a opção `single_cluster_mode` configurada para o valor padrão `false`, não será possível usar o IBM Multicloud Manager. No entanto, é possível ativar e usar o IBM Multicloud Manager após a instalação. Para obter mais tópicos do IBM Multicloud Manager e, para assegurar que seus clusters estejam preparados, consulte Preparando-se para a instalação do IBM Multicloud Manager.

Configurando o hub-cluster do IBM Multicloud Manager após a instalação

1. Efetue login na console de gerenciamento do IBM Cloud Private e clique em **Cargas de Trabalho > Liberações do Helm**. Localize o `multicluster-hub`, que é o nome da liberação do hub-cluster.
2. Clique em **Fazer upgrade** para a liberação do multicluster-hub e marque **Ativar back-end > Ativar ETCD** e outras funcionalidades.
3. Atualize a console de gerenciamento do IBM Cloud Private.
4. Visualize a console de gerenciamento do IBM Cloud Private, na qual o *Multicloud Manager* é exibido na página *Introdução*.

Configurando o managed-cluster do IBM Multicloud Manager

Configure os managed-clusters com o comando `cloudctl mc cluster import` e com os arquivos de configuração. É possível importar clusters de diferentes provedores de nuvem do Kubernetes, incluindo o IBM Cloud Private.

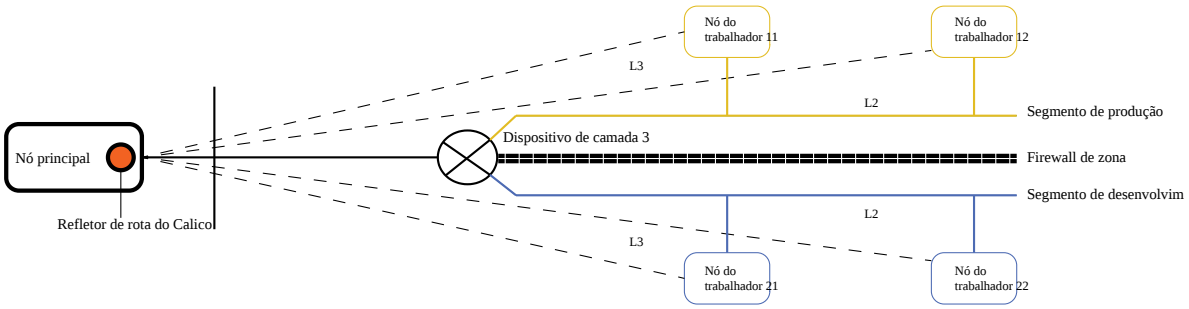
Para saber mais sobre as opções de importação, consulte Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager.

Implementando o IBM Cloud Private em segmentos isolados da Camada 3

Calico é o provedor Container Network Interface (CNI) padrão no IBM® Cloud Private. O Calico espera conectividade da Camada 3 entre todos os nós no cluster. Quando quiser implementar o cluster do IBM® Cloud Private em um ambiente com várias zonas, ou seja, um ambiente com diferentes segmentos da Camada 3, e não quiser conectividade da Camada 3 nessas zonas, você poderá configurar um refletor de rota Calico.

Topologia de exemplo

A ilustração a seguir é uma topologia com diferentes zonas da Camada 3. Uma zona para carga de trabalho de produção e outra zona para carga de trabalho de desenvolvimento ou teste. A terceira zona é uma zona de gerenciamento com conectividade com as outras duas zonas.



A zona de gerenciamento contém nós como principal, de gerenciamento, VA, etcd e refletor de rota.

O refletor de rota Calico pode ser implementado em grupos de hosts dedicados ou em um ou mais nós principais. Por padrão, quando o parâmetro `calico-route-reflector` é ativado, o refletor de rota é implementado nos nós principais em seu cluster.

O refletor de rota Calico elimina a necessidade de uma rede Protocolo de Roteamento de Borda (BGP). Ele é responsável por trocar rotas entre os nós nas zonas isoladas.

Configuração e arquitetura de referência

Considere a seguinte topologia de exemplo:

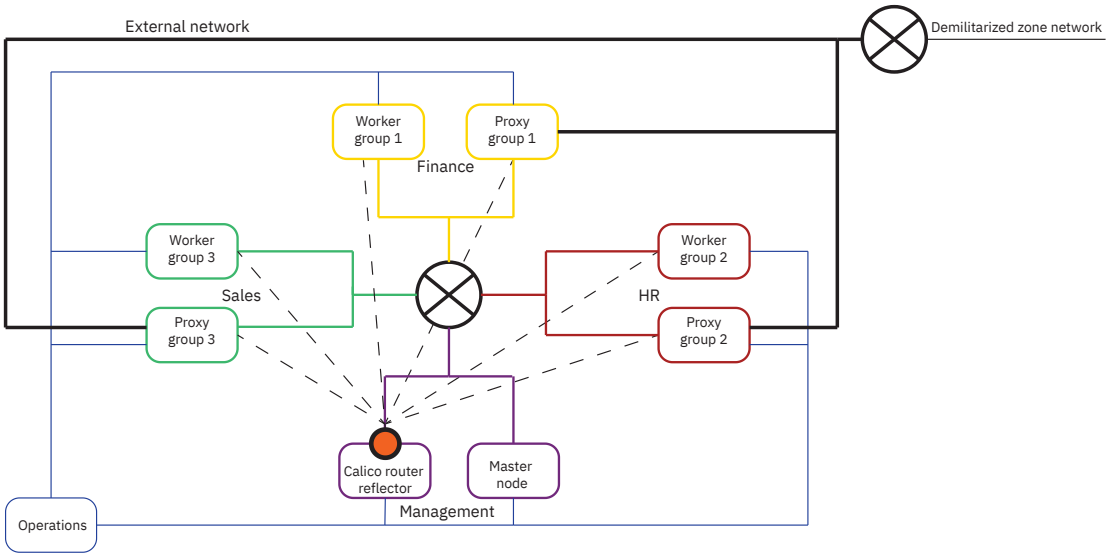


Tabela 1. Legenda para Infraestrutura isolada, localizada e distribuída

Division	Rede Local Virtual (VLAN)	Subrede	Gateway
Venda	300	192.168.30.0/24	192.168.30.1
Finanças	100	192.168.10.0/24	192.168.10.1
HR	200	192.168.20.0/24	192.168.20.1
Nó principal ou de gerenciamento	900	192.168.90.0/24	192.168.90.1
Operações ou administração	800	172.68.0.0/16	172.68.10.1

- A rede externa é opcional. A conversão de endereço de rede de destino (DNAT) para os endereços IP do nó do proxy pode ser usada em roteadores de zona desmilitarizada (DMZ).
- O peer do Protocolo de Roteamento de Borda (BGP) conecta o refletor de rota do Calico a todos os nós do trabalhador e nós do proxy. Quando o túnel IP-in-IP do Calico é usado, as rotas são compartilhadas diretamente nos nós por meio do route-reflector do Calico. Quando o túnel IP-in-IP do Calico não é usado, um roteador de infraestrutura também precisa ser configurado como um peer BGP para o route-reflector do Calico.
- Cada organização ou divisão de negócios possui sua própria infraestrutura isolada e localizada.

arquivo host

A seguir estão os detalhes de configuração no arquivo `<installation_directory>/cluster/hosts`:

```
[master]
192.168.90.2
192.168.90.3
192.168.90.4

[hostgroup-rr]
192.168.90.30
192.168.90.31
192.168.90.32

[proxy]
192.168.90.2
192.168.90.3
192.168.90.4

[hostgroup-sales]
192.168.30.10
192.168.30.11

[hostgroup-sales-proxy]
192.168.30.20
192.168.30.21

[hostgroup-hr]
192.168.20.10
192.168.20.11
```

```
[hostgroup-hr-proxy]
192.168.20.20
192.168.20.21

[hostgroup-finance]
192.168.10.10
192.168.10.11

[hostgroup-finance-proxy]
192.168.10.20
192.168.10.21
```

Arquivo config.yaml

Para a topologia de exemplo, a seguinte parte de código é incluída no arquivo `<installation_directory>/cluster/config.yaml`:

```
network_type: calico

management_services:
  calico-route-reflector: "enabled"

calico_rr_group: rr
```

Para configurar o refletor de rota Calico após a instalação do IBM Cloud Private, consulte Configurando o refletor de rota Calico após a instalação do IBM Cloud Private.

Exemplo: Ativando FIPS no IBM Cloud Private

Implementando FIPS no IBM Cloud Private

Implemente os padrões que são definidos pelo FIPS para uma instalação do IBM Cloud Private. Sua instalação do IBM Cloud Private usa algoritmos altamente seguros para criptografar dados em repouso e dados em movimento.

Nota: antes de implementar o FIPS no IBM Cloud Private, certifique-se de que todos os nós estejam em execução no mesmo sistema operacional. Diferentes sistemas operacionais usam diferentes implementações para criptografar o tráfego de rede com o IPsec. Assegure-se de que seus nós sejam executados no mesmo sistema operacional para evitar incompatibilidades de IPsec.

No exemplo a seguir, as atualizações do FIPS são aplicadas a um cluster de três nós do IBM Cloud Private.

As configurações do sistema operacional para os três nós são listadas na tabela a seguir:

Tabela 1. Cluster de três nós do IBM Cloud Private

Nó	CPU	Memória	Interfaces de Rede	Disco do sistema operacional	IBM Cloud Private Disco
Principal ou gerenciamento	8	16 GB	2	/dev/vda, 250 GB	/dev/vdb, 300 GB
Trabalhador	4	8 GB	2	/dev/vda, 250 GB	/dev/vdb, 300 GB
Trabalhador	4	8 GB	2	/dev/vda, 250 GB	/dev/vdb, 300 GB

Visualize as tarefas a seguir para ativar o FIPS para o exemplo de cluster de três nós do IBM Cloud Private. Se você implementar o FIPS em seu ambiente, conclua as tarefas em ordem.

- Ativando o FIPS em sistemas operacionais que usam o IBM Cloud Private
- Criptografando volumes usados pelo IBM Cloud Private
- Criptografando comunicações executadas pelo IBM Cloud Private
- Verificação

Ativando o FIPS em sistemas operacionais que usam o IBM Cloud Private

Cada fornecedor do sistema operacional envia suas bibliotecas de criptografia para a certificação de conformidade com o FIPS. Para usar as bibliotecas certificadas pelo FIPS, deve-se seguir um procedimento dependente do sistema operacional.

À medida que você aplica os padrões do FIPS a cada sistema operacional, considere os pontos a seguir:

- Os pacotes que são instalados incluem pacotes que são necessários para a ativação do FIPS e também para as configurações `dm-crypt` e `ipsec`.
- Prepare-se para reinicializar seu nó.
- Deve-se aplicar o FIPS a cada um de seus nós.
- À medida que você aplica o padrão FIPS ao Red Hat Enterprise Linux (RHEL), você deve executar os comandos como um usuário root. Ao aplicar o FIPS ao Ubuntu, deve-se executar os comandos com sudo.
- Os procedimentos a seguir não substituem nenhum dos procedimentos de instalação do IBM Cloud Private. Para obter mais informações sobre a instalação, consulte Instalando o IBM Cloud Private.

A conformidade com o FIPS requer que o kernel do Linux® opere no modo FIPS. Aplique o padrão FIPS aos sistemas operacionais a seguir que são suportados pelo IBM Cloud Private:

1. Red Hat Enterprise Linux 7.5
2. Ubuntu 16.0.4 LTS

Atualizações do sistema operacional para o RHEL 7.5

Conclua as seguintes etapas para ativar o FIPS para o RHEL 7.5:

1. Instale os pacotes que são necessários para ativar o FIPS. Execute o comando a seguir:

   ```
   yum install dracut-fips dracut-fips-aesni cryptsetup libreswan yum-utils device-mapper-persistent-data lvm2 curl ca-certificates
   ```

2. Certifique-se de que as instruções do processador de Padrão de Criptografia Avançado estejam disponíveis em seus nós Intel ™. Deve-se ter a funcionalidade `aes` no processador:

   ```
   grep -qw aes /proc/cpuinfo && echo YES || echo no
   YES
   ```

3. Faça backup e crie novamente o `ramdisk` inicial com suporte para FIPS. Execute os comandos a seguir:

   ```
   mv -v /boot/initramfs-$(uname -r).img{,.bak}

   dracut
   ```

4. Para incluir o parâmetro de ativação do FIPS no kernel do Linux, execute os seguintes comandos:

   ```
   grubby -- update-kernel = $(grubby --default-kernel) -- args=fips= 1

   uuid=$ (findmnt -no uuid /boot)

   [[ -n $uuid ]] && grubby --update-kernel=$(grubby --default-kernel) --args=boot=UUID=${uuid}
   ```

   - Apenas no IBM Z, aplique as mudanças na configuração do FIPS executando o comando a seguir:

     ```
     zipl
     ```

5. Reinicialize seu nó. Execute o comando a seguir:

   ```
   reboot
   ```

6. Aguarde que seu sistema reinicialize e, em seguida, efetue login novamente. Verifique se o FIPS está ativado executando o comando a seguir:

   ```
   sysctl crypto.fips_enabled
   ```

   Sua saída deve ser semelhante ao texto a seguir:

   ```
   crypto.fips_enabled = 1
   ```

   Nota: o valor de 1 indica que o FIPS foi ativado com sucesso.

7. (Recomendado) Se você já executou o comando `grub2-mkconfig` manualmente, o FIPS está desativado. Execute os comandos a seguir para assegurar que o FIPS permaneça ativado à medida que você executa o comando `grub2-mkconfig` manualmente:

```
sed -i '/ ^ GRUB_CMDLINE_LINUX=/s/ "$/ fips= 1" /' /etc/default/grub

uuid=$ (findmnt -no uuid /boot)

[[ -n $uuid ]] && sed -i "/^GRUB_CMDLINE_LINUX=/s/\"$/ boot=UUID=${uuid}\"/" /etc/default/grub
```

Para obter informações adicionais para aplicar o FIPS ao RHEL, consulte Tornar o RHEL 6 ou 7 compatível com o FIPS.

Atualizações do sistema operacional para o Ubuntu 16.0.4 LTS

Conclua as etapas a seguir para ativar o FIPS no sistema operacional Ubuntu 16.04 LTS:

1. Configure seu sistema para usar o repositório FIPS a partir do Canonical. Deve-se solicitar e ter acesso concedido ao PPA do FIPS antes de continuar. Execute os comandos a seguir:

   ```
   sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
   A166877412DAC26E73CEBF3FF6C280178D13028C

   sudo add-apt-repository -u 'deb https://<your-launchpad-id>:<PPA-password>@private-
   ppa.launchpad.net/ubuntu-advantage/fips/ubuntu xenial main'

   atualização do apt sudo apt
   ```

2. Execute o comando a seguir para instalar os pacotes do FIPS:

   ```
   sudo apt install openssh-client openssh-client-hmac openssh-server openssh-server-hmac openssl
   libssl1.0.0 libssl1.0.0-hmac fips-initramfs linux-fips strongswan strongswan-hmac cryptsetup
   ```

3. Defina a configuração do GRUB para uma partição `/boot` executando os comandos a seguir:

   ```
   cat /etc/fstab | inicialização grep
   ```

   Sua saída pode ser semelhante ao texto a seguir:

   ```
   /boot estava no /dev/vda1 durante a instalação

   UUID=87d50882-8bcc-4951-820f-e6e446b134c4 /boot           ext2    defaults        0       2
   ```

   Continue definindo a configuração do GRUB executando os comandos a seguir:

   ```
   mkdir -p /etc/default/grub.d

   cd /etc/default/grub.d/
   ```

4. Mude o valor de `UUID` para o valor `UUID` na etapa *Configuração de GRUB*. Execute os comandos a seguir:

   ```
   sudo echo 'GRUB_CMDLINE_LINUX_DEFAULT = "$GRUB_CMDLINE_LINUX_DEFAULT fips= 1 bootdev = UUID=
   87d50882-8bcc-4951-820f-e6e446b134c4"' > 99-fips.cfg

   sudo update-grub
   ```

   Sua saída pode ser semelhante ao texto a seguir:

   ```
   Gerando arquivo de configuração grub ...

   Localizada imagem do linux: /boot/vmlinuz-4.4.0-1002-fips

   Localizada imagem de initrd: /boot/initrd.img-4.4.0-1002-fips

   Localizada imagem do linux: /boot/vmlinuz-4.4.0-133-generic

   Localizada imagem de initrd: /boot/initrd.img-4.4.0-133-generic

   Localizada imagem do linux: /boot/vmlinuz-4.4.0-21-generic

   Localizada imagem de initrd: /boot/initrd.img-4.4.0-21-generic

   done
   ```

5. Reinicialize seu nó. Execute o comando a seguir:

   ```
   sudo reboot
   ```

6. Efetue login novamente e verifique se o FIPS está ativado executando o comando a seguir:

```
cat /proc/sys/crypto/fips_enabled
```

Sua saída deve ser semelhante ao texto a seguir:

```
1
```

Nota: a saída de 1 significa que o FIPS está ativado adequadamente.

7. Verifique se os pacotes do FIPS não foram substituídos. Execute o comando a seguir:

```
sudo apt-mark hold openssh-client openssh-client-hmac

openssh-client set on hold.

openssh-client-hmac set on hold.
```

Para obter mais informações, consulte FIPS para o Ubuntu 16.04.

Continue implementando e ative o FIPS para o exemplo. Consulte Exemplo: ativando o FIPS no IBM Cloud Private para obter mais detalhes.

Criptografando volumes que são usados pelo IBM Cloud Private

Criptografe os sistemas de arquivos usados pelo IBM Cloud Private com a criptografia Linux® Unified Key Setup (LUKS) no Linux. Assegure-se de que seu sistema tenha espaço em disco disponível. Consulte Requisitos de espaço em disco para obter mais informações.

À medida que você criptografa os diretórios que você deseja usar com o IBM Cloud Private, um sistema de arquivos é criptografado e os diretórios a seguir são montados em seu sistema de arquivos criptografado:

- /etc/cfc
- /var/lib/etcd
- /var/lib/icp
- /opt/ibm
- /var/lib/registry
- /var/lib/kubelet
- /var/lib/docker

Nota: `/var/lib/kubelet` e `/var/lib/docker` não são necessários para seu ambiente, se você estiver criptografando apenas dados em repouso do IBM Cloud Private.

Para criptografar um sistema de arquivos em todos os seus nós do IBM Cloud Private, conclua as etapas a seguir:

1. Para o exemplo, `/dev/vdb` é incluído no sistema. Para visualizar os dispositivos de bloco em seu ambiente, execute o comando a seguir:

```
lsblk
```

A saída pode ser semelhante ao conteúdo a seguir:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda           252:0    0   250G  0 disk

├─vda1        252:1    0     1G  0 part /boot

└─vda2        252:2    0 248.9G  0 part

    ├─rhel-root 253:0    0   241G  0 lvm  /

    └─rhel-swap 253:1    0   7.9G  0 lvm  [SWAP]
vdb           252:16   0   300G  0 disk
```

O disco `vda` possui partições que são criadas nele, já que ele é usado pelo sistema operacional. O disco `vdb` não é usado atualmente.

2. Crie um sistema de arquivos criptografado no disco `vdb` não usado. Conclua as etapas a seguir:

- Crie um grupo de volumes executando o comando a seguir:

  ```
  vgcreate CloudVG /dev/vdb
  ```

  Sua saída pode ser semelhante ao texto a seguir:

  ```
  Physical volume "/dev/vdb" successfully created
  Volume group "CloudVG" successfully created
  ```

- Crie um volume lógico que usa o espaço disponível na unidade não usada. Execute o comando a seguir:

  ```
  lvcreate -- size 250G -- name Data CloudVG
  ```

  A saída pode ser semelhante ao conteúdo a seguir:

  ```
  Volume lógico "Dados" criado
  ```

- A criptografia de LUKS requer que uma senha seja associada aos volumes criptografados. Crie um arquivo que contenha uma senha executando os comandos a seguir:

  ```
  echo 'passw0rd' > /root/.luks_key

  chmod 400 /root/.luks_key
  ```

- Execute o comando a seguir para criar um contêiner LUKS `dm-crypt` no volume com o arquivo-chave:

  ```
  cryptsetup luksFormat --batch-mode --use-random /dev/CloudVG/Data /root/.luks_key
  ```

- Abra o contêiner LUKS e mapeie o volume lógico para seu caminho:

  ```
  cryptsetup luksOpen -- key-file /root/.luks_key /dev/CloudVG/Data luks-data
  ```

- Crie um sistema de arquivos no volume lógico (formate a partição) e configure-o para que seja montado após a reinicialização do nó.

  - Crie um sistema de arquivos no volume lógico. Execute o comando a seguir:

    ```
    mkfs.ext4 /dev/mapper/luks-data
    ```

  - Configure seu sistema de arquivos a ser montado. Execute os comandos a seguir:

    ```
    echo "luks-data /dev/CloudVG/Data /root/.luks_key" > > /etc/crypttab

    echo "/dev/mapper/luks-data    /data    ext4    defaults    1 2" >> /etc/fstab
    ```

  Nota: não é necessário usar o sistema de arquivos `ext4`.

- Verifique se o volume criptografado foi configurado executando o comando a seguir:

  ```
  cryptsetup status /dev/mapper/luks-data
  ```

  A saída pode ser semelhante ao conteúdo a seguir:

  ```
  /dev/mapper/luks-os dados estão ativos. type:    LUKS1
  cipher:  aes-xts-plain64
  keysize: 256 bits
  device:  /dev/mapper/CloudVG-Data offset:  4096 sectors
  size:    524283904 sectors
  mode:    read/write\
  ```

3. Crie os diretórios que são usados pelo IBM Cloud Private e monte-os em seu arquivo criptografado. Execute os comandos a seguir:

   - Crie um diretório para montar os diretórios usados pelo IBM Cloud Private no seu volume criptografado. Execute o comando a seguir:

     ```
     mkdir /data
     mount /dev/mapper/luks-data /data
     ```

     - Crie diretórios a serem montados no volume criptografado. Execute os comandos a seguir:

     ```
     mkdir -p /var/lib/etcd /var/lib/icp /var/lib/registry /var/lib/kubelet /var/lib/docker
     /etc/cfc /opt/ibm
     ```

```
mkdir -p /data/var/lib/etcd /data/var/lib/icp /data/var/lib/registry
/data/var/lib/kubelet /data/var/lib/docker /data/etc/cfc /data/opt/ibm
```

- Inclua as entradas de montagem bind no arquivo `/etc/fstab`. Execute o comando a seguir:

```
echo "/data/opt/ibm /opt/ibm none bind 0 0" >> /etc/fstab
echo "/data/etc/cfc /etc/cfc none bind 0 0" >> /etc/fstab
echo "/data/var/lib/registry /var/lib/registry none bind 0 0" >> /etc/fstab
echo "/data/var/lib/kubelet /var/lib/kubelet none bind 0 0" >> /etc/fstab
echo "/data/var/lib/docker /var/lib/docker none bind 0 0" >> /etc/fstab
echo "/data/var/lib/icp /var/lib/icp none bind 0 0" >> /etc/fstab
echo "/data/var/lib/etcd /var/lib/etcd none bind 0 0" >> /etc/fstab
```

- Execute os comandos a seguir para efetuar montagem bind dos diretórios no mapeamento correspondente no arquivo `/data`:

```
mount --bind /data/var/lib/etcd /var/lib/etcd/
mount --bind /data/var/lib/icp/ /var/lib/icp/
mount --bind /data/var/lib/registry/ /var/lib/registry/
mount --bind /data/var/lib/docker/ /var/lib/docker/
mount --bind /data/var/lib/kubelet/ /var/lib/kubelet/
mount --bind /data/etc/cfc/ /etc/cfc/
mount --bind /data/opt/ibm/ /opt/ibm/
```

4. Reinicialize seu nó. Depois de reinicializar seu nó, as montagens bind são recriadas automaticamente. Execute o comando a seguir:

```
reboot
```

5. Depois de reinicializar seu nó, efetue login e verifique se todos os diretórios estão montados no sistema de arquivos `/data`. Execute o comando a seguir:

```
mount | grep luks-data
```

A saída pode ser semelhante ao conteúdo a seguir:

```
/dev/mapper/luks-data on /data type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /var/lib/etcd type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /var/lib/kubelet type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /opt/ibm type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /var/lib/icp type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /var/lib/registry type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /var/lib/docker type ext4 (rw,relatime,data=ordered)
/dev/mapper/luks-data on /etc/cfc type ext4 (rw,relatime,data=ordered)
```

Para obter informações adicionais sobre requisitos de disco do IBM Cloud Private, consulte Criptografando volumes usando dm-crypt.

Continue implementando e ative o FIPS para o exemplo. Consulte Exemplo: ativando o FIPS no IBM Cloud Private para obter mais detalhes.

Criptografando comunicações executadas pelo IBM Cloud Private

As comunicações de rede no ambiente do IBM Cloud Private devem ser criptografadas para conformidade com o FIPS.

Criptografe o tráfego de rede de dados do cluster com o IPsec e ative o modo FIPS para criptografia TLS de tráfego de rede para terminais externos. Conclua as etapas a seguir:

1. Instale o Docker somente para o nó de inicialização. Para obter mais informações, consulte Instalando o IBM Cloud Private Native, Enterprise e Community Editions.

2. À medida que você configura o ambiente de instalação, certifique-se de que o diretório /opt/ibm/icp esteja sendo usado como o diretório de instalação. Verifique se o diretório existe executando os comandos a seguir:

```
mkdir -p /opt/ibm/icp
```

```
cd /opt/ibm/icp
```

3. Customize a configuração do do cluster e ative o IPsec. Consulte ativar a criptografia do tráfego de rede de dados do cluster com o IPsec para obter mais informações. Edite o arquivo `/opt/ibm/icp/cluster/config.yaml` e inclua os

valores a seguir nos parâmetros:

```
## Configurações de Rede

## Calico Network Settings

calico_ipip_mode: Sempre

calico_tunnel_mtu: 1390

calico_ip_autodetection_method: interface=eth0
```

Nota: certifique-se de usar a interface correta para o `calico_ip_autodetection_method`. Deve-se criptografar a interface para comutações entre pods.

Configurando o IPsec

Criptografe a rede de dados do cluster com o IPsec. Para configurar o IPsec, conclua as etapas a seguir:

1. Localize e edite as `Configurações de malha do IPSec` em seu arquivo `config.yaml`. Inclua os valores a seguir nos parâmetros:

   ```
   ## IPSec mesh Settings
   ## If user wants to configure IPSec mesh, the following parameters
   ## should be configured through config.yaml
   ipsec_mesh:
     enable: true
     subnets: ["172.16.0.0/16"]
     exclude_ips: [ "172.16.200.0/24" ]
   ```

   A lista a seguir descreve os parâmetros para as `Configurações de malha do IPSec`:

   - `subnets`: uma lista de sub-redes criptografadas que são endereços IP no conjunto de nós de seu cluster. Os endereços de sub-rede não devem se sobrepor com os endereços IP de gerenciamento que estão incluídos no arquivo `hosts` do cluster.
   - `exclude_ips`: uma lista de endereços IP do parâmetro `subnets` que não devem ser criptografados com o IPsec.

     Nota: considere quais outras comunicações não podem ser criptografadas a partir da lista de sub-redes e exclua os endereços. Os endereços `172.16.200.0/24` são excluídos porque os servidores de nome que são usados pelos nós estão na sub-rede.

2. Verifique se o arquivo `hosts` faz referência aos endereços IP de gerenciamento dos nós do IBM Cloud Private. A configuração de IPsec não deve se sobrepor com o arquivo `hosts` do IBM Cloud Private.

   A configuração de `hosts` do cluster pode ser semelhante ao conteúdo a seguir:

   ```
   [master]

   192.168.160.145

   [ trabalhador ]

   192.168.160.157

   192.168.160.206

   [ proxy ]

   192.168.160.145
   ```

   Nota: a configuração de IPsec definida no arquivo `config.yaml` é para a rede de dados `eth0`.

Para obter mais detalhes sobre como configurar o arquivo de hosts, consulte Configurando as funções de nó no arquivo de hosts.

1. Ative o modo FIPS nos componentes no IBM Cloud Private. Em seu arquivo `config.yaml`, localize e edite o parâmetro `fips_enabled`. Seu arquivo `config.yaml` pode ser semelhante ao texto a seguir:

   ```
   fips_enabled: true
   ```

   Depois de ativar o FIPS, os componentes a seguir no IBM Cloud Private estão no modo FIPS:

- Ingresso de gerenciamento do IBM Cloud Private (console de gerenciamento)
- Controlador de ingresso NGINX (serviço de ingresso)
- Registro de Docker
- Image Manager
- WebSphere Liberty Application Server (gerenciador de autenticação)

Depois de configurar o IPsec e de ativar o FIPS, salve o arquivo `config.yaml` e continue instalando o IBM Cloud Private.

Continue implementando e ative o FIPS para o exemplo. Consulte Exemplo: ativando o FIPS no IBM Cloud Private para obter mais detalhes.

Verificação de ativação de FIPS no IBM Cloud Private

Verifique se o FIPS está ativado no IBM Cloud Private.

- Verifique o sistema operacional
- Verificar a criptografia do volume de armazenamento
- Verificando a funcionalidade IPsec
- A criptografia TLS

Verifique o sistema operacional

Verifique se o FIPS está ativado em seu sistema operacional. Conclua as etapas a seguir:

1. Visualize os parâmetros que foram transmitidos para o kernel. Execute o comando a seguir:

   ```
   cat /proc/cmdline
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   BOOT_IMAGE=/vmlinuz-4.4.0-1002-fips root=/dev/mapper/ubuntu -- vg-root ro elevator=1 bootdev =
   UUID= 87d50882-8bcc-4951-820f-e6e446b134c4
   ```

   O parâmetro `fips=1` indica que o kernel está inicializado no modo ativado para FIPS.

2. Verifique se seu kernel está configurado para FIPS. Execute o comando a seguir:

   ```
   sysctl crypto.fips_enabled
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   crypto.fips_enabled = 1
   ```

   O `crypto.fips_enabled=1` indica que o kernel está configurado para FIPS.

3. Verifique se o pacote OpenSSL é certificado pelo FIPS. Execute o comando a seguir:

   ```
   versão openssl
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   OpenSSL 1.0.2k-fips 26 Jan 2017
   ```

Seu sistema operacional está no modo ativado pelo FIPS.

Verificar criptografia de volume

Verifique se os volumes de armazenamento estão criptografados adequadamente com a criptografia LUKS. Conclua as etapas a seguir:

1. Verifique se uma partição LUKS existe em cada sistema em seu cluster. Execute o comando a seguir em cada nó:

   ```
   lsblk
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda             253:0    0  250G  0 disk

└─vda1          253:1    0  250G  0 part

 ├─system-swap 254:0    0    8G  0 lvm   [SWAP]

 └─system-root 254:1    0  220G  0 lvm   /var/lib/kubelet/pods/29467a76-e1e6-11e8-998b-
00163e01b777/volume-subpaths/logrotate-conf/icp

vdb             253:16   0  300G  0 disk

└─CloudVG-Data  254:2    0  250G  0 lvm

 └─luks-data   254:3    0  250G  0 crypt /data
```

Nota: a partição `luks-data` é montada no diretório `/data`. `luks-data` está criptografado.

2. Verifique se a instalação do IBM Cloud Private usa seu sistema de arquivos com criptografia. Execute o comando a seguir:

   ```
   mount | grep luks-data
   ```

   Sua saída deve mostrar os diretórios usados pelo IBM Cloud Private. Seu sistema pode retornar entradas adicionais. A saída pode ser semelhante ao conteúdo a seguir:

   ```
   /dev/mapper/luks-data on /data type ext4 (rw, relatime, data = ordered)

   /dev/mapper/luks-data on /var/lib/etcd type ext4 (rw, relatime, data = ordered)

   /dev/mapper/luks-data on /var/lib/kubelet type ext4 (rw, relatime, data = ordered)

   /dev/mapper/luks-data on /opt/ibm type ext4 (rw, relatime, data = ordered)

   /dev/mapper/luks-data on /var/lib/icp type ext4 (rw, relatime, data = ordered)

   /dev/mapper/luks-data on /var/lib/registry type ext4 (rw, relatime, data = ordered)

   /dev/mapper/luks-data on /var/lib/docker type ext4 (rw,relatime,data=ordered)

   /dev/mapper/luks-data on /etc/cfc type ext4 (rw, relatime, data = ordered)
   ```

3. Verifique o status da criptografia. Execute o comando a seguir:

   ```
   cryptsetup status /dev/mapper/luks-data
   ```

   Sua saída pode ser semelhante ao texto a seguir:

   ```
   /dev/mapper/luks-data is active and is in use.
   type:    LUKS1
   cipher:  aes-xts-plain64
   keysize: 256 bits
   device:  /dev/mapper/CloudVG-Data
   offset:  4096 sectors
   size:    524283904 sectors
   mode:    read/write
   ```

   Nota: o LUKS sempre usa uma cifra que é compatível com FIPS 140-2. Para obter mais informações, consulte LUKS RedHat.

Seus volumes de armazenamento são criptografados com a criptografia LUKS.

Verificando a funcionalidade do IPsec

Verifique se o IPsec está protegendo seu tráfego de rede. Conclua as etapas a seguir:

1. Verifique se o serviço IPsec está em execução. Verifique o status do serviço IPsec em seu sistema operacional.
   - Para o Red Hat Linux®, execute o seguinte comando:

     ```
     systemctl status ipsec.service
     ```

   - Para Ubuntu, execute o comando a seguir:

```
 systemctl status strongswan
```

A saída pode ser semelhante ao conteúdo a seguir:

```
ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec

  Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled; vendor preset:
disabled)
  Active: active (running) since Mon 2018-10-29 08:25:39 PDT; 34min ago
      Docs: man:ipsec(8)
          man:pluto (8)
          man:ipsec.conf (5)
  Process: 25415 ExecStartPre=/usr/sbin/ipsec --checknflog (code=exited,
status=0/SUCCESS)
  Process: 25410 ExecStartPre=/usr/sbin/ipsec --checknss (code=exited, status=0/SUCCESS)
  Process: 25131 ExecStartPre=/usr/libexec/ipsec/_stackmanager start (code=exited,
status=0/SUCCESS)
  Process: 25129 ExecStartPre=/usr/libexec/ipsec/addconn --config /etc/ipsec.conf --
checkconfig (code=exited, status=0/SUCCESS)
  Main PID: 25426 (pluto)
    Status: "Startup completed."
```

Nota: se o serviço IPsec não estiver em execução, inicie o serviço agora e certifique-se de que ele esteja configurado para iniciar após a reinicialização de seu nó.

2. Verifique se o IPsec está em execução no modo FIPS. Verifique o status do IPsec em execução no modo FIPS para seu sistema operacional.

   - Para o Red Hat Linux, execute o seguinte comando:

     ```
     ipsec status | grep fips
     ```

     Sua saída pode ser semelhante ao texto a seguir, se o FIPS estiver ativado:

     ```
     000 fips mode=enabled;
     ```

   - Para Ubuntu, execute o comando a seguir:

     ```
     ipsec statusall | grep -i fips
     ```

     A saída pode ser semelhante ao conteúdo a seguir:

     ```
     Status of IKE charon daemon (strongSwan 5.3.5, Linux 4.4.0-1002-fips, x86_64):
     loaded plugins: charon test-vectors nonce x509 revocation constraints pubkey pkcs1 pkcs7
     pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf agent gcm attr kernel-netlink resolve
     socket-default connmark farp stroke updown eap-identity eap-sim eap-sim-pcsc eap-aka eap-
     aka-3gpp2 eap-simaka-pseudonym eap-simaka-reauth eap-md5 eap-gtc eap-mschapv2 eap-dynamic
     eap-radius eap-tls eap-ttls eap-peap eap-tnc xauth-generic xauth-eap xauth-pam xauth-
     noauth tnc-tnccs tnccs-20 tnccs-11 tnccs-dynamic dhcp lookip error-notify certexpire led
     addrblock unity
     ```

3. Verifique se o serviço IPsec está criptografando o tráfego.

   1. Instale o tcpdump para visualizar os dados na rede.

   2. Visualize e verifique se os pacotes em sua interface configurada estão criptografados.

      Nota: o `eth0` é a interface configurada neste exemplo. Certifique-se de usar a interface que tenha o IPsec definido para a configuração de instalação do IBM Cloud Private.

      Execute o comando a seguir:

      ```
      tcpdump -i eth0 | grep ESP
      ```

      A saída pode ser semelhante ao conteúdo a seguir:

      ```
      tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on
      eth0, link-type EN10MB (Ethernet), capture size 262144 bytes 12:43:58.255908 IP wap-
      worker-1.fyre.ibm.com > wap-master.fyre.ibm.com: ESP(spi=0x1d59f23e,seq=0x3647), length
      88
      12:43:58.255995 IP wap-master.fyre.ibm.com > wap-worker-1.fyre.ibm.com:
      ESP(spi=0xd02dc971,seq=0x19d2), length 68
      12:43:59.667642 IP wap-worker-2.fyre.ibm.com > wap-master.fyre.ibm.com:
      ESP(spi=0x0cedc094,seq=0x53d4), length 156
      ```

```
12:43:59.667823 IP wap-worker-2.fyre.ibm.com > wap-master.fyre.ibm.com:
ESP(spi=0x0cedc094,seq=0x53d5), length 156
12:43:59.667862 IP wap-worker-2.fyre.ibm.com > wap-master.fyre.ibm.com:
ESP(spi=0x0cedc094,seq=0x53d6), length 236
12:43:59.667969 IP wap-worker-2.fyre.ibm.com > wap-master.fyre.ibm.com:
ESP(spi=0x0cedc094,seq=0x53d7), length 236
12:43:59.668594 IP wap-master.fyre.ibm.com > wap-worker-2.fyre.ibm.com:
ESP(spi=0x3fd0dc47,seq=0x2358), length 88
12:43:59.668634 IP wap-master.fyre.ibm.com > wap-worker-2.fyre.ibm.com:
ESP(spi=0x3fd0dc47,seq=0x2359), length 88
12:43:59.668995 IP wap-master.fyre.ibm.com > wap-worker-2.fyre.ibm.com:
ESP(spi=0x3fd0dc47,seq=0x235a), length 920
12:43:59.669203 IP wap-master.fyre.ibm.com > wap-worker-2.fyre.ibm.com:
ESP(spi=0x3fd0dc47,seq=0x235b), length 920
```

Os pacotes `ESP` são pacotes IPsec criptografados e indicam que a comunicação de rede está criptografada corretamente com o IPsec.

Verifique a criptografia TLS

Em cada componente do IBM Cloud Private, verifique se o modo FIPS está ativado para a criptografia TLS do tráfego de rede para terminais externos.

- WebSphere Liberty Application Server (gerenciador de autenticação)
- Componentes de gerenciamento de imagem
- ingresso de gerenciamento
- Controlador de ingresso NGINX

WebSphere Liberty Application Server (gerenciador de autenticação)

Verifique se o WebSphere Liberty Application Server é executado com o FIPS ativado. Conclua as etapas a seguir:

1. Obtenha o nome do pod que está no contêiner `platform-auth-service`. Execute o comando a seguir:

   ```
   kubectl get po -n kube-system | grep auth-idp
   ```

   Sua saída pode ser semelhante ao texto a seguir:

   ```
   auth-idp-xpxjn                                    4/4       Running     10         1d
   ```

2. Para abrir um ambiente de shell para o pod, execute o comando a seguir:

   ```
   kubectl exec -it -n kube-system auth-idp-xpxjn -c platform-auth-service -- /bin/bash
   ```

3. Verifique se o FIPS está ativado no ambiente de contêiner `platform-auth-service`. Execute o comando a seguir:

   ```
   env | grep FIPS
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   FIPS_ENABLED=true
   ```

4. Verifique se o WebSphere Liberty Application Server é iniciado no modo FIPS. Execute o comando a seguir:

   ```
   ps -ef | grep java
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   10 root       9:50 /opt/ibm/java/jre/bin/java -javaagent:/opt/ibm/wlp/bin/tools/ws-
   javaagent.jar -Djava.awt.headless=true -Dcom.ibm.jsse2.usefipsprovider=true -jar
   /opt/ibm/wlp/bin/tools/ws-server.jar defaultServer
   ```

5. Saia do shell que você abriu para o contêiner platform-auth-service.

Componentes de gerenciamento de imagem

Conclua as etapas a seguir para verificar se o FIPS está ativado para os componentes de gerenciamento de imagem:

1. Execute o comando a seguir para visualizar os logs `image-manager`:

   ```
   kubectl logs -n kube-system image-manager-0 -c image-manager | grep FIPS
   ```

A saída pode ser semelhante ao conteúdo a seguir:

```
 time="2018-11-01T16:22:14Z" level=info msg="handler.APIHandler.ServeCmd (serve.go:99) OpenSSL
FIPS mode is set to: True\n "
```

2. Visualize os logs `icp-registry`. Execute o comando a seguir:

```
 kubectl logs image-manager-0 -n kube-system -c icp-registry | grep FIPS
```

Sua saída pode ser semelhante ao texto a seguir:

```
 2018/11/01 19:33:43 OpenSSL FIPS mode is set to: True.
```

Gerenciamento de ingresso

Verifique se o FIPS está ativado no componente de ingresso de gerenciamento. Conclua as etapas a seguir:

1. Obtenha o nome do pod que está no componente de ingresso de gerenciamento. Execute o comando a seguir:

```
 kubectl get po -n kube-system | grep icp-management-ingress
```

A saída pode ser semelhante ao conteúdo a seguir:

```
 icp-management-ingress-h7rzq                                      1/1       Running
0          53m
```

2. Verifique se o ingresso de gerenciamento está em execução no modo FIPS. Execute o comando a seguir:

```
 kubectl logs -n kube-system icp-management-ingress-h7rzq | grep FIPS
```

A saída pode ser semelhante ao conteúdo a seguir:

```
 2018/11/14 21:09:22 [notice] 24#24: FIPS_mode_set() successfully (SSL:)
```

Controlador de ingresso NGINX

Verifique se o FIPS está ativado no controlador de ingresso NGINX. Conclua as etapas a seguir:

1. Obtenha o nome do pod que está no controlador de ingresso NGINX. Execute o comando a seguir:

```
 kubectl get po -n kube-system | grep nginx-ingress-controller
```

A saída pode ser semelhante ao conteúdo a seguir:

```
 nginx-ingress-controller-tg8zd                                    1/1       Running     0
58m
```

2. Para verificar se o controlador de ingresso NGINX está em execução no modo FIPS, execute o comando a seguir:

```
 kubectl logs -n kube-system nginx-ingress-controller-tg8zd | grep FIPS
```

A saída pode ser semelhante ao conteúdo a seguir:

```
 2018/11/14 21:07:19 [notice] 36#36: FIPS_mode_set() successfully (SSL:)
```

Você verificou que seu cluster do IBM Cloud Private está ativado pelo FIPS.

Acessando seu cluster

Saiba como acessar seu cluster.

- Acessando o cluster do IBM® Cloud Private usando a console de gerenciamento
- Gerenciando seu cluster a partir da console de gerenciamento com o terminal da web
- Gerenciando rótulos de cluster
- Acessando o cluster do IBM Cloud Private usando a CLI kubectl

Acessando o cluster do IBM® Cloud Private usando a console de gerenciamento

É possível acessar a console de gerenciamento de seu cluster do IBM Cloud Private por meio de um navegador da web suportado.

Pré-requisitos

Deve-se instalar o IBM Cloud Private. Visualize Instalando o IBM Cloud Private para obter mais informações.

1. Configure o IBM Multicloud Manager. Visualize as Opções de configuração durante a instalação para obter mais tópicos.
2. Conecte-se à URL `https://<Cluster Master Host>:<Cluster Master API Port>` ou `https://<Cluster Master Host>:<Cluster Master API Port>/multicloud` e use as credenciais apropriadas.

   Nota: o `<Cluster Master Host>` e o `<Cluster Master API Port>` estão definidos em Terminais principais.
3. Acesse seu cluster a partir de um navegador da web. Para obter uma lista de navegadores suportados, consulte Navegadores suportados.

Acessando o console de gerenciamento

1. Efetue login em seu cluster com suas credenciais.
2. Visualize o menu de navegação e o cabeçalho para as capacidades.

Componentes console

- Página Introdução
- Página de Procura
- Página de Visão Geral
- Página Topologia
- Página Políticas

Página Introdução

No IBM Cloud Private e no IBM Multicloud Manager, é possível visualizar descrições e tarefas, bem como instalar várias ferramentas da CLI.

Página de Procura

Nota: a procura não é suportada para os usuários do IBM Z.

No IBM Cloud Private e no IBM Multicloud Manager, é possível procurar por recursos do Kubernetes em qualquer cluster e filtrar sua procura pelos campos de recurso. Os resultados da procura são baseados fora de seus objetos de cluster.

Use a barra de procura para procurar por itens em seu cluster. Por exemplo, é possível filtrar sua procura pelas categorias a seguir. As opções de procura dependem de seus objetos de cluster. Consulte os exemplos a seguir:

- amável
- name
- namespace
- status
- memory
- cpu
- created (o quão recentemente o objeto foi criado)
- cluster (o cluster no qual o objeto se encontra)

Só é possível procurar por recursos com base em seu nível de acesso. Para obter mais informações sobre o acesso da função, consulte Controle de acesso baseado na função para IBM Multicloud Manager. Se você deseja salvar sua procura, clique no ícone **Salvar disco**. Clique na guia **Nova procura** para iniciar uma nova procura.

Página de Visão Geral

Use a página Visão geral para gerenciar e reorganizar o painel de suas informações de cluster. É possível visualizar detalhes de seus clusters do IBM Cloud Private e de outros provedores de serviço de nuvem que o IBM Cloud Private suporta. Também é possível visualizar detalhes sobre seus aplicativos. O painel Visão Geral é atualizado continuamente em tempo real.

Reorganizando seu painel

É possível personalizar sua visualização do painel Visão Geral reorganizando os cartões de visão geral do recurso. É possível visualizar as informações a seguir sobre seus clusters:

- Nome do serviço de nuvem com o número de clusters
- Conformidade de
- Detalhes do pod
- Status do Pod
- Recursos de cluster (uso de VCPU/Memory)
- Uso de armazenamento

Visualizando Detalhes do Aplicativo

É possível visualizar as informações de cada aplicativo. Clique em **Expandir detalhes** para visualizar as seguintes informações:

- Número de clusters
- Número de tipos do Kubernetes
- Número de regiões
- Número de nós
- Número de pods

Visualizando seu funcionamento do pod

Visualize o funcionamento do pod de todos os seus clusters expandindo o Mapa de Calor. O Mapa de Calor exibe as caixas coordenadas por cor que representam o limite de uso de VCPU de seus nós.

Clique em **Expandir detalhes** para visualizar o mapa. O tamanho das caixas com coordenadas coloridas representa a quantia de nós em seu cluster. Passe o mouse sobre a caixa para visualizar o tempo de resposta de seu cluster.

Filtrando seus resultados

É possível personalizar a página Visão Geral ainda mais com o recurso de filtragem. Clique em **Filtrar resultados** para especificar quais informações são exibidas em sua página.

Página de topologia

A visualização Topologia permite que você visualize a comunicação entre as dependências dentro de um cluster e a comunicação entre os próprios clusters. Para reduzir os gráficos na página, é possível filtrar a visualização por Clusters, Namespaces, Tipos e Rótulos. Também é possível filtrar o design selecionando o ícone que representa os controladores do Kubernetes.

Página Políticas

Use a página Políticas para criar e gerenciar políticas de segurança que são definidas para controles de segurança que são ativados. Consulte a seguinte lista de recursos para o painel Políticas:

- Visualize as violações de política de segurança e de cluster que existem.
- Corrija suas violações de segurança.
- Execute uma análise de causa raiz de violações de segurança.
- Visualize os controles de segurança para cada controle de certificação de conformidade.
- Configure os controles de segurança.
- Forneça interfaces para inserir dados em outras ferramentas.
- Visualize um resumo de relatórios de risco para cada controle de segurança.
- Visualize relatórios periódicos de disponibilidade de segurança.

Importante: Você deve ativar e gerenciar todos os controles de segurança e padrões para a segurança interna corporativa e padrões de conformidade regulamentares externos para as implementações do IBM Cloud Private.

Customize a página Políticas filtrando os padrões, categorias, controles e tipo de correção.

Na guia *Visão geral* na página *Políticas*, é possível visualizar as *Principais violações*, que são políticas de cluster com a maioria das violações. É possível filtrar o painel *Principais violações* pelos rótulos *Clusters* ou *Políticas*. Também é possível filtrar o painel *Visão geral de política* por *Categorias* ou *Padrões*.

Na guia *Todas as políticas*, é possível visualizar uma tabela de suas políticas. As seguintes informações sobre sua política estão disponíveis na tabela:

- Namespace
- Correção
- Compatível com o Cluster
- Controles
- Padrões
- Categorias

Selecione uma política para visualizar os seguintes detalhes da política:

- Detalhes de Critérios
- Modelo YAML de Política
- Status da Política
- Políticas
- Políticas de Colocação
- Ligações de Placement

Consulte a documentação a seguir para saber como gerenciar o painel Políticas:

- Gerenciando uma política de segurança

Gerenciando seu cluster a partir do console de gerenciamento com o terminal da web

O IBM Cloud Private inclui um terminal da web que é executado sequencialmente com a console de gerenciamento. É possível comunicar-se com o cluster sem fazer download e configurar ferramentas de CLI da Internet.

O terminal da web autentica automaticamente o usuário para efetuar login no cluster e configura as ferramentas para o usuário atual. O terminal da web é um shell restrito e possui uma quantidade limitada de ferramentas de shell.

Importante: o `kubectl`, o `helm` e o `cloudctl` são as únicas ferramentas de CLI disponíveis no terminal da web.

Para gerenciar seu cluster com o terminal da web, conclua as etapas a seguir:

1. Selecione um tópico no qual o ícone de terminal da web será exibido.
2. Clique no ícone do terminal da web para exibir o dock do terminal.

Quando não há dados enviados para o terminal da web depois que ele é aberto, sua sessão não é válida.

No cabeçalho do dock do terminal, há ícones para ajuda, fechamento do terminal e reposicionamento do terminal.

- O ícone de ajuda exibe links para a documentação da ferramenta CLI instalada.
- O ícone de fechamento do terminal fecha o terminal.
- O ícone de reposicionamento exibe as opções a seguir para mover o terminal da web: Esquerda, Inferior, Direita.

  Atenção: se você navegar para outra guia na qual o ícone de terminal da web não aparece, o terminal da web será fechado automaticamente e o conteúdo no terminal será perdido.

Se a conexão de seu contêiner em execução para o terminal da web falhar, consulte A conexão falha no terminal da web.

Gerenciando rótulos de cluster

Inclua um rótulo em seu cluster para selecionar os recursos do grupo. Consulte Rótulos e seletores para obter mais informações.

É possível incluir novos rótulos, remover rótulos existentes e editar rótulos existentes para seus clusters. Conclua as etapas a seguir para gerenciar seus rótulos:

1. No menu de navegação, clique em **Clusters**.

2. Clique no ícone **Abrir e fechar lista de opções**.
3. Clique em **Editar rótulos**.
4. Na caixa de diálogo Rótulos do Gerenciador, designe um valor ao seu rótulo. Depois de designar um valor, seu rótulo pode ser semelhante à designação a seguir:

   ```
   environment=Dev
   ```

5. Clique em **Salvar**.

   Seus rótulos criados aparecem na coluna Rótulos.

- Se você deseja remover um rótulo existente, selecione um rótulo e clique no ícone **Remover um item**.
- Se você deseja atualizar um rótulo existente, selecione e edite o rótulo. Depois de editar seu rótulo, clique no ícone **Incluir um novo item**.

  Seu rótulo foi atualizado.

Acessando seu cluster a partir da CLI do Kubernetes (kubectl)

Estas informações foram movidas para o guia da CLI na documentação do produto. Consulte Instalando a CLI do Kubernetes (kubectl) para obter instruções.

Consulte o Guia de ferramentas da CLI para obter mais opções de ferramenta da CLI.

Guia do Operador

Este guia contém as tarefas diárias para gerenciamento de sua plataforma IBM® Cloud Private. Como operador, é possível gerenciar seu cluster e usar a medição, o monitoramento e a criação de log para melhorar sua plataforma. Também é possível gerenciar a segurança, a rede, os nós, o armazenamento, os arquivos de configuração e as políticas.

- Administração de cluster e de plataforma
- Guia de segurança
- Guia de rede
- Guia de armazenamento
- Medição, monitoramento e criação de log

Administração de cluster e da plataforma

Aprenda a manter seu cluster e plataforma.

Nota: quando as mudanças na configuração de seu cluster não são feitas com o seguinte comando do instalador: `sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee xxx`, suas mudanças não serão aplicadas após o upgrade ou a substituição do certificado.

- Corrigindo seu cluster
- Reiniciando seu cluster
- Manutenção do nó
- Desinstalando o
- Fazendo upgrade do
- Revertendo
- Configurando o TLS e conjuntos de cifras para o gerenciador de imagem e o registro do IBM Cloud Private
- Ativando e desativando componentes do IBM Cloud Private
- Gerenciando clusters etcd
- Criando ConfigMaps
- Configurando uma mensagem de notificação de uso do sistema
- Incluindo ou removendo nós do cluster
- Recursos
- Gerenciando políticas
- Gerenciando segredos

- Reconfigurando o Kubelet em um cluster em tempo real

Corrigindo seu cluster

Depois de instalar o IBM Cloud Private, é possível verificar o IBM® Fix Central para ver se as correções que precisam ser aplicadas em seu cluster estão disponíveis.

O IBM® Fix Central contém correções e atualizações para o produto.

Consulte Minhas Notificações para obter atualizações por meio de e-mail ou feed RSS sobre os boletins de segurança mais recentes, problemas conhecidos e fix packs que estão sendo liberados.

Reiniciando seu cluster

Como parar o nó do cluster do IBM® Cloud Private:

Importante: antes de parar os serviços `kubelet` e `docker` no nó, marque o nó como não planejável. Execute o comando a seguir:

```
kubectl cordon 9.111.255.122
```

Nota: marcar o nó como não planejável desativa o planejamento de novos pods no nó.

1. Encerre o sistema parando o kubelet no nó de destino executando o comando a seguir:

   ```
   sudo sistemctl pare kubelet
   ```

2. Pare os contêineres de docker ou o tempo de execução do docker executando o comando a seguir:

   ```
   sudo systemctl parar docker
   ```

Como iniciar o IBM Cloud Private nó do cluster

1. Reinicie o Docker executando o comando a seguir:

   ```
   sudo systemctl start docker
   ```

2. Reinicie o kubelet e assegure-se de que ele esteja em execução com sucesso por meio do comando a seguir:

   ```
   sudo systemctl start kubelet
   sudo systemctl status kubelet
   ```

3. Se o serviço kubelet for mal sucedido, visualize os logs para o kubelet executando o comando a seguir:

   ```
   sudo journalctl -e -u kubelet
   ```

4. Saia da manutenção executando o comando a seguir:

   ```
   kubectl uncordon 9.111.255.122
   ```

Manutenção do Nó

Para executar a manutenção em um nó, é possível cancelar o planejamento e drenar um nó.

1. Marque o nó como não planejável executando o comando a seguir:

   ```
   kubectl cordon 9.111.255.122
   ```

   Nota: marcar o nó como não planejável desativa o planejamento de novos pods no nó.

2. Drene o nó em preparação para manutenção para remover os pods que estão em execução no nó executando o comando a seguir:

   ```
   kubectl de dreno 9.111.255.122 -- grace-period=300 -- ignore-daemonsets=true
   ```

Para obter informações adicionais sobre como drenar o nó, consulte a ajuda para o comando `kubectl drain` inserindo: `kubectl help drain`. Como o nó já estava marcado como não planejável, os pods das implementações `ReplicationController`, `ReplicaSet`, `Job` e `StatefulSet` não são planejados para esse nó. O planejador move todas as cargas de trabalho para outro nó que é planejável.

Nota: a drenagem do nó a partir da implementação `DaemonSet` não é necessária.

3. Saia da manutenção executando o comando a seguir:

```
kubectl uncordon 9.111.255.122
```

Nota: o recurso alfa `TaintNodesByCondition` para o kube-controller-manager em todos os nós principais é ativado por padrão.

Fazendo backup do ambiente do

IBM Cloud Private

É importante fazer backup de seu ambiente do IBM® Cloud Private para agilizar e facilitar a recuperação de um desastre. O procedimento para backup pode diferir de acordo com o componente.

Alguns ou todos os seus dados podem ser perdidos de seu ambiente do IBM Cloud Private quando ocorre um desastre. Isso pode incluir informações de criação de log, informações de monitoramento e informações de configuração.

Faça backup de suas informações de configuração imediatamente após a instalação usando a ferramenta de software de backup da VM, que fornece uma imagem inicial que pode ser usada como uma origem para arquivos, caso precise recriar seu ambiente. Esse backup inicial é o único que deve ser feito com a ferramenta VM, a menos que você tenha necessidades específicas para criar as imagens de backup completo com mais frequência.

Em muitos casos, a criação de backups periódicos de componentes selecionados fornece as capturas instantâneas necessárias dos dados naquele momento. Os componentes dentro dos nós principais requerem os procedimentos de backup adicionais. Esses backups periódicos asseguram que o backup do conteúdo essencial seja mantido atual, de acordo com seus requisitos. A frequência de seu backup de dados determina a quantia de dados que será perdida durante um problema de rede. Backups mais frequentes significam que há menos tempo entre as capturas para que os dados sejam coletados e, potencialmente, perdidos.

Como os nós de gerenciamento, os nós do proxy e os nós do trabalhador do cluster são facilmente recriados, eles não precisam ser atualizados após a instalação inicial. **Importante:** como os dados históricos nesses nós, como informações de criação de log e de monitoramento, são perdidos após um desastre de rede, novos nós precisam ser implementados. Se você tiver um requisito histórico para reter esses dados, inclua esses nós em seu plano de backup.

Fazendo backup do nó Principal

O nó Principal é geralmente o único nó do qual é necessário fazer backup periodicamente. O nó Principal contém o componente `etcd`, que não pode usar a ferramenta VM que é fornecida com o hypervisor para copiá-lo.

Componente Etcd

Para fazer backup do componente `etcd`, conclua as etapas a seguir em um nó Principal único:

1. Efetue logon no nó Principal com a função *Administrador de Cluster* ou superior.
2. Execute os comandos a seguir para exportar as variáveis de ambiente necessárias:

```
export org=ibmcom
export repo=etcd
export tag=v3.2.14
export endpoint=etcd_member_IP
```

Substitua *etcd_member_IP* pelo endereço IP de um de seus membros `etcd`.

3. Copie o arquivo *etcdctl* para */user/local/bin/* inserindo o comando a seguir:

```
sudo docker run --rm -v /usr/local/bin:/data
$org/$repo:$tag cp /usr/local/bin/etcdctl /data
```

4. Execute o script `etcd.sh` inserindo o comando a seguir:

```
. ./etcd.sh
```

5. Valide o status do cluster etcd executando os comandos a seguir:

```
etcdctl2 cluster-health
etcdctl2 member list
```

Os resultados retornados devem ser semelhantes ao conteúdo a seguir:

```
# etcdctl2 cluster-health
member 7a5703380976f596 is healthy: got healthy result from https://192.0.2.0:24
member 7c2ce9ea4a75caaa is healthy: got healthy result from https://192.0.2.1:24
member fd529306e0ed0813 is healthy: got healthy result from https://192.0.2.2:24
cluster is healthy

# etcdctl2 member list
7a5703380976f596: name=etcd1 peerURLs=https://192.0.2.0:2380 clientURLs=https://192.0.2.0:4001
isLeader=false
7c2ce9ea4a75caaa: name=etcd2 peerURLs=https://192.0.2.1:2380
clientURLs=https://9.111.255.178:4001 isLeader=true
fd529306e0ed0813: name=etcd0 peerURLs=https://192.0.2.2:2380 clientURLs=https://192.0.2.2:4001
isLeader=false
```

6. Faça uma captura instantânea dos dados etcd inserindo o comando a seguir:

```
etcdctl3 snapshot save /data/etcd.db
```

O arquivo de dados etcd está disponível no diretório `/data` no nó principal. É possível mudar o diretório para armazená-lo em outro local.

Componente de registro do Docker

Conclua as etapas a seguir para criar um backup periódico de seu registro do Docker:

1. Inclua uma imagem no registro do Docker do IBM Cloud Private:

a. Se você ainda não tiver configurado a autenticação para a CLI do Docker, conclua o procedimento em Configurando a autenticação para a CLI do Docker.

Nota: você deve estar apto para executar os comandos em qualquer sistema que tenha acesso ao nó principal do IBM Cloud Private e que tenha um mecanismo do Docker instalado.

b. Puxe uma imagem `nginx` inserindo o comando a seguir:

```
docker pull nginx
```

Deverá ser exibida uma saída semelhante ao seguinte texto:

```
patro:icp-backup edu$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8176e34d5d92: Pull complete
5b19c1bdd74b: Pull complete
4e9f6296fa34: Pull complete
Digest: sha256:4771d09578c7c6a65299e110b3ee1c0a2592f5ea2618d23e4ffe7a4cab1ce5de
Status: Downloaded newer image for nginx:latest
```

c. Efetue login no seu registro do Docker inserindo o comando a seguir:

```
docker login mycluster.icp:8500
```

d. Digite seu ID do usuário e sua senha.

e. Identifique a imagem executando o comando a seguir:

```
docker tag nginx mycluster.icp:8500/default/nginx
```

f. Envie a imagem por push para o registro do Docker executando o comando a seguir:

```
docker push mycluster.icp:8500/default/nginx
```

A saída retornada deve ser semelhante ao texto a seguir:

```
patro:.docker edu$ docker push mycluster.icp:8500/default/nginx
The push refers to repository [mycluster.icp:8500/default/nginx]
```

```
e89b70d28795: Pushed
832a3ae4ac84: Pushed
014cf8bfcb2d: Pushed
latest: digest: sha256:600bff7fb36d7992512f8c07abd50aac08db8f17c94e3c83e47d53435a1a6f7c size:
948
```

g. Abra seu navegador para o link a seguir para verificar se a imagem nginx está listada:

```
https://$MASTER_ID:8443/console/images
```

2. Faça backup do registro do Docker:

a. Mude para o diretório no qual o registro está armazenado inserindo o comando a seguir:

```
cd /var/lib/registry
```

b. Crie um arquivo `tar` com a imagem de backup inserindo o comando a seguir:

```
tar czvf /tmp/icp_dr.tar.gz .
```

c. Mova o arquivo `/tmp/icp_dr.tar.gz` para um local que esteja fora do nó principal para armazenamento.

Nota: se você automatizar esse procedimento, mova esse arquivo para um local que esteja em seu armazenamento compartilhado ou em sua rotina de backup no host.

Componente MongoDB

O armazenamento de dados do IBM MongoDB é usado pelo IBM Cloud Private para armazenar informações para o serviço OIDC, para o serviço de medição (IBM Cloud Product Insights), para o servidor do repositório Helm, para o servidor de API do Helm, etc. Ele é executado como um conjunto **icp-mongodb** stateful do Kubernetes nos Nós Principais. Se você inspecionar seu cluster, veja que os pods nesse conjunto stateful denominado **icp-mongodb-(increment)** são executados um por principal e que o armazenamento é montado no caminho do host local. O conjunto stateful é exposto como um serviço como `mongodb`.

Conclua o procedimento a seguir para fazer backup do MongoDB:

1. Carregue dados no MongoDB de amostra, se necessário.

a. Execute o comando a seguir para conectar-se ao serviço `mongodb`:

```
kubectl exec -n kube-system -it icp-mongodb-0 -- sh -c 'mongo --host rs0/mongodb:27017 --
username $ADMIN_USER --password $ADMIN_PASSWORD --authenticationDatabase admin --ssl --
sslCAFile /data/configdb/tls.crt --sslPEMKeyFile /work-dir/mongo.pem'
```

Faça as substituições necessárias para *$ADMIN_USER* e *ADMIN_PASSWORD*.

b. Abra a CLI do MongoDB.

c. No prompt da CLI, insira os comandos a seguir para carregar alguns dados de amostra:

```
db.myCollection.insertOne({ key1: "value1" });
db.myCollection.insertOne({ key2: "value2" });
```

d. Execute o comando a seguir para recuperar os valores:

```
db.myCollection.find()
```

2. Faça backup do MongoDB para o sistema de arquivos local do nó principal usando a ferramenta *mongodump*. É possível efetuar dump dos dados de backup para um sistema de arquivos local do nó principal ou para um volume persistente. Se você deseja fazer backup para um volume persistente, vá para a etapa 3.

a. Crie um backup das informações no sistema de arquivos local inserindo o comando a seguir:

```
kubectl -n kube-system exec icp-mongodb-0 -- sh -c 'mkdir -p /work-dir/Backup/mongodump;
mongodump --oplog --out /work-dir/Backup/mongodump --host rs0/mongodb:27017 --username
$ADMIN_USER --password $ADMIN_PASSWORD --authenticationDatabase admin --ssl --sslCAFile
/data/configdb/tls.crt --sslPEMKeyFile /work-dir/mongo.pem'
```

A execução desse comando cria um dump no diretório `/var/lib/icp/mongodb/work-dir/backup/mongodump`.

b. Crie um archive do diretório com um registro de data e hora e mova-o para um local que não esteja no nó principal.

3. Faça backup do MongoDB para um PersistentVolume (PV) usando a ferramenta *mongodump*. É possível efetuar dump dos dados de backup para um PV ou para um sistema de arquivos local do nó principal. Se você deseja fazer backup para um sistema de arquivos local do nó principal, consulte a etapa 2.

   Nota: os arquivos *mongodump-pv.yaml*, *mongodump-pvc.yaml*, *icp-mongodb-mongodump-job.yaml* e *icp-mongodb-mongorestore-job.yaml* podem ser localizados em `icp-backup/resources` no Repositório Git ibm-cloud-architecture/icp-backup.

   a. Crie um PV inserindo o comando a seguir:

   ```
   kubectl apply -f mongodump-pv.yaml
   ```

   Consulte o tópico do Kubernetes intitulado Volumes Persistentes para obter mais instruções, dependendo do tipo de PV que você está criando. **Nota:** para este exemplo, já criamos um diretório NFS e incluímos o IP e o diretório do servidor NFS no `mongodump-pv.yaml`.

   b. Crie um PersistentVolumeClaim (PVC), que nossas tarefas possam usar para obter acesso ao PV executando o comando a seguir:

   ```
   kubectl apply -f mongodump-pvc.yaml
   ```

   c. Execute o comando a seguir para efetuar dump do banco de dados MongoDB:

   ```
   kubectl apply -f icp-mongodb-mongodump-job.yaml
   ```

   Esta tarefa do Kubernetes efetua dump dos bancos de dados MongoDB no PV que você criou. Se este for seu backup de cluster do IBM Cloud Private, certifique-se de que este PV esteja protegido, submetido a backup e salvo. Este conteúdo é necessário para restaurar a instância.

Restaurando seu ambiente do IBM Cloud Private

Um backup completo inicial na instalação e backups periódicos de componentes individuais permitem restaurar seu ambiente do IBM® Cloud Private quando ocorre um desastre.

Se precisar recuperar seu ambiente do IBM Cloud Private após um desastre, os dados submetidos a backup serão necessários para os componentes selecionados em seus nós principais. Para obter instruções sobre como capturar os backups, consulte Fazendo backup de seu ambiente do IBM Cloud Private.

Os nós de gerenciamento, os nós de proxy e os nós do trabalhador do cluster são recriados automaticamente quando os nós Principais são restaurados. **Importante:** como os dados históricos nesses nós, como informações de criação de log e de monitoramento, são perdidos após um desastre, novos nós precisam ser implementados. Se for necessário reter esses dados históricos, inclua esses nós no seu plano de backup.

Restaurando o nó principal

O nó Principal é geralmente o único nó que é submetido a backup periodicamente em preparação para um possível desastre. Os componentes a seguir devem ser restaurados para configurar o nó Principal e fornecem as informações que são necessárias para criar os outros nós que estavam no ambiente original.

Supõe-se que você esteja utilizando o processo de restauração para uma das seguintes razões:

- Você está recuperando um Nó Principal para um cluster de Nó Principal único.
- Você está recuperando um Nó Principal em um ambiente de múltiplos Nós Principais que requer que o estado inicial do etcd seja restaurado manualmente (para acomodar sua estratégia e metodologia da ferramenta de backup).

Restaurar o banco de dados MongoDB

Restaurar backup a partir do sistema de arquivos local

Nas instruções de backup, você empacotou o banco de dados MongoDB no diretório /var/lib/icp/mongodb/work-dir/backup/mongodump e o arquivou em um local diferente. Para restaurá-lo, conclua as etapas a seguir:

1. Mova o archive de volta para o diretório `/var/lib/icp/mongodb/work-dir/backup/mongodump`.
2. Descompacte o arquivo.

3. Execute o comando `mongorestore`.

4. Execute o comando a seguir para restaurar dados salvos no sistema de arquivos do nó Principal:

```
kubectl -n kube-system exec icp-mongodb-0 -- sh -c 'mongorestore --host rs0/mongodb:27017 --
username $ADMIN_USER --password $ADMIN_PASSWORD --authenticationDatabase admin --ssl --
sslCAFile /data/configdb/tls.crt --sslPEMKeyFile /work-dir/mongo.pem /work-
dir/Backup/mongodump'
```

Restaurar backup a partir de um PersistentVolume

Execute o comando a seguir para restaurar dados salvos em um PersistentVolume:

```
kubectl apply -f icp-mongodb-mongorestore-job.yaml
```

Valide os dados que foram restaurados

1. **Dentro** do pod da CLI do MongoDB, execute o comando a seguir:

```
kubectl exec -n kube-system -it icp-mongodb-0 -- sh -c 'mongo --host rs0/mongodb:27017 --
username $ADMIN_USER --password $ADMIN_PASSWORD --authenticationDatabase admin --ssl --
sslCAFile /data/configdb/tls.crt --sslPEMKeyFile /work-dir/mongo.pem'
```

2. Execute o comando a seguir para localizar os pares chave-valor:

```
db.myCollection.find()
```

Os pares chave-valor devem ser exibidos.

Restaurando o registro do Docker

Para restaurar o registro do Docker, conclua as seguintes etapas:

1. Copie o arquivo de backup de seu registro do Docker, `/tmp/icp_dr.tar.gz` para o diretório `/tmp`.

2. Mude para o diretório de registro:

```
cd /var/lib/registry
```

3. Extraia o arquivo de backup para o diretório correto:

```
tar xvzf /tmp/icp_dr.tar.gz
```

4. Execute o comando a seguir para reciclar o pod do gerenciador de imagem:

```
kubectl delete pod image-manager-0 -n kube-system
```

5. Reabra `https://$MASTER_ID:8443/console/images`. As imagens são restauradas.

Restaurando o componente etcd - nó Principal único

Para restaurar seu armazenamento de dados etcd para um dos nós Principais, conclua as etapas a seguir:

Nota: essas instruções não incluem o processo para restaurar o etcd para topologias que tenham externalizado o cluster etcd. No entanto, é possível criar os processos de backup e de restauração com base nas ferramentas e etapas a seguir.

1. Pare o Pod etcd inserindo o comando a seguir:

```
mkdir -p /etc/cfc/podbackup
```

2. Mova o arquivo `etcd.json` para um diretório de backup inserindo o comando a seguir:

```
mv /etc/cfc/pods/etcd.json /etc/cfc/podbackup/
```

3. Verifique se o pod foi interrompido executando o comando a seguir:

```
docker ps | grep etc
```

Se o pod estiver interrompido, nenhuma resposta será retornada.

4. Limpe os dados etcd executando o comando a seguir:

   ```
   rm -rf /var/lib/etcd
   ```

5. Localize o último arquivo de backup para o cluster, que deve ser nomeado como algo semelhante a: `/tmp/etcd.the-date-and-time.db`. Substitua *your-date-and-time* pelo nome de seu arquivo de backup.

6. Execute o comando a seguir para restaurar os dados do `etcd`:

   ```
   ./restoreEtcd.sh etcd.your-date-and-time.db
   ```

   Deverá ser exibida uma resposta semelhante ao seguinte texto:

   ```
   root@eduardo-icp:~/icp-backup/scripts# ./restoreEtcd.sh etcd.your-date-and-time.db
   Restore snapshot etcd.your-date-and-time.db
   your-date-and-time I | mvcc: restore compact to **your size value here**
   your-date-and-time I | etcdserver/membership: added member **the ID for the member**
   [https://169.61.93.24:2380] to cluster **your cluster id**
   ```

   Os dados estão no diretório `/var/lib/etcd/restored`.

7. Mova os dados para o diretório necessário executando o comando a seguir:

   ```
   mv /var/lib/etcd/restored/* /var/lib/etcd/
   ```

8. Exclua o diretório a partir do qual você moveu o arquivo.

   ```
   rmdir /var/lib/etcd/restored
   ```

9. Ative o pod etcd executando o comando a seguir:

   ```
   mv /etc/cfc/podbackup/etcd.json /etc/cfc/pods/
   ```

   Dependendo de seu ambiente, pode levar alguns minutos para que o componente etcd seja iniciado. É possível ver o progresso executando o comando a seguir:

   ```
   docker ps | grep
   ```

   Deverá ser exibida uma resposta semelhante ao seguinte texto:

   ```
    root@icp-master:~# docker ps | grep etcd
    999c8e48c0e3        ibmcom/etcd                         "etcd --name=etcd0 -…"   About a minute
   ago   Up About a minute                       k8s_etcd_k8s-etcd-10.0.0.1_kube-
   system_349da84ef01d46f51daacdd97b2991e1_0
    747287ff5b4f        ibmcom/pause:3.0                    "/pause"                 About a minute
   ago   Up About a minute                       k8s_POD_k8s-etcd-10.0.0.1_kube-
   system_349da84ef01d46f51daacdd97b2991e1_0
   ```

10. Valide se o novo ambiente possui os dados restaurados no etcd executando o comando a seguir para exibir os ConfigMaps do Kubernetes:

    ```
    kubectl get configmaps | grep snake
    ```

    Se você carregou a amostra antes de iniciar este procedimento, será exibida uma saída semelhante ao seguinte conteúdo:

    ```
    root@icp-master:~# kubectl get configmaps | grep snake
    snake-10        1         48m
    snake-11        1         48m
    snake-12        1         48m
    snake-13        1         48m
    snake-14        1         48m
    snake-15        1         48m
    snake-16        1         48m
    snake-17        1         48m
    snake-18        1         48m
    snake-19        1         48m
    snake-20        1         48m
    snake-21        1         48m
    snake-22        1         48m
    snake-23        1         48m
    snake-24        1         48m
    snake-25        1         48m
    snake-26        1         48m
    ```

```
snake-8          1          48m
snake-9          1          48m
```

Se isso não foi feito, deverá ver que os ConfigMaps faziam parte de seu sistema no momento em que o backup foi feito.

Restaurando o componente etcd - múltiplos nós Principais

A restauração de um ambiente do IBM Cloud Private com múltiplos nós Principais pode ser feita usando um dos métodos a seguir:

- Restaure um nó Principal único e aumente o cluster para o tamanho necessário.
- Restaure o cluster inteiro a partir da imagem de infraestrutura de backup.

O procedimento a seguir aborda o método de restauração de cluster integral.

Pré-requisito

- O Red Hat Ansible está instalado no nó de inicialização. Este procedimento usa o Ansible para executar simultaneamente os comandos em todos os nós principais. As etapas podem ser concluídas sem ele, mas lembre-se de executar os comandos em todos os nós Principais. É possível verificar se o Ansible está instalado inserindo o comando a seguir no nó de inicialização:

  ```
  which ansible
  ```

  Se uma resposta vazia for retornada, o Ansible não está instalado no nó de inicialização. Para obter informações sobre como instalar o Ansible, consulte o website do Ansible.

- O processador JSON da linha de comandos jq é instalado em cada um dos nós principais. É possível verificar se você tem o jq instalado no Ubuntu executando o comando a seguir:

  ```
  ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
  key=$CLUSTER_DIR/ssh_key -m package -a "use=apt name=jq state=present"
  ```

Restaurando o cluster

Conclua as etapas a seguir para restaurar o ambiente com múltiplos nós Principais:

1. Pare o Kubernetes em todos os nós Principais. Isso para o pod etcd e evita que o Kubernetes crie automaticamente novos pods para os que estamos parando.

   a. Crie um diretório para o pod de backup inserindo o comando a seguir:

   ```
   ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
   key=$CLUSTER_DIR/ssh_key -a "mkdir -p /etc/cfc/podbackup"
   ```

   b. Mova o pod de backup para o diretório:

   ```
   ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
   key=$CLUSTER_DIR/ssh_key -m shell -a "mv /etc/cfc/pods/*.json /etc/cfc/podbackup"
   ```

   c. Aguarde o etcd parar em **todos** os nós. É possível verificar o status inserindo o comando a seguir:

   ```
   ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
   key=$CLUSTER_DIR/ssh_key -m wait_for -a  "port=4001 state=stopped"
   ```

   d. Depois de parar o etcd, pare o kubelet executando este comando em todos os nós Principais e nós de Gerenciamento:

   ```
   ansible master,management -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
   key=$CLUSTER_DIR/ssh_key -m service -a "name=kubelet state=stopped"
   ```

   e. Depois que o kubelet é interrompido, reinicie o serviço do Docker para assegurar que todos os pods que não são gerenciados pelo kubelet sejam interrompidos inserindo o comando a seguir:

   ```
   ansible master,management -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
   key=$CLUSTER_DIR/ssh_key -m service -a "name=docker state=restarted"
   ```

2. Limpar, copiar e restaurar os dados etcd.

   a. Limpe os dados etcd atuais em todos os Nós Principais executando o comando a seguir:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m shell -a "rm -rf /var/lib/etcd"
```

b. Copie a captura instantânea etcd para todos os nós principais. Supondo que você tenha o arquivo `/tmp/etcd.your-date-and-time.db` em seu ambiente, que contém um backup de seu etcd, execute o procedimento a seguir para copiar o arquivo para todos os nós Principais:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m copy -a "src=/tmp/etcd.your-date-and-time.db dest=/tmp/snapshot.db"
```

c. Restaure a captura instantânea em todos os nós Principais. Supondo que tenha clonado o repositório Git e que seu diretório atual seja `icp-backup/scripts`, execute o comando a seguir para executar o script que restaura a captura instantânea para todos os nós Principais:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m script -a "./multimaster-etcd-restore.sh"
```

Os dados são carregados no diretório `/var/lib/lib/etcd/restored` em cada um de seus nós principais, com as configurações de cluster definidas.

d. Mova o conteúdo para o diretório `/var/lib/etcd/` executando os comandos a seguir:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m shell -a "mv /var/lib/etcd/restored/* /var/lib/etcd/"
```

e. Execute o script a seguir para limpar o diretório de pods do kubelet para assegurar a consistência entre os dados kubelet e os dados etcd armazenados em cache:

```
 ansible master,management -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m script -a "./purge_kubelet_pods.sh"
```

f. Reative o pod `kubelet` inserindo o comando a seguir:

```
 ansible master,management -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m service -a "name=kubelet state=started"
```

g. Reative o pod `etcd` inserindo o comando a seguir:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m shell -a "mv /etc/cfc/podbackup/etcd.json /etc/cfc/pods"
```

h. Execute o comando a seguir para monitorar o progresso do status do componente etcd, conforme ele é iniciado:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m wait_for -a  "port=4001 state=started"
```

3. Validar o funcionamento do cluster etcd.

a. Execute o comando a seguir para configurar a ferramenta `etcdctl` para consultar o cluster etcd:

```
 export endpoint=<master-node-ip>
```

Mude o valor para **<master-node-ip>** para o endereço IP do nó Principal com o qual você está trabalhando.

b. Execute os scripts a seguir: `. ./etcd.sh`.

c. Consulte o funcionamento do cluster inserindo o comando a seguir:

```
 $etcdctl2 cluster-health
```

Deverá ser exibida uma resposta semelhante à saída a seguir:

```
 member 8211f1d0f64f3269 is healthy: got healthy result from https://10.0.0.1:2380
 member 91bc3c398fb3c146 is healthy: got healthy result from https://10.0.0.2:2380
 member fd422379fda50e48 is healthy: got healthy result from https://10.0.0.3:2380
 cluster is healthy
```

4. Inicie o restante dos pods de cluster do IBM Cloud Private inserindo o comando a seguir:

```
 ansible master -i $CLUSTER_DIR/hosts -e @$CLUSTER_DIR/config.yaml --private-
key=$CLUSTER_DIR/ssh_key -m shell -a "mv /etc/cfc/podbackup/*.json /etc/cfc/pods"
```

Esse comando permite que o kubelet inicie o restante dos pods do Kubernetes principais, que, em seguida, iniciam as cargas de trabalho que são gerenciadas pelo Kubernetes.

Leva vários minutos para que todos os pods sejam reiniciados. É possível monitorar os pods no namespace do `kube-system` executando o comando a seguir:

```
kubectl get pods --namespace=kube-system
```

Validando os resultados

Execute o comando a seguir para verificar se seu novo ambiente tem os dados restaurados no etcd:

```
kubectl get configmaps | grep snake
```

O comando exibe os ConfigMaps a partir do Kubernetes.

Se você carregou o conteúdo de amostra, será exibido um conteúdo semelhante ao texto a seguir:

```
root@icp-master:~# kubectl get configmaps | grep snake
snake-10        1         48m
snake-11        1         48m
snake-12        1         48m
snake-13        1         48m
snake-14        1         48m
snake-15        1         48m
snake-16        1         48m
snake-17        1         48m
snake-18        1         48m
snake-19        1         48m
snake-20        1         48m
snake-21        1         48m
snake-22        1         48m
snake-23        1         48m
snake-24        1         48m
snake-25        1         48m
snake-26        1         48m
snake-8         1         48m
snake-9         1         48m
```

Se você não usou o conteúdo de amostra, será necessário verificar se os ConfigMaps faziam parte de seu sistema quando o backup foi feito.

Desinstalação

É possível desinstalar o IBM® Cloud Private de seu cluster.

- Desinstalando o IBM Cloud Private
- Desinstalando o IBM Cloud Private-CE

Desinstalando o IBM Cloud Private

Desinstale o IBM Cloud Private.

1. Efetue login no nó de inicialização como um usuário com permissões raiz. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre tipos de nó, consulte Arquitetura. Durante a instalação, você especifica os endereços IP para cada tipo de nó.

2. Mude o diretório `cluster` dentro do seu diretório de instalação do IBM Cloud Private:

   ```
   cd /<installation_directory>/cluster
   ```

3. Desinstale o IBM Cloud Private.

   ```
   sudo docker run -e LICENSE=accept --net=host \
   -t -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed
  's/x86_64/amd64/g'):3.2.0-ee uninstall
   ```

Nota: se o processo de desinstalação for interrompido, reinicialize o nó e, em seguida, execute o comando de desinstalação.

4. Reinicie o Docker em cada nó em seu cluster. Execute o comando a seguir em cada nó:

   ```
   sudo systemctl restart docker
   ```

5. Reinicie todos os nós em seu cluster.

Nota: para concluir a desinstalação, reinicialize sua máquina.

Nota: se você tiver um cluster GlusterFS, a desinstalação do IBM Cloud Private excluirá os pods Heketi e GlusterFS. No entanto, os volumes lógicos não são removidos. É possível recuperar dados desses volumes lógicos. Para obter mais informações sobre como recuperar os dados, consulte Recuperando dados de um volume do GlusterFS. Se for necessário reutilizar os discos para uma reinstalação de GlusterFS, consulte Preparar os discos a serem usados para instalação do GlusterFS.

Desinstalando o IBM Cloud Private-CE

Desinstale o IBM Cloud Private-CE (Community Edition).

1. Efetue login no nó de inicialização como um usuário com permissões raiz. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre tipos de nó, consulte Arquitetura. Durante a instalação, você especifica os endereços IP para cada tipo de nó.

2. Mude para o diretório `cluster` dentro de seu diretório de instalação do IBM Cloud Private-CE.

   ```
   cd /<installation_directory>/cluster
   ```

3. Desinstale o IBM Cloud Private-CE.

   ```
   docker run -e LICENSE=accept --net=host \
     -t -v "$(pwd)":/installer/cluster \
     ibmcom/icp-inception:3.2.0 uninstall
   ```

   Nota: se o processo de desinstalação for interrompido, reinicialize o nó e, em seguida, execute o comando de desinstalação.

4. Reinicie o Docker em cada nó em seu cluster. Execute o comando a seguir em cada nó:

   ```
   service docker restart
   ```

5. Reinicie todos os nós em seu cluster.

Nota: se você tiver um cluster GlusterFS, a desinstalação do IBM Cloud Private-CE excluirá os pods Heketi e GlusterFS. No entanto, os volumes lógicos não são removidos. É possível recuperar dados desses volumes lógicos. Para obter mais informações sobre como recuperar os dados, consulte Recuperando dados de um volume do GlusterFS. Se for necessário reutilizar os discos para uma reinstalação de GlusterFS, consulte Preparar os discos a serem usados para instalação do GlusterFS.

Atualizando

Faça upgrade de seu cluster do IBM Cloud Private.

- Fazendo upgrade do IBM Cloud Private
- Fazendo upgrade do IBM Cloud Private-CE
- Fazendo upgrade de gráficos do Helm no Catalog
- Fazendo upgrade do pacote do Docker do IBM Cloud Private

Atualizando IBM Cloud Private

É possível fazer upgrade do IBM Cloud Private a partir de versões anteriores específicas.

Caminhos de Upgrade Suportados

É possível fazer upgrade apenas dos caminhos suportados a seguir:

- IBM Cloud Private versão 3.1.2 para 3.2.0

- IBM Cloud Private versão 3.1.1 para 3.2.0
- IBM Cloud Private versão 3.1.0 para 3.2.0

Se você usar uma versão anterior do IBM Cloud Private, deverá fazer upgrade primeiro para a versão 3.1.0.

Nota: assegure-se de revisar e verificar se você atende aos requisitos de memória aumentados. Para obter mais informações, consulte Requisitos de hardware .

Durante o processo de upgrade, não é possível acessar o IBM Cloud Private console de gerenciamento. Também não é possível configurar opções de provedor em nuvem, como configurar um provedor em nuvem vSphere, ou usar NSX-T.

Se sua instalação atual do IBM Cloud Private estiver configurada para usar o plug-in do Key Management Service (KMS) para criptografia, será possível preservar a configuração do KMS existente. Para obter mais informações, consulte Retendo a configuração do KMS durante o upgrade.

Atualizando

1. Efetue login no nó de inicialização como um usuário com permissões raiz. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre tipos de nó, consulte Arquitetura. Durante a instalação, você especifica os endereços IP para cada tipo de nó.
2. Faça download dos arquivos de instalação para o IBM Cloud Private. Esses arquivos estão disponíveis para download por meio do website IBM Passport Advantage®.
   - Para um cluster do Linux® x86_64, faça download do arquivo `ibm-cloud-private-x86_64-3.2.0.tar.gz`.
   - Para um cluster do Linux® on Power® (ppc64le), faça download do arquivo `ibm-cloud-private-ppc64le-3.2.0.tar.gz`.
   - Para um cluster do IBM® Z, faça download do arquivo `ibm-cloud-private-s390x-3.2.0.tar.gz`.
3. Extraia as imagens e carregue-as no Docker. Extrair as imagens pode levar alguns minutos.
   - Para o Linux x86_64, execute o seguinte comando:

     ```
     tar xf ibm-cloud-private-x86_64-3.2.0.tar.gz -O | sudo docker load
     ```

   - Para o Linux on Power (ppc64le), execute o seguinte comando:

     ```
     tar xf ibm-cloud-private-ppc64le-3.2.0.tar.gz -O | sudo docker load
     ```

   - Para um cluster do Linux on IBM Z and LinuxONE, execute o comando a seguir:

     ```
     tar xf ibm-cloud-private-s390x-3.2.0.tar.gz -O | sudo docker load
     ```

4. Crie um diretório de instalação e copie os diretórios do `cluster` do diretório de instalação anterior para a nova pasta do IBM Cloud Private `cluster`. Use um diretório de instalação diferente do que você usou para a versão anterior. Por exemplo, para armazenar os arquivos de configuração em `/opt/ibm-cloud-private-3.2.0`, execute os comandos a seguir:

   ```
    sudo mkdir -p /opt/ibm-cloud-private-3.2.0
   cd /opt/ibm-cloud-private-3.2.0
   sudo cp -r /<installation_directory>/cluster .
    sudo rm -rf cluster/.upgrade
   ```

   Nota: `/<installation_directory>` é o caminho completo para o diretório de instalação da versão 3.1.2 e `/<new_installation_directory>` é o caminho completo para o diretório de instalação da versão 3.2.0. Não é necessário copiar o pacote de instalação da imagem inteiro da versão anterior.

5. Verifique o valor de parâmetro `calico_ipip_enabled` na versão a partir da qual você está fazendo upgrade.
   - Se o parâmetro foi configurado como `calico_ipip_enabled: true`, substitua o parâmetro no `/<new_installation_directory>/cluster/config.yaml` por `calico_ipip_mode: Always`.
   - Se o parâmetro foi configurado como `calico_ipip_enabled: false`, substitua o parâmetro no `/<new_installation_directory>/cluster/config.yaml` por `calico_ipip_mode: Never`.
6. Mova os arquivos de imagem para seu cluster para a pasta `/<new_installation_directory>/cluster/images`.
   - Para o Linux x86_64, execute o seguinte comando:

     ```
     sudo mv /<path_to_images_file>/ibm-cloud-private-x86_64-3.2.0.tar.gz cluster/images/
     ```

- Para o Linux on Power (ppc64le), execute o seguinte comando:

  ```
  sudo mv /<path_to_images_file>/ibm-cloud-private-ppc64le-3.2.0.tar.gz cluster/images/
  ```

- Para um cluster do Linux on IBM Z and LinuxONE, execute o comando a seguir:

  ```
  sudo mv /<path_to_images_file>/ibm-cloud-private-s390x-3.2.0.tar.gz cluster/images/
  ```

  Nesse comando, `path_to_images_file` é o caminho para o arquivo de imagens.

7. Implemente seu ambiente concluindo as etapas a seguir:

   1. Mude a pasta `cluster` em seu diretório de instalação.

      ```
      cd / < new_installation_directory> /cluster
      ```

   2. Prepare o cluster para upgrade:

      - Para o Linux x86_64, execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-amd64:3.2.0-ee upgrade-prepare
        ```

      - Para o Linux on Power (ppc64le), execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-ppc64le:3.2.0-ee upgrade-prepare
        ```

      - Para o Linux on IBM Z and LinuxONE, execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-s390x:3.2.0-ee upgrade-prepare
        ```

      Se a preparação do cluster falhar, revise a mensagem de erro e resolva qualquer problema. Em seguida, execute o comando `upgrade-prepare` novamente.

   3. Faça upgrade do Kubernetes:

      - Para o Linux x86_64, execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-amd64:3.2.0-ee upgrade-k8s
        ```

      - Para o Linux on Power (ppc64le), execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-ppc64le:3.2.0-ee upgrade-k8s
        ```

      - Para o Linux on IBM Z and LinuxONE, execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-s390x:3.2.0-ee upgrade-k8s
        ```

      - Se o upgrade do Kubernetes falhar com uma mensagem diferente, revise a mensagem de erro e resolva possíveis problemas. Em seguida, execute novamente o comando de upgrade dos serviços do Kubernetes.

   4. Faça upgrade dos gráficos:

      - Para o Linux x86_64, execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ibmcom/icp-inception-amd64:3.2.0-ee upgrade-chart
        ```

      - Para o Linux on Power (ppc64le), execute o seguinte comando:

        ```
        sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
        \
        ```

```
ibmcom/icp-inception-ppc64le:3.2.0-ee upgrade-chart
```

- Para o Linux on IBM Z and LinuxONE, execute o seguinte comando:

```
sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster
\
ibmcom/icp-inception-s390x:3.2.0-ee upgrade-chart
```

- Se o upgrade do gráfico falhar com uma mensagem diferente, revise a mensagem de erro e resolva os problemas. Em seguida, execute novamente o comando do gráfico de upgrade.

5. Se o GlusterFS estiver instalado em seu cluster, deve-se fazer upgrade do cliente GlusterFS para a versão 4.1.5.
6. Atualize a anotação rewrite-target do ingresso NGINX. Para obter mais informações, consulte A anotação rewrite-target do ingresso NGINX falha ao fazer upgrade para o IBM Cloud Private Versão 3.2.0.

8. Verifique o status de seu upgrade.
   - Se o upgrade foi bem-sucedido, as informações de acesso para o seu cluster serão exibidas:

     `A URL da UI é https://<Cluster Master Host>:<Cluster Master API Port>`

     O valor `<Cluster Master Host>:<Cluster Master API Port>` é definido no Terminal Principal.
   - Se você encontrar erros, consulte Resolução de problemas.
9. Limpe o cache do seu navegador.
10. Se possuir aplicativos que usam recursos GPU ou uma cota de recursos para recursos GPU, será necessário atualizar manualmente a cota de aplicativo ou recurso com o novo nome do recurso GPU `nvidia.com/gpu`.
    - Para aplicativos que usam recursos GPU, siga as etapas em Criando uma implementação com recursos GPU conectados para executar um aplicativo GPU de amostra. Para seu próprio aplicativo GPU, é preciso atualizar o aplicativo para usar o novo nome de recurso GPU `nvidia.com/gpu`. Por exemplo, para atualizar as propriedades de implementação, é possível usar o console de gerenciamento (consulte Modificando uma implementação) ou a CLI `kubectl`.
    - Para atualizar a cota de recursos para recursos GPU, siga as etapas em Configurando a cota de recursos para configurar uma cota de recursos para seu namespace. Pra fazer upgrade, é preciso atualizar a cota de recurso para usar o nome do recurso GPU `nvidia.com/gpu`. Por exemplo, é possível definir a cota da GPU para `requests.nvidia.com/gpu: "2"`.
11. Acesse o seu cluster. Em um navegador da web, navegue para a URL para o seu cluster. Para obter uma lista de navegadores suportados, consulte Navegadores suportados.
    - Para obter mais informações sobre como acessar o cluster usando o IBM Cloud Private console de gerenciamento por meio de um navegador da web, consulte Acessando o seu cluster do IBM Cloud Private usando o console de gerenciamento.
    - Para obter mais informações sobre como acessar o cluster usando a linha de comandos do Kubernetes (kubectl), consulte Acessando o seu cluster do IBM Cloud Private usando a CLI kubectl. **Nota:** Após o upgrade, a política de segurança de pod para seus clusters é ativada automaticamente, mas configurada para a configuração menos restritiva para evitar problemas de acesso. Consulte Segurança de pod para obter informações sobre como gerenciar as configurações da política de segurança de pod.
12. Assegure-se de que todas as portas padrão do IBM Cloud Private estejam abertas. Para obter mais informações sobre as portas padrão do IBM Cloud Private, consulte Portas padrão.
13. Faça backup do nó de inicialização. Copie o seu diretório `/<new_installation_directory>/cluster` para um local seguro.
14. Se você usar Cloud Automation Manager em seu cluster do IBM Cloud Private, também deverá atualizá-lo. Consulte Fazendo upgrade do Cloud Automation Manager.
15. Limpe os gráficos obsoletos. Na liberação IBM Cloud Private 3.2.0, alguns gráficos, como `mariadb`, `auth-apikeys`, `unified-router` e `heapster`, são removidos; porém, eles não são excluídos automaticamente, caso você queira reverter para a liberação anterior. Apenas após você verificar o upgrade e o cluster estar operacional, execute o comando a seguir para remover os gráficos obsoletos:

```
helm delete --purge --tls --timeout=600  mariadb
helm delete --purge --tls --timeout=600  heapster
```

```
helm delete --purge --tls --timeout=600  unified-router
helm delete --purge --tls --timeout=600  auth-apikeys
```

Atualizando IBM Cloud Private-CE

É possível fazer upgrade do IBM Cloud Private-CE a partir de versões anteriores específicas.

Caminhos de Upgrade Suportados

É possível fazer upgrade apenas dos caminhos suportados a seguir:

- IBM Cloud Private-CE versão 3.1.2 para 3.2.0
- IBM Cloud Private-CE versão 3.1.1 para 3.2.0
- IBM Cloud Private-CE versão 3.1.0 para 3.2.0

Se você usar uma versão anterior do IBM Cloud Private-CE, deverá fazer upgrade primeiro para a versão 3.1.0.

É possível fazer upgrade somente de uma versão do IBM Cloud Private-CE para outra versão do IBM Cloud Private-CE. Não é possível fazer upgrade do IBM Cloud Private-CE para as edições Cloud Native ou Enterprise do IBM Cloud Private.

Durante o processo de upgrade, não é possível acessar a console de gerenciamento do IBM Cloud Private. Também não é possível configurar as opções do provedor em nuvem, como a configuração de um vSphere Cloud Provider, ou optar por usar o NSX-T.

Atualizando

1. Efetue login no nó de inicialização como um usuário com permissões raiz. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre tipos de nó, consulte Arquitetura. Durante a instalação, você especifica os endereços IP para cada tipo de nó.

2. Puxe a imagem do instalador do IBM Cloud Private-CE do Docker Hub.

   ```
   sudo docker pull ibmcom/icp-inception:3.2.0
   ```

3. Crie um diretório de instalação e copie os diretórios do `cluster` do diretório de instalação anterior para a nova pasta do IBM Cloud Private `cluster`. Use um diretório de instalação diferente do que você usou para a versão anterior. Por exemplo, para armazenar os arquivos de configuração em `/opt/ibm-cloud-private-3.2.0`, execute os comandos a seguir:

   ```
   mkdir -p /opt/ibm-cloud-private-3.2.0
   cd /opt/ibm-cloud-private-3.2.0
   cp -r /<installation_directory>/cluster .
   sudo rm -rf .upgrade upgrade_version
   ```

   Nota: `/<installation_directory>` é o caminho para o diretório de instalação da versão 3.1.2 e `/<new_installation_directory>` é o caminho completo para o diretório de instalação da versão 3.2.0.

4. Verifique o valor de parâmetro `calico_ipip_enabled` na versão a partir da qual você está fazendo upgrade.

   - Se o parâmetro foi configurado como `calico_ipip_enabled: true`, substitua o parâmetro no `/<new_installation_directory>/cluster/config.yaml` por `calico_ipip_mode: Always`.
   - Se o parâmetro foi configurado como `calico_ipip_enabled: false`, substitua o parâmetro no `/<new_installation_directory>/cluster/config.yaml` por `calico_ipip_mode: Never`.

5. Implemente seu ambiente concluindo as etapas a seguir:

   1. Mude a pasta `cluster` em seu diretório de instalação.

      ```
      cd / < new_installation_directory> /cluster
      ```

   2. Prepare o cluster para upgrade.

      ```
      sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster \
      ibmcom/icp-inception:3.2.0 upgrade-prepare
      ```

      Se a preparação do cluster falhar, revise a mensagem de erro e resolva qualquer problema. Em seguida, execute o comando `upgrade-prepare` novamente.

   3. Faça upgrade do Kubernetes.

```
sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster \
ibmcom/icp-inception:3.2.0 upgrade-k8s
```

- Se o upgrade do Kubernetes falhar com uma mensagem diferente, revise a mensagem de erro e resolva possíveis problemas. Em seguida, recupere os serviços do Kubernetes e execute o comando de upgrade dos serviços do Kubernetes novamente.

4. Gráfico de Upgrade.

```
sudo docker run -e LICENSE=accept --net=host --rm -t -v "$(pwd)":/installer/cluster \
ibmcom/icp-inception:3.2.0 upgrade-chart
```

- Se o upgrade do gráfico falhar com uma mensagem diferente, revise a mensagem de erro e resolva os problemas. Em seguida, execute novamente o comando do gráfico de upgrade.

5. Se o GlusterFS estiver instalado em seu cluster, deve-se fazer upgrade do cliente GlusterFS para a versão 4.1.5.
6. Atualize a anotação rewrite-target do ingresso NGINX. Para obter mais informações, consulte A anotação rewrite-target do ingresso NGINX falha ao fazer upgrade para o IBM Cloud Private Versão 3.2.0.

6. Verifique o status de seu upgrade.
   - Se o upgrade for bem-sucedido, as informações de acesso para seu cluster serão exibidas. O `<Cluster Master Host>` está definido em Terminal principal.

     `A URL da UI é https://<Cluster Master Host>:<Cluster Master API Port>`

     Em que `<Cluster Master Host>:<Cluster Master API Port>` é definido em Terminal principal.
   - Se você encontrar erros, consulte Resolução de problemas.
7. Limpe o cache do seu navegador.
8. Se possuir aplicativos que usam recursos GPU ou uma cota de recursos para recursos GPU, será necessário atualizar manualmente a cota de aplicativo ou recurso com o novo nome do recurso GPU `nvidia.com/gpu`.
   - Para aplicativos que usam recursos GPU, siga as etapas em Criando uma implementação com recursos GPU conectados para executar um aplicativo GPU de amostra. Para seu próprio aplicativo GPU, é preciso atualizar o aplicativo para usar o novo nome de recurso GPU `nvidia.com/gpu`. Por exemplo, para atualizar as propriedades de implementação, é possível usar o console de gerenciamento (consulte Modificando uma implementação) ou a CLI `kubectl`.
   - Para atualizar a cota de recursos para recursos GPU, siga as etapas em Configurando a cota de recursos para configurar uma cota de recursos para seu namespace. Pra fazer upgrade, é preciso atualizar a cota de recurso para usar o nome do recurso GPU `nvidia.com/gpu`. Por exemplo, é possível definir a cota da GPU para `requests.nvidia.com/gpu: "2"`.
9. Acesse o seu cluster. Em um navegador da web, navegue para a URL para o seu cluster. Para obter uma lista de navegadores suportados, consulte Navegadores suportados.
   - Para obter mais informações sobre como acessar o cluster usando o IBM Cloud Private console de gerenciamento por meio de um navegador da web, consulte Acessando o seu cluster do IBM Cloud Private usando o console de gerenciamento.
   - Para obter mais informações sobre como acessar o cluster usando a linha de comandos do Kubernetes (kubectl), consulte Acessando o seu cluster do IBM Cloud Private usando a CLI kubectl. **Nota:** Após o upgrade, a política de segurança de pod para seus clusters é ativada automaticamente, mas configurada para a configuração menos restritiva para evitar problemas de acesso. Consulte Segurança de pod para obter informações sobre como gerenciar as configurações da política de segurança de pod.
10. Assegure-se de que todas as portas padrão do IBM Cloud Private estejam abertas. Para obter mais informações sobre as portas padrão do IBM Cloud Private, consulte Portas padrão.
11. Faça backup do nó de inicialização. Copie o seu diretório `/<new_installation_directory>/cluster` para um local seguro.

Fazendo upgrade de gráficos do Helm no Catalog

É possível fazer upgrade dos gráficos do Helm para outros produtos que foram incluídos no IBM Cloud Private Catalog.

É possível obter os arquivos compactados para esses produtos por meio do IBM Passport Advantage®. Depois de instalar os novos arquivos, deve-se fazer upgrade das liberações existentes do Helm.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Antes de fazer upgrade de um gráfico, conclua os pré-requisitos a seguir:

- Certifique-se de atender aos pré-requisitos para instalar os produtos em pacote configurável. Consulte Instalando o software IBM no IBM Cloud Private.
- Instale a interface da linha de comandos (CLI) de Helm. Consulte Instalando a CLI do Helm (helm) .

Para fazer upgrade do software:

1. Obtenha o arquivo compactado a partir do IBM Passport Advantage ®.
2. Assegure-se de que você tenha espaço em disco suficiente para carregar as imagens nos arquivos compactados em seu computador.
    1. Verifique o uso do disco do Docker executando o seguinte comando:

        ```
        docker system df
        ```

        Para mais opções de comando, consulte docker system df na documentação do Docker.

    2. Se precisar de mais espaço em disco, execute uma das seguintes ações:
        - Remova as imagens antigas do Docker.
        - Aumente a quantia de armazenamento que o daemon do Docker usa. Para aumentar a quantia de armazenamento que o daemon do Docker usa, consulte a entrada para `dm.basesize` na documentação do Docker dockerd.
3. Efetue login em seu cluster a partir da CLI do IBM Cloud Private e efetue login no registro de imagem privado do Docker:

    ```
    cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
    docker login <cluster_CA_domain>:8500
    ```

    O `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal, e `cluster_CA_domain` é o domínio da autoridade de certificação (CA). Se você não especificou um domínio de CA, o valor padrão é `mycluster.icp`. Consulte Especificando sua própria autoridade de certificação (CA) para serviços IBM Cloud Private.

4. Instale uma nova versão do gráfico por meio do Passport Advantage:

    ```
    cloudctl catalog load-archive --archive <compressed_file_name>
    ```

    O `compressed_file_name` é o nome do arquivo transferido por download a partir do Passport Advantage, `cluster_CA_domain` é o domínio (CA) e `namespace` é o namespace do Docker que hospeda a imagem do Docker.

5. Visualize o gráfico no IBM Cloud PrivateCatalog:
    1. Na console de gerenciamento do IBM Cloud Private, selecione **Gerenciar** > **Repositórios do Helm**. A entrada da coluna *Última atualização* mostra quando esse repositório foi atualizado pela última vez.
    2. Clique em **Sincronizar repositórios**. Para sincronizar todos os repositórios.

        Tipo de usuário ou nível de acesso necessário para a sincronização, inclusão ou remoção de repositórios: Administrador de cluster

        Dica: Também é possível sincronizar um único repositório, selecionando o menu de ação (**...**), em seguida, selecionando **Sincronizar este repositório**.

    3. Selecione **Catálogo**.

        Ao instalar a nova versão do gráfico, os gráficos ficam visíveis no Catalog. É possível instalar qualquer versão em seu cluster.

6. Faça upgrade da liberação do Helm existente que usa o gráfico executando o seguinte comando:

    ```
    helm upgrade <releaseName> http://<Cluster Master Host>:<Cluster Master API Port>/helm-repo/requiredAssets/<chartName>-<chartVersion>.tgz
    ```

Neste comando:

- `<releaseName>` é o nome da liberação do Helm existente para fazer upgrade. É possível localizar o nome da liberação na console de gerenciamento.
- `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal.
- `<chartName>` é o nome do gráfico que a liberação do Helm usa. É possível obter o nome do gráfico na página do Catalog.
- `<chartVersion>` é a nova versão do gráfico a ser usada. É possível obter a versão do gráfico clicando no novo gráfico no Catalog.

Fazendo upgrade do pacote do Docker do IBM Cloud Private

Fazendo upgrade dos mecanismos do Docker que foram instalados usando o pacote do Docker do IBM Cloud Private.

- Fazendo upgrade do pacote do Docker do IBM Cloud Private (nó de inicialização)
- Fazendo upgrade do pacote do Docker do IBM Cloud Private (nós do cluster)

Fazendo upgrade do pacote do Docker do IBM Cloud Private (nó de inicialização)

Faça upgrade de um nó de inicialização que foi instalado usando o pacote do Docker do IBM Cloud Private.

1. Faça download do pacote do Docker para a sua plataforma. Consulte Pacotes do Docker do IBM Cloud Private.
2. Faça upgrade do Docker em seu nó de inicialização.
   - Para o Linux®, execute este comando:

     ```
     chmod +x icp-docker-18.03.1_x86_64.bin
     sudo ./icp-docker-18.03.1_x86_64.bin --upgrade
     ```

   - Para o Linux® on Power® (ppc64le), execute este comando:

     ```
     chmod +x icp-docker-18.03.1_ppc64le.bin
     sudo ./icp-docker-18.03.1_ppc64le.bin --upgrade
     ```

3. Assegure-se de que o mecanismo de Docker esteja iniciado. Execute o comando a seguir:

   ```
   sudo systemctl start docker
   ```

Fazendo upgrade do pacote do Docker do IBM Cloud Private (nós do cluster)

Faça upgrade dos nós do cluster que foram instalados usando o pacote do Docker do IBM Cloud Private.

1. Alterne para o diretório `/<installation_directory>/cluster/`.

   ```
   cd /<installation_directory>/cluster/
   ```

1. Fazer upgrade do Docker.

   ```
   sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee upgrade-docker
   ```

   Também é possível usar a opção `-l` para fazer upgrade de nós do cluster específicos.

   - Para nós do trabalhador, use `-l worker`.
   - Em ambientes de HA, use o `-l <host_ip>` para fazer upgrade dos nós principal e do proxy, um de cada vez.

2. Em todos os nós, assegure-se de que mecanismo do Docker esteja iniciado. Execute o comando a seguir:

   ```
   sudo systemctl start docker
   ```

Retendo os dados de monitoramento durante o upgrade

Retenha os dados de monitoramento durante o upgrade.

No IBM® Cloud Private Versão 3.1.2, se você provisionou dinamicamente o armazenamento para o serviço de monitoramento, os dados serão perdidos durante o upgrade. Se você usou armazenamento local para os dados de monitoramento, é possível concluir as etapas nas seções a seguir para reter os dados durante o upgrade.

Atualizar volumes persistentes existentes

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Obtenha uma lista de todos os volumes persistentes (PVs) em seu cluster. Anote o PV que cada componente do serviço de monitoramento usa.

   ```
   kubectl get pv
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                            CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS    CLAIM
   STORAGECLASS               REASON    AGE
   helm-repo-pv                    5Gi        RWO            Delete           Bound     kube-
   system/helm-repo-pvc                  helm-repo-storage                     1d
   image-manager-10.10.24.83       20Gi       RWO            Retain           Bound     kube-
   system/image-manager-image-manager-0   image-manager-storage                1d
   logging-datanode-10.10.24.83    20Gi       RWO            Retain           Bound     kube-
   system/data-logging-elk-data-0         logging-storage-datanode             1d
   mongodb-10.10.24.83             20Gi       RWO            Retain           Bound     kube-
   system/mongodbdir-icp-mongodb-0        mongodb-storage                      1d
   alertmanager-pv                 1Gi        RWO            Delete           Bound
   default/my-release-prometheus-alertmanager
   17h
   ```

3. Execute o comando a seguir para cada PV que o serviço de monitoramento usa. O comando muda a política de recuperação PV de `Delete` para `Retain`:

   ```
   kubectl patch pv <PV name> -p '{"spec":{"persistentVolumeReclaimPolicy":"Retain"}}'
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl patch pv alertmanager-pv -p '{"spec":{"persistentVolumeReclaimPolicy":"Retain"}}'
   persistentvolume "alertmanager-pv" patched
   ```

4. Verifique se os PVs são atualizados.

   ```
   kubectl get pv
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                            CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS    CLAIM
   STORAGECLASS               REASON    AGE
   helm-repo-pv                    5Gi        RWO            Delete           Bound     kube-
   system/helm-repo-pvc                  helm-repo-storage                     1d
   image-manager-10.10.24.83       20Gi       RWO            Retain           Bound     kube-
   system/image-manager-image-manager-0   image-manager-storage                1d
   logging-datanode-10.10.24.83    20Gi       RWO            Retain           Bound     kube-
   system/data-logging-elk-data-0         logging-storage-datanode             1d
   mongodb-10.10.24.83             20Gi       RWO            Retain           Bound     kube-
   system/mongodbdir-icp-mongodb-0        mongodb-storage                      1d
   alertmanager-pv                  1Gi       RWO            Retain           Bound     default/my-
   release-prometheus-alertmanager                                                       17h
   ```

Criar volumes persistentes

Para cada PV que os componentes de serviço de monitoramento usam, crie um novo PV. Deve-se designar o novo PV ao mesmo nó para o qual o PV antigo foi designado.

Os requisitos de armazenamento padrão para PersistentVolumes são os seguintes:

- Prometheus: 10Gi
- Grafana: 1Gi

- Gerenciador de Alerta: 1Gi

Deve-se atualizar as definições PersistentVolume com base nos requisitos de armazenamento que você define no gráfico Helm. Para assegurar que os dados existentes sejam preservados durante o upgrade, deve-se usar a mesma classe de armazenamento que você usou nos PVs existentes.

A seguir está um exemplo de definição de um novo PV para o Gerenciador de alertas.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: alertmanager-data
  labels:
    component: alertmanager
  annotations:
    "volume.alpha.kubernetes.io/node-affinity": '{
      "RequiredDuringSchedulingIgnoredDuringExecution": {
        "NodeSelectorTerms": [
          {
            { "key": "kubernetes.io/hostname",
              "operator": "In",
              "values": [  "10.10.24.83" ]
            }
          ]}
         ]}
        }'
spec:
  storageClassName: monitoring-storage
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: "/opt/ibm/cfc/monitoring/alertmanager"
  persistentVolumeReclaimPolicy: Retain
```

Atualize o arquivo config.yaml

Atualize o arquivo `config.yaml` na pasta `/<installation_directory>/cluster`. Consulte Configurando o serviço de monitoramento. Assegure-se de mudar o valor do parâmetro `persistentVolume.enabled` de `false` para `true`. Além disso, assegure-se de incluir o nome da classe de armazenamento que você está usando para o volume persistente no parâmetro `persistentVolume.storageClass`.

Em seguida, continue com o upgrade do cluster.

Retendo a configuração do KMS durante o upgrade

Se você configurar o IBM Cloud Private para usar o plug-in do KMS para criptografia, será possível concluir as etapas na seção a seguir para reter sua configuração do KMS existente.

1. Faça backup dos arquivos a seguir para uma pasta fora do IBM Cloud Private. Por exemplo:

   ```
   cp /etc/cfc/conf/encryption-config.yaml /root/backup/
   cp /etc/cfc/conf/kmsplugin-config.yaml /root/backup/
   ```

2. Desative a criptografia.

   1. Coloque o provedor de identidade como a primeira entrada no arquivo `/etc/cfc/conf/encryption-config.yaml`.

      ```
      apiVersion: apiserver.config.k8s.io/v1
         kind: EncryptionConfiguration
         resources:
             - resources:
                - secrets
                providers:
                - identity: {}
                - kms:
                   name : KmsPlugin
      ```

```
endpoint: unix:///var/run/keyprotectprovider.sock
cachesize: 100
```

2. Reinicie os processos `kube-apiserver` e `kmsplugin`.

3. Decriptografe todos os segredos que foram criptografados. Por exemplo, execute o comando a seguir para forçar todos os segredos no namespace `default` a serem decriptografados.

   ```
   kubectl get secrets -n default -o json | kubectl replace -f -
   ```

3. Siga o Procedimento de upgrade do IBM Cloud Private para continuar com seu upgrade.

4. Após o upgrade ser concluído, siga o procedimento Criptografando segredos do Kubernetes com o plug-in do Key Management Service e use seus arquivos de backup para ativar a criptografia.

5. Criptografe segredos existentes. Por exemplo, execute o comando a seguir para forçar todos os segredos no namespace `default` a serem criptografados.

   ```
   kubectl get secrets -n default -o json | kubectl replace -f -
   ```

Revertendo

É possível reverter seu cluster do IBM® Cloud Private .

- Retendo os dados de monitoramento durante o upgrade
- Revertendo para uma versão anterior do IBM Cloud Private
- Revertendo para uma versão anterior do IBM Cloud Private-CE
- Revertendo o pacote Docker do IBM Cloud Private

Revertendo para uma versão anterior do IBM Cloud Private

Depois de fazer upgrade para o IBM Cloud Private 3.2.0, será possível reverter para sua última versão instalada.

É possível reverter seu cluster com upgrade efetuado somente nos cenários a seguir:

- Se o upgrade falhar.
- Você não criou novas cargas de trabalho ou implementações. Se essas operações foram executadas com sucesso em seu cluster, a reversão poderá falhar.

Para reverter sua versão instalada, conclua as etapas a seguir.

1. Retroceda os gráficos:

   ```
   sudo docker run -e LICENSE=accept --net=host --rm -t -v \
   "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee
   rollback-chart
   ```

2. Roll back Kubernetes:

   ```
   sudo docker run -e LICENSE=accept --net=host --rm -t -v \
   "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee
   rollback-k8s
   ```

Revertendo para uma versão anterior do IBM Cloud Private-CE

Depois de fazer upgrade para o IBM Cloud Private-CE 3.2.0, será possível reverter para sua última versão instalada.

É possível reverter seu cluster com upgrade efetuado somente nos cenários a seguir:

- Se o upgrade falhar.
- Você não criou novas cargas de trabalho ou implementações. Se essas operações foram executadas com sucesso em seu cluster, a reversão poderá falhar.

Para reverter sua versão instalada, conclua as etapas a seguir.

1. Retroceda os gráficos.

```
sudo docker run -e LICENSE=accept --net=host --rm -t -v \
"$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 rollback-chart
```

2. Recuperar o Kubernetes.

```
sudo docker run -e LICENSE=accept --net=host --rm -t -v \
  "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 rollback-k8s
```

Revertendo o pacote Docker do IBM Cloud Private

Reverta a versão do pacote do Docker do IBM Cloud Private, que é usado em seu cluster, para uma versão anterior.

- Revertendo o pacote do Docker do IBM Cloud Private (nó de inicialização)
- Revertendo o pacote do Docker do IBM Cloud Private (nós do cluster)

Revertendo o pacote do Docker do IBM Cloud Private (nó de inicialização)

Reverta a versão do pacote do Docker do IBM Cloud Private, que é usado em sua inicialização do nó.

1. Faça download do pacote do Docker para a sua plataforma. Consulte IBM Cloud Private Pacotes do Docker.
2. Reverta a versão do Docker em seu nó de inicialização.
   - Para o Linux®, execute este comando:

     ```
     chmod +x icp-docker-18.03.1_x86_64.bin
     sudo ./icp-docker-18.03.1_x86_64.bin --rollback
     ```

   - Para o Linux® on Power® (ppc64le), execute este comando:

     ```
     chmod +x icp-docker-18.03.1_ppc64le.bin
     sudo ./icp-docker-18.03.1_ppc64le.bin --rollback
     ```

3. Assegure-se de que o mecanismo de Docker esteja iniciado. Execute o comando a seguir:

   ```
   sudo systemctl start docker
   ```

Revertendo o pacote do Docker do IBM Cloud Private (nós do cluster)

Reverta a versão do Docker para nós do cluster que foram instalados usando o pacote do Docker do IBM Cloud Private.

1. Alterne para o diretório `/<installation_directory>/cluster/`.

   ```
   cd /<installation_directory>/cluster/
   ```

2. Reverta a versão do Docker.

   ```
   sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster ibmcom/icp-
   inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee rollback-docker
   ```

   Também é possível usar a opção `-l` para fazer upgrade de nós do cluster específicos.

   - Para nós do trabalhador, use `-l worker`.
   - Em ambientes de HA, use o `-l <host_ip>` para fazer upgrade de nós principal e do proxy, um por vez.

3. Em todos os nós, assegure-se de que mecanismo do Docker esteja iniciado. Execute o comando a seguir:

   ```
   sudo systemctl start docker
   ```

Configurando o TLS e conjuntos de cifras para o gerenciador de imagem e o registro do IBM Cloud Private

É possível customizar os conjuntos de cifras do TLS e a versão mínima do TLS para o registro do IBM Cloud Private. As cifras listadas são ativadas no componente e todas as outras são desativadas.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Pré-requisitos

- Deve-se ter um cluster funcional do IBM Cloud Private com pods `image-manager` em execução.
- Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

Ativando o TLS e conjuntos de cifras para o gerenciador de imagem e o registro do IBM Cloud Private

1. Use a CLI de `kubectl` para editar o statefulset de `image-manager`:

   ```
   kubectl edit statefulset -n kube-system image-manager
   ```

2. No contêiner de `image-manager`, após o bloco `env`, inclua as seguintes configurações:

   ```
    - image: ibmcom/icp-image-manager-amd64:2.2.4
      env:
      - name: TLS_MIN_VERSION
        value: tls1_2
      - name: TLS_CIPHERS_SUITES
        value:
   TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_W
   ITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
   ,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SH
   A384,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA
   ```

3. No contêiner `icp-registry`, após o bloco `env`, inclua as seguintes configurações:

   ```
    - image: ibmcom/registry-amd64:2.6.2.2
      env:
      - name: TLS_MIN_VERSION
        value: tls1_2
      - name: TLS_CIPHERS_SUITES
        value:
   TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WIT
   H_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA
   384,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
   ```

   A configuração é semelhante ao seguinte exemplo:

   ```
    spec:
      containers:
        - image: ibmcom/registry-amd64:2.6.2.2
          env:
          - name: TLS_MIN_VERSION
            value: tls1_2
          - name: TLS_CIPHERS_SUITES
            value:
   TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WIT
   H_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA
   384,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

      - command:
        - /icp-image-manager
        - serve
        - --listen-address=0.0.0.0:8600
        - --private-key-path=/etc/icp-image-manager/tls.key
        - --oidc-url=http://platform-identity-provider:4300
        - --registry-url=https://127.0.0.1:8500
        - --registry-server-name=mycluster.icp:8500
        - --enable-https=true
        env:
        - name: TLS_MIN_VERSION
          value: tls1_2
        - name: TLS_CIPHERS_SUITES
          value:
   TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_W
   ITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
   ,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SH
   A384,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA
        image: ibmcom/icp-image-manager-amd64:2.2.4
   ```

4. Salve as alterações e saia do editor. O Kubernetes reinicia os pods do gerenciador de imagem do IBM Cloud Private para incluir as mudanças mais recentes.

Verificando a versão do TLS e os conjuntos de cifras que foram ativados

1. Crie um script. Por exemplo:

```
#!/usr/bin/env bash

if [[ $# -eq 0 ]] ; then
    echo 'Usage:  ./scan_ciphers.sh hostname:port'
    exit 1
fi

SERVER=$1
DELAY=0.01
ciphers=$(openssl ciphers 'ALL:eNULL' | sed -e 's/:/ /g')
protocols="ssl3 tls1 tls1_1 tls1_2"

echo Obtaining cipher list from $(openssl version).

for protocol in ${protocols[@]} ; do
  echo -n "${protocol/_/.} " | tr '[:lower:]' '[:upper:]' ; echo -e "ciphers :"
  for cipher in ${ciphers[@]} ; do
    openssl s_client -connect $SERVER -cipher $cipher -$protocol < /dev/null > /dev/null 2>&1
&& echo -e "\t$cipher"
    sleep $DELAY
  done
done
```

2. Inclua a permissão executável para o script:

```
chmod +x scan-ciphers.sh
```

3. Para a porta de registro 8500, executando o seguinte comando:

```
./scan-ciphers.sh <master_IP>:8500
```

Sua saída é semelhante ao seguinte exemplo:

```
Obtaining cipher list from OpenSSL 1.0.2k-fips 26 Jan 2019.
SSL3 ciphers :
TLS1 ciphers :
TLS1.1 ciphers :
TLS1.2 ciphers :
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-SHA256
```

4. Para a porta 8600 do `image-manager`, executando o seguinte comando:

```
./scan-ciphers.sh <master_IP>:8600
```

Sua saída pode ser semelhante ao seguinte código:

```
Obtaining cipher list from OpenSSL 1.0.2k-fips 26 Jan 2019.
SSL3 ciphers :
TLS1 ciphers :
TLS1.1 ciphers :
TLS1.2 ciphers :
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA
AES256-GCM-SHA384
AES256-SHA
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-SHA
AES128-GCM-SHA256
AES128-SHA
```

Ativando e desativando componentes do IBM Cloud Private

O IBM® Cloud Private inclui vários componentes que são compostos de um ou mais serviços de gerenciamento.

Depois de instalar o IBM Cloud Private, é possível ativar ou desativar serviços de gerenciamento que são incluídos em um componente. Para obter mais informações sobre os valores padrão para os serviços de gerenciamento, consulte Customizando o cluster com o arquivo config.yaml. Para obter mais informações sobre os componentes que estão disponíveis e sobre os serviços de gerenciamento que estão incluídos com o componente, consulte Componentes do IBM® Cloud Private. Este tópico abrange a plataforma com a qual um serviço ou dependências de serviço podem ser executadas.

Consulte Componentes do IBM® Cloud Private para obter mais informações sobre os componentes que estão disponíveis e os serviços de gerenciamento que estão incluídos com o componente.

Tipo de usuário ou nível de acesso necessário: Administrador de cluster.

Se você estiver ativando ou desativando um serviço, deve-se configurar a interface da linha de comandos (CLI) do `helm` como um usuário administrador de cluster. Para obter mais informações sobre como configurar a CLI do Helm, consulte Instalando a CLI do Helm (helm).

1. Se você estiver fazendo upgrade para a versão 3.1.0 ou mais recente, deverá reformatar a seção de serviços de gerenciamento no arquivo `config.yaml` antes de fazer upgrade. A seção do arquivo antes do upgrade é semelhante ao exemplo a seguir:

   ```
   disabled_management_services: [ "istio", "vulnerability-advisor", "custom-metrics-adapter" ]
   ```

   A seção do arquivo após as mudanças para o upgrade é semelhante ao exemplo a seguir:

   ```
   management-services:
     istio: disabled
     vulnerability-advisor: disabled
     custom-metrics-adapter: disabled
   ```

   Se você estiver ativando o `vulnerability-advisor` após o upgrade, implemente os novos nós do consultor de vulnerabilidade (VA). Para obter mais informações sobre como implementar os novos nós do orientador de vulnerabilidade, consulte Incluindo um nó de cluster do IBM Cloud Private.

   Nota: se você ativou `vulnerability-advisor` na versão anterior, assegure-se de que a entrada `vulnerability-advisor` esteja ativada na seção `management-services` do arquivo `config.yaml` após o upgrade. Seu parâmetro `vulnerability-advisor` pode ser semelhante ao seguinte valor de parâmetro: `vulnerability-advisor: enabled`. A configuração é desativada, por padrão, na versão de upgrade e a configuração não é retida automaticamente durante o upgrade.

2. Inclua um serviço na lista de parâmetros `management_services` no arquivo `config.yaml` para desativar ou ativar um serviço. Mude o valor de parâmetro de serviço para `disabled` para desativar um serviço ou mude o valor de parâmetro de serviço para `enabled` para ativar o serviço.

   Importante: deve-se também ativar ou desativar todos os serviços que compõem um componente. Os serviços a seguir não podem ser desativados: `tiller`, `calico/nsx-t`, `kube-dns`, `monitoring-crd`, `cert-manager`.

3. Execute o comando de complemento para ativar ou desativar o serviço em sua arquitetura de CPU:

   ```
    docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
   ```

   Se o IBM Cloud Private estiver instalado com o OpenShift, execute o seguinte comando para ativar ou desativar o serviço:

   ```
    sudo docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-rhel-ee install-on-openshift
   ```

Os serviços de gerenciamento do IBM Cloud Private têm relacionamentos de dependência entre si. Por exemplo, o serviço `auth-idp` depende do serviço `mongodb`. Se o `mongodb` estiver desativado, o serviço `auth-idp` não poderá funcionar.

Nota: os relacionamentos de dependência são válidos apenas quando `tiller`, `calico/nsx-t`, `kube-dns`, `monitoring-crd` e `cert-manager` estiverem ativados.

Visualize a tabela a seguir dos serviços de gerenciamento do IBM Cloud Private e das dependências:

Tabela 1. Dependências do serviço de gerenciamento doIBM Cloud Private

Serviço de gerenciamento	Dependências	Plataformas suportadas

Serviço de gerenciamento	Dependências	Plataformas suportadas
kmsplugin	IAM, key-management	IBM Cloud Private
tiller		IBM Cloud Private
image-manager		IBM Cloud Private
kube-dns		IBM Cloud Private
calico		IBM Cloud Private
nsx-t		IBM Cloud Private
cert-manager		IBM Cloud Private, IBM Cloud Private with OpenShift
mongodb		IBM Cloud Private, IBM Cloud Private with OpenShift
monitoring-crd		IBM Cloud Private, IBM Cloud Private with OpenShift
auth-idp	mongodb	IBM Cloud Private, IBM Cloud Private with OpenShift
auth-apikeys	mongodb	IBM Cloud Private, IBM Cloud Private with OpenShift
auth-pap	mongodb	IBM Cloud Private, IBM Cloud Private with OpenShift
auth-pdp	mongodb, auth-idp, auth-pap, auth-apikeys	IBM Cloud Private, IBM Cloud Private with OpenShift
catalog-ui	auth-idp, platform-api, helm-api, helm-repo, multicluster-hub	IBM Cloud Private, IBM Cloud Private with OpenShift
custom-metrics-adapter	monitoring	IBM Cloud Private, IBM Cloud Private with OpenShift
helm-api	mongodb, platform-api, icp-management-ingress, helm-repo, mgmt-repo	IBM Cloud Private, IBM Cloud Private with OpenShift
helm-repo	mongodb	IBM Cloud Private, IBM Cloud Private with OpenShift
icp-management-ingress		IBM Cloud Private, IBM Cloud Private with OpenShift
image-security-enforcement		IBM Cloud Private
nvidia-device-plugin		IBM Cloud Private
key-management	IAM, mongodb	IBM Cloud Private
logging	IAM	IBM Cloud Private, IBM Cloud Private with OpenShift
metering	mongodb, IAM	IBM Cloud Private, IBM Cloud Private with OpenShift
metrics-server		IBM Cloud Private, IBM Cloud Private with OpenShift
nginx-ingress		IBM Cloud Private, IBM Cloud Private with OpenShift
mgmt-repo	mongodb	IBM Cloud Private, IBM Cloud Private with OpenShift
monitoring	IAM	IBM Cloud Private, IBM Cloud Private with OpenShift
multicluster-hub	mongodb monitorando o IAM	IBM Cloud Private
platform-api	IAM	IBM Cloud Private, IBM Cloud Private with OpenShift
platform-ui	auth-idp, platform-api, catalog-ui, image-manager	IBM Cloud Private, IBM Cloud Private with OpenShift
secret-watcher		IBM Cloud Private, IBM Cloud Private with OpenShift

Serviço de gerenciamento	Dependências	Plataformas suportadas
security-onboarding	IAM	IBM Cloud Private, IBM Cloud Private with OpenShift
service-catalog	metrics-server	IBM Cloud Private
storage-glusterfs	monitoring	IBM Cloud Private
storage-minio	icp-management-ingress, monitoring	IBM Cloud Private
vulnerability-advisor	logging, image-manager, IAM	IBM Cloud Private
web-terminal	platform-api, IAM	IBM Cloud Private, IBM Cloud Private with OpenShift
multicluster-hub	IAM, monitoring, mongodb	IBM Cloud Private
multicluster-endpoint	monitoring	IBM Cloud Private
system-healthcheck-service	icp-management-ingress	IBM Cloud Private

Nota: o Identity and Access Management (IAM) inclui os serviços a seguir: `auth-idp`, `auth-pap`, `auth-pdp`, `auth-apikeys` e `secret-watcher`.

Gerenciando Clusters etcd

IBM® Cloud Private usa etcd. Use a documentação do etcd, como um guia para manter o etcd no IBM Cloud Private.

Cota de espaço

Use o comando de sinalização `--quota-backend-bytes` para configurar a cota de espaço. O valor padrão para a cota de espaço é 2 GB, que é uma cota de espaço conservativa adequada para a maioria dos aplicativos. O valor máximo é 8 GB.

É possível alterar o valor da cota de espaço antes ou após a instalação.

- Para configurar o `--quota-backend-bytes` antes de instalar o IBM Cloud Private, edite `cluster/config.yaml` da seguinte forma, usando 1G como exemplo:

  ```
  etcd_extra_args: [ "---cota-backend-bytes=1073741824" ]
  ```

- Para configurar `--quota-backend-bytes` depois de instalar o IBM Cloud Private, edite o arquivo `/etc/cfc/pods/etcd.json` no nó principal do IBM Cloud Private e inclua a seguinte linha no comando etcd:

  ```
  "--cota-backend-bytes=1073741824",
  ```

Para obter mais informações sobre a cota de espaço, consulte a documentação do etcd.

Compactação de histórico

O IBM® Cloud Private inclui a sinalização `--etcd-compaction-interval` para a configuração do intervalo de compactação do etcd no servidor de API. O valor padrão do intervalo de compactação é de 5 minutos, que também é o valor usado pelo IBM Cloud Private.

É possível mudar o valor do intervalo antes ou depois da instalação.

- Para configurar o `--etcd-compaction-interval` antes de instalar o IBM Cloud Private, edite o cluster/config.yaml como segue, usando 1 hora como exemplo:

  ```
  kube_apiserver_extra_args: ["--etcd-compaction-interval=1h"]
  ```

- Para configurar o `--etcd-compaction-interval` depois de instalar o IBM Cloud Private, edite o arquivo `/etc/cfc/pods/master.json` no nó principal do IBM Cloud Private e inclua a linha a seguir no comando `apiserver`:

  ```
  "--etcd-compaction-interval=1h",
  ```

Para obter mais informações sobre a compactação, consulte a documentação do etcd.

Desfragmentação

A desfragmentação libera o espaço de armazenamento de volta para o sistema de arquivos.

O IBM Cloud Private versão 3.10 não fornece uma configuração padrão para a desfragmentação. É possível executar uma tarefa para executar o processo de desfragmentação. Como alternativa, execute uma tarefa cron para executar a desfragmentação periodicamente para evitar que a cota de espaço seja atingida com base na carga de trabalho do cluster.

Nota: a desfragmentação de um membro em tempo real evita que o sistema leia e grave dados durante a reconstrução de seus estados. Considere executar sua tarefa durante o tempo de manutenção.

Para obter mais informações sobre a desfragmentação, consulte a documentação do etcd.

Executando uma Tarefa de Desfragmentação

Conclua as etapas a seguir para criar uma tarefa e executar o processo de desfragmentação.

1. No exemplo de arquivo `job etcd-defrag-job.yaml` a seguir, substitua `10.10.25.10 10.10.25.11` pelo seu IP de nó etcd (separado por um espaço).

```
apiVersion: batch/v1
kind: Job
metadata:
name: etcd-defrag-job
spec:
template:
 spec:
   containers:
   - name: etcd
     image: ibmcom/etcd:v3.2.18
     args:
     - /bin/sh
     - -c
     - etcdctl='etcdctl --cacert=/etc/cfc/conf/etcd/ca.pem --cert=/etc/cfc/conf/etcd/client.pem
--key=/etc/cfc/conf/etcd/client-key.pem';
       export ETCDCTL_API=3;
       for endpoint in 10.10.25.10 10.10.25.11 10.10.25.12;
       do
         $etcdctl --endpoints="https://${endpoint}:4001" defrag;
         $etcdctl --endpoints="https://${endpoint}:4001" --write-out=table endpoint status;
       done;
       $etcdctl --endpoints="https://${endpoint}:4001" alarm disarm;
       $etcdctl --endpoints="https://${endpoint}:4001" alarm list;
     volumeMounts:
     - mountPath: /etc/cfc/conf/etcd
       name: etcd-certs
   volumes:
   - hostPath:
       path: /etc/cfc/conf/etcd
       type: ""
        name: etcd-certs restartPolicy: OnFailure nodeSelector: etcd: "true" tolerations: -
key: "dedicated" operator: "Exists" effect: "NoSchedule"
```

2. Crie uma tarefa a partir da IU da web ou executando o seguinte comando:

```
$ kubectl create -f ./etcd-defrag-job.yaml -n kube-system
job.batch/etcd-defrag-job created
```

3. Depois de criar a tarefa, insira o comando a seguir para ver o status da tarefa:

```
$ kubectl get job -n kube-system | grep etcd-defrag-job
NAME              DESIRED   SUCCESSFUL   AGE
etcd-defrag-job   1         1            1m
```

4. Consulte os logs do pod para visualizar detalhes de desfragmentação.

```
$ kubectl logs etcd-defrag-job-48kxs -n kube-system
Finished defragmenting etcd member[https://10.10.25.10:4001]
+--------------------------+------------------+---------+---------+-----------+-----------+----
--------+
|         ENDPOINT         |        ID        | VERSION | DB SIZE | IS LEADER | RAFT TERM |
RAFT INDEX |
+--------------------------+------------------+---------+---------+-----------+-----------+----
--------+
| https://10.10.25.10:4001 | 8271bc8ee51f9f39 |  3.2.18 |  9.4 MB |     false |      6051 |
```

```
255138 |
+--------------------------+------------------+---------+---------+-----------+-----------+---
--------+
Finished defragmenting etcd member[https://10.10.25.11:4001]
+--------------------------+------------------+---------+---------+-----------+-----------+---
---------+
|         ENDPOINT         |        ID        | VERSION | DB SIZE | IS LEADER | RAFT TERM |
RAFT INDEX |
+--------------------------+------------------+---------+---------+-----------+-----------+---
---------+
| https://10.10.25.11:4001 | 6e235e51838ea635 |  3.2.18 |  9.3 MB |     false |      6051 |
255152 |
+--------------------------+------------------+---------+---------+-----------+-----------+---
---------+
…
```

Utilizando uma Tarefa Cron para Desfragmentação

É possível executar uma tarefa cron durante um horário de manutenção planejado. O exemplo a seguir de uma tarefa cron é executado a cada minuto para teste. Conclua as etapas a seguir antes de criar a tarefa cron.

1. Substitua `10.10.25.10 10.10.25.11` pelo seu IP de nó etcd (separado por um espaço).
2. Modifique `spec.schedule` para configurar sua própria tabela de tempo.

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
name: etcd-defrag-cronjob
spec:
schedule: "*/1 * * * *"
  jobTemplate:
 spec:
   template:
     spec:
       containers:
       - name: etcd
         image: ibmcom/etcd:v3.2.18
         args:
         - /bin/sh
         - -c
         - etcdctl='etcdctl --cacert=/etc/cfc/conf/etcd/ca.pem --
cert=/etc/cfc/conf/etcd/client.pem --key=/etc/cfc/conf/etcd/client-key.pem';
           export ETCDCTL_API=3;
           for endpoint in 10.10.25.10 10.10.25.11 10.10.25.12 ;
           do
             $etcdctl --endpoints="https://${endpoint}:4001" defrag;
             $etcdctl --endpoints="https://${endpoint}:4001" --write-out=table endpoint status;
           done;
           $etcdctl --endpoints="https://${endpoint}:4001" alarm disarm;
           $etcdctl --endpoints="https://${endpoint}:4001" alarm list;
         volumeMounts:
         - mountPath: /etc/cfc/conf/etcd
           name: etcd-certs
       volumes:
       - hostPath:
           path: /etc/cfc/conf/etcd
           type: ""
            name: etcd-certs restartPolicy: OnFailure nodeSelector: etcd: "true" tolerations: -
key: "dedicated" operator: "Exists" effect: "NoSchedule"
```

Criando ConfigMaps

É possível armazenar dados de configuração em pares chave-valor para aplicar pods ou componentes do sistema, como controladores.

Os arquivos de configuração ou ConfigMaps, contêm informações de configuração que são desacopladas do conteúdo da imagem. Ao desacoplar os dados de configuração, os aplicativos conteinerizados são mais móveis.

Para obter mais informações sobre os arquivos ConfigMap, consulte https://kubernetes.io/docs/tasks/configure-pod-container/configmap/.

Dois formatos estão disponíveis para você criar arquivos de configuração na console de gerenciamento.

É possível criar arquivos de configuração inserindo os valores de parâmetro na janela Criar ConfigMaps ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando arquivos de configuração usando a janela Criar ConfigMaps

1. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
2. Clique em **Criar ConfigMap**.
3. Forneça um nome para seu ConfigMap e insira os detalhes de configuração em pares chave-valor.
4. Clique em **Criar**.

Criando arquivos de configuração usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Cole um arquivo JSON ou YAML na caixa de diálogo "Criar recurso".
3. Clique em **Criar**.

Configurando uma mensagem de notificação de uso do sistema

Uma mensagem de notificação de uso do sistema fornece termos de uso de seu sistema. É possível configurar e atualizar a caixa de diálogo de notificação do sistema para seu sistema. É possível ativar a notificação do sistema antes ou depois de instalar o IBM Cloud Private.

Configurando a notificação de uso do sistema (pré-instalação)

Antes de instalar o IBM Cloud Private, configure a notificação do sistema.

Em seu arquivo `config.yaml`, inclua a configuração a seguir:

```
uiconfig:
  loginDialog:
    enable: true
    headerText: "Header text here"
    dialogText: "Your dialog text here"
    acceptText: "Your acceptance text here"
```

Salve a configuração que você incluiu e continue com sua instalação do IBM Cloud Private.

Configurando a notificação de uso do sistema (pós-instalação)

Antes de ativar a caixa de diálogo de notificação do sistema, deve-se instalar o IBM Cloud Private.

Conclua as etapas a seguir para ativar a caixa de diálogo de notificação do sistema:

1. SSH em seu nó de inicialização / principal. Para obter mais informações, consulte Compartilhando chaves SSH.

2. Execute o comando a seguir para editar seu ConfigMap:

   ```
   kubectl edit configmap platform-ui-config -n kube-system
   ```

3. Em seu ConfigMap, localize o parâmetro `ui-config.json`. Localize o parâmetro `loginDialog` . Ative a caixa de diálogo de notificação do sistema mudando o valor `enable` para `true`.

   Seu mapa de configuração pode ser semelhante à saída a seguir:

   ```
   ui-config.json:
     {
       "loginDialog": {
       "enable": "true",
       "headerText": "Header text here",
       "dialogText": "Your dialog text here",
       "acceptText": "Your acceptance text here"
   ```

```
    }
}
```

4. Atualize os parâmetros a seguir: `headerText`, `dialogText`, `acceptText`.

5. Salve seu mapa de configuração com suas mudanças atualizadas.

Nota: sua mensagem de notificação de uso do sistema aparece antes de você efetuar login em seu cluster do IBM Cloud Private. Os termos para seu sistema devem ser reconhecidos e aceitos.

Incluindo ou removendo nós do cluster do IBM Cloud Private

É possível incluir ou remover nós do cluster. Não é possível incluir ou remover nós principais.

- Incluindo um nó do cluster do IBM Cloud Private
- Incluindo um nó do cluster do IBM Cloud Private-CE
- Removendo um nó do cluster do IBM Cloud Private
- Removendo um nó do cluster do IBM Cloud Private-CE
- Removendo um nó do cluster do IBM Cloud Private não responsivo
- Mudando o endereço IP ou o nome do host de um nó do cluster do IBM Cloud Private

Incluindo um IBM Cloud Private nó do cluster

Inclua os nós do trabalhador, de proxy, de gerenciamento, do orientador de vulnerabilidade e do grupo de host customizado no cluster do IBM® Cloud Private.

- Preparando o novo nó para instalação
- Preparando um novo nó arch para instalação
- Incluindo nós
  - Incluindo um nó do trabalhador
  - Incluindo um nó de gerenciamento
  - Incluindo um nó do proxy
  - Incluindo um Nó do Orientador de Vulnerabilidade
  - Incluindo um grupo de hosts

Preparando o novo nó para instalação

Conclua as etapas a seguir no novo nó:

1. Assegure-se de que todas as portas padrão estejam abertas, mas não estejam em uso. Nenhuma regra de firewall deve bloquear essas portas. Durante a instalação, o instalador também confirma que essas portas estão abertas. Para obter mais informações sobre as portas padrão do IBM Cloud Private, consulte Portas padrão.

   Para verificar manualmente se uma porta está aberta e disponível, é possível executar um dos dois comandos a seguir, onde `port_numbers` representa a porta TCP/UDP ou as portas a serem verificadas:

   - Execute o `ss` comando:

     ```
     ss -tnlp | awk '{print $4}'| egrep -w "<port_numbers>"
     ```

     Se a porta não está em uso, a saída está vazia. Se a porta está em uso, a saída é exibida como no exemplo a seguir:

     ```
     # ss -tnlp | awk '{print $4}' | egrep -w "8001|8500|3306"
     :::8001 :::3306 :::8500
     ```

   - Ou, se você instalou os utilitários de rede, execute o comando `netstat`:

     ```
     netstat -tnlp | awk '{print $4}' | egrep -w "<port_numbers>"
     ```

     Se a porta está em uso, a saída é exibida como no exemplo a seguir:

     ```
     # netstat -tnlp | awk '{print $4}' | egrep -w "8001|8500|3306"
     :::8001 :::3306 :::8500
     ```

     Os números de porta devem ser separados com o caractere `|`. Consulte o seguinte exemplo:

```
netstat -tnlp | awk '{print $4}'| egrep -w 8101|8500|3306|
```

2. Configure os utilitários do DNS para certificar-se de que o nome do host e o nome completo do domínio (FQDN) do nó recém-incluído sejam resolvidos dentro do cluster.

3. Assegure a conectividade de rede entre o novo nó e todos os outros nós em seu cluster.
4. Sincronize o clock do novo nó com o restante dos nós do cluster. Para sincronizar os seus clocks, é possível usar o protocolo de tempo de rede (NTP). Para obter mais informações sobre como configurar o NTP, consulte a documentação do usuário para o seu sistema operacional.

5. No novo nó, confirme se uma versão suportada do Python está instalada. Os Python versões 2.6 a 2.9.x e 3.5 ou mais recente são suportados.

   ```
   python --version
   ```

6. Assegure-se de que um cliente SSH esteja instalado no novo nó.

7. Se você usar autenticação de chave pública SSH para criar a conexão segura entre seus nós de cluster, inclua a chave pública SSH no novo nó. No nó de inicialização, inclua a chave pública SSH no nó executando o seguinte comando:

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub <user>@<node_ip_address>
   ```

   Em que `<user>` é o nome de usuário para o nó e `<node_ip_address>` é o endereço IP desse nó.

8. Se você instalar manualmente o Docker em seus nós não de inicialização, instale o Docker no novo nó. Consulte Instalando o Docker no IBM Cloud Private.

Preparando um novo nó arch para instalação

Se o novo nó for um nó arch, conclua as etapas a seguir no novo nó antes de incluir o nó em seu cluster. Essas etapas são necessárias, além das etapas anteriores para preparação de um novo nó. Não é necessário concluir essas etapas adicionais ao incluir um nó arch existente.

Por exemplo, se seu cluster incluir um nó arch `amd64` e você desejar incluir um novo nó arch `ppc64le`, será necessário concluir estas etapas para primeiro preparar o novo nó arch.

1. Copie o pacote off-line do novo nó arch para o diretório `/<installation_directory>/cluster/images`.
2. Execute o comando a seguir para enviar por push as novas imagens do nó arch e para construir as imagens multi-arch:

   ```
   docker run -e LICENSE=accept --net=host \
   -v "$(pwd)":/installer/cluster \
   ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee multi-arch-image
   ```

Incluindo Nós

Conclua as etapas a seguir no nó de inicialização que foi usado para instalar seu cluster.

1. Mude para o diretório `cluster` dentro de seu diretório de instalação do IBM Cloud Private.

   ```
   cd /<installation_directory>/cluster
   ```

2. Certifique-se de que o instalador para a plataforma na qual o novo nó é executado esteja disponível no diretório `/<installation_directory>/cluster/images`.

   - Para um nó do Linux®, é necessário o arquivo `ibm-cloud-private-x86_64-3.2.0.tar.gz` ou `ibm-cp-app-mod-x86_64-3.2.0.tar.gz`.
   - Para um nó do Linux® on Power® (ppc64le), é necessário o arquivo `ibm-cloud-private-ppc64le-3.2.0.tar.gz` ou `ibm-cp-app-mod-ppc64le-3.2.0.tar.gz`.
   - Para um nó do trabalhador do IBM® Z, é necessário o arquivo `ibm-cloud-private-s390x-3.2.0.tar.gz`.

3. Inclua o novo nó.

   - Para nós do trabalhador, consulte Incluindo um nó do trabalhador.
   - Para nós de gerenciamento, consulte Incluindo um nó de gerenciamento.
   - Para nós do proxy, consulte Incluindo um nó do proxy.
   - Para os nós do orientador de vulnerabilidade, consulte Incluindo um nó do orientador de vulnerabilidade.
   - Para um grupo de hosts, consulte Incluindo um grupo de hosts.

Incluindo um nó do trabalhador

Nota: Para incluir um nó do IBM Z em seu cluster, inclua o endereço IP para o nó do trabalhador Z no arquivo `/<installation_directory>/hosts`.

Para incluir um nó do trabalhador, execute o comando apropriado com base nos nós de trabalho a serem incluídos:

- Para incluir nós do Linux x86_64, execute este comando:

  ```
  docker run -e LICENSE=accept --net=host \
  -v "$(pwd)":/installer/cluster \
  ibmcom/icp-inception-amd64:3.2.0-ee worker -l \
  ip_address_workernode1,ip_address_workernode2
  ```

- Para incluir nós do Linux on Power (ppc64le), execute este comando:

  ```
  docker run -e LICENSE=accept --net=host \
      -v "$(pwd)":/installer/cluster \
      ibmcom/icp-inception-ppc64le:3.2.0-ee worker -l \
      ip_address_workernode1,ip_address_workernode2
  ```

- Para incluir nós do Linux® on IBM® Z and LinuxONE, execute este comando:

  ```
  docker run -e LICENSE=accept --net=host \
  -v "$(pwd)":/installer/cluster \
  ibmcom/icp-inception-s390x:3.2.0-ee worker -l \
  ip_address_workernode1,ip_address_workernode2
  ```

  Nesse comando, `ip_address_workernode1` e `ip_address_workernode2` são endereços IP de novos nós do trabalhador. Ao executar esse comando, os endereços IP especificados são incluídos nos arquivos host.

Incluindo um nó de gerenciamento

Para incluir um nó de gerenciamento, execute o comando a seguir:

```
docker run -e LICENSE=accept --net=host \
-v "$(pwd)":/installer/cluster \
ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee management -l \
ip_address_managementnode1,ip_address_managementnode2
```

Nesse comando, `ip_address_managementnode1` e `ip_address_managementnode2` são endereços IP dos novos nós de gerenciamento. Ao executar esse comando, os endereços IP especificados são incluídos nos arquivos host.

Incluindo um nó do proxy

Esse procedimento é suportado apenas em ambientes de HA do proxy.

Para incluir um nó do proxy, execute o comando a seguir:

```
docker run -e LICENSE=accept --net=host \
-v "$(pwd)":/installer/cluster \
ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee proxy -l \
ip_address_proxynode1,ip_address_proxynode2
```

Nesse comando, `ip_address_proxynode1` e `ip_address_proxynode2` são endereços IP de novos nós do proxy. Ao executar esse comando, os endereços IP especificados são incluídos nos arquivos host.

Incluindo um Nó do Orientador de Vulnerabilidade

Para incluir um nó do orientador de vulnerabilidade, execute o comando a seguir:

```
docker run --rm -t -e LICENSE=accept --net=host -v \
$(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee va -l
\
ip_address_vanode1,ip_address_vanode2
```

Neste comando, `ip_address_vanode1` e `ip_address_vanode2` são os endereços IP de novos nós do orientador de vulnerabilidade. Ao executar esse comando, os endereços IP especificados são incluídos nos arquivos host.

Nota: o serviço do Vulnerability Advisor não é ativado no nó depois de incluí-lo em seu cluster. Para obter mais informações, consulte Ativando e desativando os serviços de gerenciamento do IBM Cloud Private.

Incluindo um grupo de hosts

Nós do grupo de host são nós do trabalhador.

1. Assegure-se de que o grupo de hosts esteja definido no arquivo host. Consulte Configurando as funções do nó nos arquivos host.

2. Inclua o grupo de hosts.

   - Para configurar um grupo de hosts, execute o comando a seguir:

     ```
     docker run -e LICENSE=accept --net=host \
     -v "$(pwd)":/installer/cluster \
     ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee hostgroup -l
     [hostgroup-name]
     ```

     Nota: se desejar instalar múltiplos grupos de hosts em um único comando, omita a opção `-l`.

   - Para incluir um host específico em um grupo de hosts, execute o comando a seguir:

     ```
     docker run -e LICENSE=accept --net=host \
     -v "$(pwd)":/installer/cluster \
     ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee hostgroup -l \
     ip_address_hostgroupnode1,ip_address_hostgroupnode2
     ```

     Nesse comando, `ip_address_hostgroupnode1` e `ip_address_hostgroupnode2` são endereços IP dos novos nós do grupo de hosts.

Incluindo um IBM Cloud Private-CE nó do cluster

Inclua os nós do trabalhador e do grupo de hosts customizados em seu cluster do IBM® Cloud Private-CE (Community Edition).

Preparando o nó para instalação

1. Assegure-se de que todas as portas padrão estejam abertas, mas não estejam em uso. Nenhuma regra de firewall deve bloquear essas portas. Para obter mais informações sobre as portas padrão do IBM® Cloud Private, consulte Portas padrão. Para verificar se uma porta está aberta, execute o comando a seguir:

   ```
   ssh -p <port_number> localhost
   ```

   Em que `port_number` é o número da porta a ser verificado. Se o comando retornar uma saída, a porta estará em uso. Se uma porta estiver em uso, é possível executar o comando `ss -nlp | grep <port_number>` para ver qual serviço está utilizando-a.

2. Configure o arquivo `/etc/hosts` em cada nó em seu cluster.

   1. Inclua os endereços IP e nomes de host para todos os nós no arquivo `/etc/hosts` em cada nó.
      - **Importante**: certifique-se de que o nome do host seja listado pelo endereço IP para o host local. Não é possível listar o nome do host pelo endereço de loopback, `127.0.0.1`.
      - Os nomes do host no arquivo `/etc/hosts` não podem conter letras maiúsculas.
      - Se seu cluster contiver um único nó, será necessário listar o endereço IP e o nome do host.
   2. Comente a linha do arquivo que inicia com `127.0.1.1` e `::1 localhost`.

   O arquivo `/etc/hosts` para um cluster que contém um nó principal, um nó do proxy e dois nós do trabalhador é semelhante ao código a seguir:

   ```
   127.0.0.1       localhost
   # 127.0.1.1     <host_name>
   # The following lines are desirable for IPv6 capable hosts
   #::1     localhost ip6-localhost ip6-loopback
   ff02::1 ip6-allnodes
   ff02::2 ip6-allrouters
   ```

```
<master_node_IP_address> <master_node_host_name>
<worker_node_1_IP_address> <worker_node_1_host_name>
<worker_node_2_IP_address> <worker_node_2_IP_host_name>
<proxy_node_IP_address> <proxy_node_host_name>
```

3. Assegure a conectividade de rede entre todos os nós em seu cluster. Confirme se cada nó está conectado a todos os outros nós no cluster.

4. Sincronize o clock do novo nó com o restante dos nós do cluster. Para sincronizar os seus clocks, é possível usar o protocolo de tempo de rede (NTP). Para obter mais informações sobre como configurar o NTP, consulte a documentação do usuário para o seu sistema operacional.

5. No novo nó, confirme se uma versão suportada do Python está instalada. Os Python versões 2.6 a 2.9.x e 3.5 ou mais recente são suportados.

   ```
   python --version
   ```

6. Assegure-se de que um cliente SSH esteja instalado no novo nó.

7. Se você usar SSH para criar a conexão segura entre os nós do cluster, inclua a chave SSH no novo nó.

   1. No nó de inicialização, inclua a chave pública SSH no nó.

      ```
      ssh-copy-id -i ~/.ssh/id_rsa.pub <user>@<node_ip_address>
      ```

      Em que `<user>` é o nome de usuário para o nó e `<node_ip_address>` é o endereço IP desse nó.

   2. Efetue login no nó.
   3. Reinicie o serviço SSH:

      ```
      sudo systemctl restart sshd
      ```

8. Se você instalar manualmente o Docker em seus nós não de inicialização, instale o Docker no novo nó. Consulte Instalando o Docker no IBM Cloud Private
9. Mude para o diretório `cluster` dentro de seu diretório de instalação do IBM Cloud Private-CE.

   ```
   cd /<installation_directory>/cluster
   ```

Incluindo Nós

Incluindo um nó do trabalhador

Nota: Para incluir um nó do IBM® Z em seu cluster, inclua o endereço IP para o nó do trabalhador Z no arquivo `/<installation_directory>/hosts`.

Para incluir um nó do trabalhador, execute o comando a seguir:

```
docker run -e LICENSE=accept --net=host \
        -v "$(pwd)":/installer/cluster \
        ibmcom/icp-inception:3.2.0 worker -l \
        ip_address_workernode1,ip_address_workernode2
```

Nesse comando, `ip_address_workernode1` e `ip_address_workernode2` são endereços IP de novos nós do trabalhador. Ao executar esse comando, os endereços IP especificados são incluídos nos arquivos host.

Incluindo um grupo de hosts

Os nós do grupo de hosts são um conjunto de nós do trabalhador que estão reservados para executar aplicativos ou processos específicos.

1. Assegure-se de criar um nome para o grupo de hosts e de incluir os IPs para cada nó no grupo. Consulte Definindo grupos de hosts customizados.

2. Inclua o grupo de hosts.

   - Para configurar um grupo de hosts, execute o comando a seguir:

     ```
     docker run -e LICENSE=accept --net=host \
       -v "$(pwd)":/installer/cluster \
       ibmcom/icp-inception:3.2.0 hostgroup -l [hostgroup-name]
     ```

Nota: se desejar instalar múltiplos grupos de hosts em um único comando, omita a opção `-l`.

- Para incluir um host específico em um grupo de hosts, execute o comando a seguir:

```
docker run -e LICENSE=accept --net=host \
    -v "$(pwd)":/installer/cluster \
    ibmcom/icp-inception:3.2.0 hostgroup -l \
    ip_address_hostgroupnode1,ip_address_hostgroupnode2
```

Nesse comando, `ip_address_hostgroupnode1` e `ip_address_hostgroupnode2` são endereços IP dos novos nós do grupo de hosts.

Removendo um nó do cluster do IBM Cloud Private

Remova um nó de cluster do cluster do IBM® Cloud Private.

1. Mude o diretório `cluster` dentro do seu diretório de instalação do IBM Cloud Private:

```
cd /<installation_directory>/cluster
```

2. Se desejar remover um nó do trabalhador do IBM® Z de seu cluster, a pasta `/<installation_directory>/cluster/images` deve conter o arquivo `ibm-cloud-private-s390x-3.2.0.tar.gz`.

3. Para remover um nó, execute o comando a seguir:

```
docker run -e LICENSE=accept --net=host \
-v "$(pwd)":/installer/cluster \
ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee uninstall -l \
ip_address_clusternode1,ip_address_clusternode2
```

Nesse comando, `ip_address_clusternode1` e `ip_address_clusternode2` são endereços IP de nós que estão listados nos arquivos host. Esses nós podem ser uma combinação de nós do trabalhador, do proxy ou de gerenciamento.

Removendo um nó do cluster do IBM Cloud Private-CE

Remova um nó de seu cluster do IBM® Cloud Private-CE (Community Edition) .

Para o cluster do IBM® Cloud Private-CE, é possível remover somente nós do trabalhador.

1. Mude para o diretório `cluster` dentro de seu diretório de instalação do IBM Cloud Private-CE.

```
cd /<installation_directory>/cluster
```

2. Remova o nó de seu cluster.

```
docker run -e LICENSE=accept --net=host \
  -v "$(pwd)":/installer/cluster \
  ibmcom/icp-inception:3.2.0 uninstall -l \
  ip_address_clusternode1,ip_address_clusternode2
```

Nesse comando, `ip_address_clusternode1` e `ip_address_clusternode2` são endereços IP de nós que estão listados nos arquivos host. Esses nós devem ser nós do trabalhador.

Os nós são removidos de seu cluster e seus endereços IP são removidos dos arquivos host.

Removendo um nó do cluster não responsivo do IBM Cloud Private

Remova um nó não responsivo de seu cluster do IBM® Cloud Private.

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Obtenha uma lista de seus nós do cluster.

```
kubectl get nodes
```

O seguinte é uma saída de amostra:

```
NAME             STATUS    ROLES                          AGE       VERSION
172.16.151.126   Ready     etcd,management,master,proxy   39d       v1.11.1+icp-ee
172.16.151.182   NotReady  worker                         39d       v1.11.1+icp-ee
<<<<<<< unresponsive node
172.16.155.135   Ready     worker                         39d       v1.11.1+icp-ee
```

3. Exclua o nó não responsivo de seu cluster.

   ```
   kubectl delete node <node-IP-address>
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl delete node 172.16.151.182
   node "172.16.151.182" deleted
   ```

4. Remova o endereço IP do nó não responsivo do arquivo `<installation_directory>/cluster/hosts`.

Mudando o endereço IP ou o nome do host de um nó do cluster do IBM Cloud Private

Não é possível mudar o endereço IP ou o nome do host de um nó depois que seu cluster do IBM® Cloud Private é instalado.

No entanto, é possível concluir as etapas a seguir para mudar o endereço IP ou o nome do host de um nó em seu cluster:

1. Remova o nó do cluster. Veja Removendo um nó do cluster do IBM Cloud Private ou Removendo um nó do cluster do IBM Cloud Private-CE.
2. Mude o endereço IP ou o nome do host do nó, conforme necessário.
3. Inclua de volta o nó em seu cluster. Veja Incluindo um nó do cluster do IBM Cloud Private ou Incluindo um nó do cluster do IBM Cloud Private-CE.

Nota: não é possível remover e incluir de volta um nó principal depois que seu cluster do IBM® Cloud Private é instalado. Portanto, não é possível mudar o endereço IP ou o nome do host de um nó principal que faz parte de um cluster instalado.

Recursos

Ao criar um aplicativo, opcionalmente, é possível especificar quantos recursos de unidade central de processamento (CPU), memória (RAM) e unidade de processamento gráfico (GPU) cada contêiner pode usar.

CPU, memória e GPU são referidos coletivamente como recursos de cálculo ou apenas recursos. Os recursos de cálculo são quantidades mensuráveis de recursos de cluster que podem ser solicitados, alocados e usados.

Se você especificar limites de recursos para os contêineres, a distribuição de contêineres de aplicativo em nós do trabalhador poderá ser otimizada.

Para obter informações adicionais sobre como gerenciar recursos de cálculo, consulte Conceitos do Kubernetes.

O uso de memória real, CPU ou GPU de um aplicativo em um nó varia ao longo do tempo. O uso total de memória, CPU ou GPU em um nó também varia ao longo do tempo. É possível monitorar o uso de memória e de CPU do cluster na página do painel da console de gerenciamento. Também é possível monitorar o uso de recursos de cada nó na página do nó. O uso real do recurso relatado inclui as estatísticas de todos os contêineres no nó. Esses números de uso incluem os recursos que são usados pelo IBM® Cloud Private e os recursos que são usados por outros processos, tal como `docker run`.

- Suporte à GPU do Nvidia
- Configurando um nó do trabalhador de GPU

Suporte à GPU do Nvidia

É possível designar os recursos da unidade de processamento gráfico (GPU) em seu cluster do IBM® Cloud Private para aplicativos e tarefas.

A GPU é o poder de processamento por trás de novas cargas de trabalho que estão abrindo caminho em campos como aprendizado de máquina e sistemas de computação de alto desempenho. Começando com o Kubernetes 1.6.1, agora é possível gerenciar GPU de uma forma semelhante àquela de outros recursos como CPU e memória.

As restrições a seguir se aplicam ao uso da GPU no cluster do IBM Cloud Private:

- Deve-se representar recursos de GPU com valores de número inteiro positivo que indicam o número de núcleos físicos da GPU. A alocação de núcleo de GPU parcial não é suportada.
- Deve-se usar o driver de GPU versão 352 ou mais recente. As versões mais antigas podem não ser compatíveis com o IBM Cloud Private.
- É possível designar recursos de GPU a contêineres específicos em um Pod. Não é possível compartilhar os recursos de GPU com outros contêineres no Pod.
- É possível declarar apenas limites de recursos de GPU, não solicitações.

Para obter mais informações sobre como implementar um aplicativo com os recursos de GPU conectados, consulte Criando uma implementação com recursos de GPU conectados.

Configurando um nó do trabalhador de GPU

Verifique se os nós do trabalhador GPU estão prontos para implementação.

Preparando os nós GPU

Conclua estas etapas em todos os nós GPU:

1. Configure um repositório.
   - Para o Red Hat Enterprise Linux, execute estes comandos:

     a. Configure um repositório do RHEL. Use as credenciais do gerenciador de assinaturas para registrar.

     ```
     subscription-manager register --username <username> --password <password>
     ```

     b. Configure um repositório do Extra Packages for Enterprise Linux (EPEL).

     ```
     yum install http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
     ```

   - Para Ubuntu, o repositório está disponível por padrão.
2. Instale o driver NVIDIA.

   a. Configure o repositório local do driver NVIDIA.
   Primeiro, faça download do pacote da versão correta do driver para o GPU a partir de Downloads do driver NVIDIA. Em seguida, instale o pacote executando o seguinte comando:

   - Para o RHEL, execute o seguinte comando:

     ```
     yum localinstall <package-name>.rpm -y
     ```

   - Para Ubuntu, execute o comando a seguir:

     ```
     dpkg -i <package-name>.deb
     ```

   b. Instale o driver.

   - Para o RHEL, execute este comando:

     ```
     yum install nvidia-kmod nvidia-driver-NVML nvidia-persistenced -y
     ```

   - Para Ubuntu, execute este comando:

     ```
     apt install nvidia-driver-<version>
     ```

3. Remova as regras udev de memória hot-plug existentes.
   Comente a regra hotadd de memória a partir das regras udev.

   ```
   # Memory hotadd request
   #SUBSYSTEM!="memory", ACTION!="add", GOTO="memory_hotplug_end"
   #PROGRAM="/bin/uname -p", RESULT=="s390*", GOTO="memory_hotplug_end"
   ```

   O arquivo de regras udev está disponível nos seguintes locais:

   - Para o RHEL, o local e o nome do arquivo são `/lib/udev/rules.d/40-redhat.rules`.

- Para Ubuntu, o local e o nome do arquivo são `/lib/udev/rules.d/40-vm-hotadd.rules`.

4. Ative o serviço `nvidia-persistenced`.

   ```
   systemctl enable nvidia-persistenced
   ```

5. Reinicie o nó.

6. Valide a configuração executando o seguinte comando:

   ```
   nvidia-smi
   ```

Atualizando a versão do driver de GPU

É possível atualizar seu driver de GPU antes ou após a instalação do IBM® Cloud Private.

Importante: o NVIDIA Container Runtime não é usado pelo ambiente do IBM Cloud Private. O NVIDIA Container Runtime e suas dependências não devem ser instalados em nenhum nó GPU do IBM Cloud Private.

1. Para atualizar sua versão do driver de GPU. Consulte http://www.nvidia.com/Download/index.aspx.

2. Depois de atualizar a versão do driver GPU, reinicie o Kubelet para permitir que o Kubernetes selecione as mudanças no driver GPU.

   ```
   systemctl restart kubelet
   ```

Verificando se os nós estão prontos para implementação

Deve-se executar essas etapas de verificação a partir do nó do trabalhador no qual o driver Nvidia GPU está instalado.

1. Verifique se o Nvidia está funcionando.

   ```
   nvidia-smi
   ```

   A saída se assemelha ao código a seguir:

   ```
   Thu Nov  9 16:44:28 2017
   +-----------------------------------------------------------------------------+
   | NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
   |-------------------------------+----------------------+----------------------+
   | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
   | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
   |===============================+======================+======================|
   |   0  Tesla K80           Off  | 0000:08:00.0     Off |                    0 |
   | N/A   47C    P8    26W / 149W |      0MiB / 11439MiB |      0%      Default |
   +-------------------------------+----------------------+----------------------+
   |   1  Tesla K80           Off  | 0000:09:00.0     Off |                    0 |
   | N/A   36C    P8    31W / 149W |      0MiB / 11439MiB |      0%      Default |
   +-------------------------------+----------------------+----------------------+

   +-----------------------------------------------------------------------------+
   | Processes:                                                       GPU Memory |
   |  GPU       PID  Type  Process name                               Usage      |
   |=============================================================================|
   |  No running processes found                                                 |
   +-----------------------------------------------------------------------------+
   ```

   Se uma mensagem de erro for retornada, reinstale o driver de GPU no nó. Consulte http://www.nvidia.com/Download/index.aspx

2. Assegure-se de que as pastas do driver de GPU estejam disponíveis.

   1. Verifique se a pasta `/var/lib/kubelet/device-plugins/nvidia-driver` existe.
   2. Verifique se há pelo menos duas pastas sob a pasta `/var/lib/kubelet/device-plugins/nvidia-driver`. Os nomes das pastas são `init` e `<driver-version-number>`.

   Se alguma pasta não existir, assegure-se de que os drivers de GPU estejam instalados corretamente. Em seguida, execute os comandos a seguir:

   1. Exclua a pasta que tem os arquivos do driver.

```
rm -rf /var/lib/kubelet/device-plugins/nvidia-driver
```

2. Reinicie o serviço kubelet.

```
systemctl restart kubelet
```

3. Verifique se os recursos de GPU estão disponíveis para uso do Kubernetes.

```
nós descrevemos nós de kubectl
```

Na saída de comando, o nó com GPU deve ter as entradas a seguir:

```
Capacity:
[snip]
nvidia.com/gpu:     2
[snip]
Allocatable:
[snip]
nvidia.com/gpu:     2
```

Se você não vir entradas `nvidia.com/gpu` para o nó que tem GPUs do NVIDIA, a causa provável é uma instalação incorreta do driver de GPU. Talvez seja necessário reinstalar o driver de GPU.

Agora você está pronto para implementar aplicativos que usam recursos de GPU em seu nó do trabalhador. Consulte Criando uma implementação com recursos de GPU conectados.

Gerenciando políticas

Crie e mantenha políticas de implementação. As políticas de implementação escalam automaticamente o número de réplicas de implementação.

O uso de CPU de implementações conduz o ajuste de escala de réplica em um cluster.

Aumento de capacidade automático

Um aumento de capacidade é acionado durante o aumento da demanda de CPU, como quando um servidor experimenta altos níveis de solicitações. A condição a seguir aciona automaticamente uma operação de aumento de capacidade:

```
CURRENT_CPU_UT >1.1*TARGET_CPU_UT
```

Quando uma implementação tem a capacidade aumentada, seus valores de parâmetro `DESIRED`, `CURRENT` e `READY` aumentam. É possível verificar esses valores na página inicial de implementações.

Diminuição de capacidade automática

Uma diminuição de capacidade é acionada quando um servidor entra em um estado inativo ou a demanda de CPU diminui. A condição a seguir aciona automaticamente uma operação de diminuição de capacidade:

```
CURRENT_CPU_UT < 0.9*TARGET_CPU_UT
```

Quando uma implementação tem sua capacidade diminuída, seus valores de parâmetro DESIRED, CURRENT e READY diminuem. É possível verificar esses valores na página inicial de implementações.

- Criando uma política de implementação
- Atualizando uma política de implementação
- Removendo uma política de implementação

Criando uma política de implementação

Use políticas para especificar o ajuste de escala e a alocação de recurso para um pod.

Todos os usuários podem definir uma nova política no namespace. Para administradores, a política é criada, por padrão, no namespace de administrador.

1. No menu de navegação, clique em **Configuração** > **Políticas de ajuste de escala**.
2. Clique em **Criar política**.

3. Forneça os detalhes da política. Forneça valores individuais na janela Criar política.
    1. Forneça a política **Nome**.
    2. No campo **Destino da escala**, insira o nome do aplicativo ao qual a política se aplica.
    3. Forneça o valor **Mínimo de replicações**. O valor padrão é 1.
    4. Insira o valor **Máximo de replicações**. Esse valor é o número máximo de replicações que são permitidas durante um aumento da capacidade.
    5. Insira o valor **CPU de destino**. Esse valor é a porcentagem da CPU disponível que o aplicativo pode alocar. Se você não especificar os limites de recursos para seu contêiner, o valor será configurado como ilimitado.
4. Clique em **Criar**.

Uma nova política é exibida na página inicial Políticas de ajuste de escala.

Para visualizar informações detalhadas de política, clique no nome da política.

Atualizando uma política de implementação

Atualizar uma política de implementação.

1. No menu de navegação, clique em **Configuração** > **Políticas de ajuste de escala**.
2. Selecione **Ação** > **Editar**. O arquivo JSON da política é exibido.
3. Atualize as propriedades da política.
4. Clique em **Enviar**.

Removendo uma política de implementação

Remover uma política de implementação.

1. No menu de navegação, clique em **Configuração** > **Políticas de ajuste de escala**.
2. Para a política que você deseja remover, selecione **Ação** > **Remover**. Uma caixa de diálogo de confirmação é exibida.
3. Clique em **Remover política**.

A política é removida da lista de políticas de implementação.

Gerenciando segredos

Use Segredos para armazenar definições de configuração confidenciais como senhas e chaves API.

É possível usar Segredos para armazenar informações confidenciais que podem ser usadas por diversos contêineres ou implementações. Os Segredos podem ser referenciados por contêineres ou implementações que estão em execução apenas no mesmo namespace.

Para obter mais informações sobre Segredos, consulte Segredos do Kubernetes.

Para visualizar uma lista de todos os Segredos no cluster, no menu de navegação, clique em **Configuração** > **Segredos**. Nessa visualização, também é possível filtrar Segredos por seus namespaces.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando segredos

Criando segredos na linha de comandos

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Alterne para o namespace no qual você deseja criar o Segredo.

```
kubectl config set-context <cluster_name>-context --user=<user_name> --namespace=
<namespace_name>
```

Em que `<cluster_name>` é o nome do cluster conforme definido em ConfigMap de configuração de cluster.

3. Execute o comando `kubectl` para o tipo de Segredo que você deseja criar.
   - Para criar Segredos para uso com um registro do Docker (segredos Dockercfg), use o comando `kubectl create secret docker-registry` na linha de comandos. Consulte Criando imagePullSecrets para um namespace específico e Comando kubectl create secret docker-registry.
   - Para criar Segredos em um arquivo ou diretório local, use o comando `kubectl create secret generic` na linha de comandos. Veja o Comando kubectl create secret generic.
   - Para criar Segredos por meio de pares de chaves públicas/privadas, use o comando `kubectl create secret tls` na linha de comandos. Veja o Comando kubectl create secret tls.

Criando segredos por meio da console de gerenciamento

É possível usar a console de gerenciamento do IBM® Cloud Private para criar Segredos por meio de valores literais. Esses valores devem ser codificados em base64.

1. Na linha de comandos, codifique seus valores de dados em `base64`. Deve-se usar a opção `-n` para assegurar que uma nova linha de caracteres finais `(\n)` não seja anexada na sequência.

   ```
   echo -n "admin" | base64
   ```

   A saída se assemelha ao código a seguir:

   ```
   YWRtaW4=
   ```

2. No menu de navegação, clique em **Configuração** > **Segredos**.
3. No menu suspenso, selecione um namespace. Se um namespace não estiver selecionado, o Segredo será criado no namespace `default`.
4. Clique em **Criar segredo**.
5. Forneça os detalhes para seu Segredo.
   - Na guia **Geral**, forneça um nome e tipo para o seu Segredo. Se um tipo não for especificado, um tipo padrão **Opaco** será designado.
   - Na guia **Dados**, insira os detalhes de configuração para seu segredo como pares chave-valor. Os valores devem ser codificados em base64.
6. Clique em **Criar**.

Atualizando um segredo por meio da console de gerenciamento

É possível atualizar segredos que são criados na linha de comandos ou na console de gerenciamento.

1. No menu de navegação, clique em **Configuração** > **Segredos**.
2. Para o Segredo que você deseja modificar, selecione **Ação** > **Editar**. O arquivo JSON do Segredo é exibido.
3. Atualize as propriedades.
4. Clique em **Enviar**. O Segredo é atualizado.

Segredos de criptografia

Antes de começar, certifique-se de que as tarefas a seguir sejam concluídas:

- O cluster do IBM® Cloud Private deve estar pronto.
- Você deve configurar a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

Configurando a Crip

Execute estes comandos em um nó principal em seu cluster.

1. Gere uma chave aleatória de 32 bytes e codifique-a em `base64`. Execute o comando a seguir:

   ```
   head -c 32 /dev/urandom | base64
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

```
7WU + F6l4JQwgiGO99CAJwHanlb2pPxwfyZpcnsPhp8k=
```

2. Crie um arquivo de configuração `encryption-config.yaml` e coloque-o na pasta `/etc/cfc/conf/`. Use a chave codificada `base64` que você gerou na etapa anterior como o `secret`:

```
kind: EncryptionConfiguration
apiVersion: v1
resources:
 - resources:
   - secrets
   providers:
   - aescbc:
       keys:
       - name: key1
         secret: <base64-encoded Secret>
   - identity: {}
```

   A seguir está um arquivo de configuração de exemplo:

```
kind: EncryptionConfiguration
apiVersion: apiserver.config.k8s.io/v1
resources:
 - resources:
   - secrets
   providers:
   - aescbc:
       keys:
       - name: key1
         secret: <base64-encoded Secret>
   - identity: {}
```

3. Em um cluster de alta disponibilidade (HA), copie o arquivo `/etc/cfc/conf/encryption-config.yaml` para todos os outros nós principais no mesmo diretório.

4. Em todos os nós principais, configure o sinalização `--encryption-provider-provider-config` em `/etc/cfc/pods/master.json` no `kube-apiserver` para apontar para o local do arquivo de configuração.

   1. Faça backup do arquivo manifest antes de editá-lo.

      ```
      cp /etc/cfc/pods/master.json ~/master.json.bak
      ```

   2. Copie o arquivo `/etc/cfc/pods/master.json` para outro local. **Nota:** o Kubelet não suporta a edição de um arquivo manifest de pod estático usando um editor. Para mudar o conteúdo do arquivo, é possível copiar e sobrescrever o arquivo.

      ```
      cp /etc/cfc/pods/master.json /tmp
      ```

   3. Edite o arquivo `master.json` para incluir a opção `--experimental-encryption-provider-config` para o kube-apiserver.

      ```
      "name": "apiserver",
             "image": "ibm1.mixhub.cn:8500/ibmcom/hyperkube:v1.11.1-ee",
             "imagePullPolicy": "IfNotPresent",
             "command": [
                    ......
                    "--experimental-encryption-provider-config=/etc/cfc/conf/encryption-
      config.yaml"
              ]
      ```

   4. Sobrescreva o arquivo manifest de pod estático original que está no diretório `/etc/cfc/pods/`.

      ```
      cp /tmp/master.json /etc/cfc/pods/
      ```

5. Aguarde o `apiserver` reiniciar.

```
docker ps | grep apiserver
```

   A saída pode ser semelhante ao conteúdo a seguir:

```
ef72af905d72     20df5d4fd446            "/hyperkube apiserve…"   About an hour ago   Up 27
minutes    \
         k8s_apiserver_k8s-master-172.29.215.1_kube-system_09a7e75fbf8bcfa2cc56478897bf8898_0
```

Identificando os dados que precisam ser criptografados

Antes de criptografar os segredos, verifique o tipo de dados secretos que é armazenado em `etcd`. Por exemplo, verifique o segredo `platform-auth-idp-credentials`, conforme mostrado na etapa a seguir:

```
kubectl -n kube-system get secret platform-auth-idp-credentials -o yaml
```

A saída pode ser semelhante ao conteúdo a seguir:

```
apiVersion: v1 data: admin_password: YWRtaW4=
admin_username: YWRtaW4= kind: Secret metadata:
creationTimestamp: 2018-09-21T08:06:07Z name:
platform-auth-idp-credentials namespace: kube-system
resourceVersion: "18766" selfLink:
/api/v1/namespaces/kube-system/secrets/platform-auth-idp-credentials
uid: 31a7a864-bd75-11e8-831a-005056a2e128 type: Opaque

# echo YWRtaW4= | base64 -d
admin
```

Use os comandos da CLI `etcdctl` a seguir para ler o segredo fora de `etcd`.

1. Obtenha o ID do contêiner etcd.

   ```
   docker ps | grep k8s_etcd_k8s-etcd
   ```

   A saída assemelha-se ao conteúdo a seguir:

   ```
   1648a7f65373        e21fb69683f3                           "etcd --name=etcd0 -…"   4 days ago
   Up 4 days                               k8s_etcd_k8s-etcd-172.29.215.1_kube-
   system_103986e50c5d7b82c6532bfe18dd9979_0
   ```

2. Copie o contêiner etcd para outro local.

   ```
   docker cp 1648a7f65373:/usr/local/bin/etcdctl /usr/local/bin/
   ```

3. Crie um alias para conectar-se com o etcd. O `endpoint` é o endereço IP de seu nó principal.

   ```
   alias etcdctl3="ETCDCTL_API=3 etcdctl --endpoints=$(endpoint):4001 --
   cacert=/etc/cfc/conf/etcd/ca.pem --cert=/etc/cfc/conf/etcd/client.pem --
   key=/etc/cfc/conf/etcd/client-key.pem"
   ```

4. Leia o segredo para fora do etcd.

   ```
   etcdctl3 get -w fields /registry/secrets/kube-system/platform-auth-idp-credentials
   ```

   A saída assemelha-se ao conteúdo a seguir:

   ```
   "ClusterID" : 16723368499499280303
   "MemberID" : 15696589318412288662
   "Revision" : 858101
   "RaftTerm" : 5
   "Key" : "/registry/secrets/kube-system/platform-auth-idp-credentials"
   "CreateRevision" : 4595
   "ModRevision" : 850330
   "Version" : 7
   "Value" : "k8s\x00\n\f\n\x02v1\x12\x06Secret\x12\xa4\x01\nf\n\x1dplatform-auth-idp-
   credentials\x12\x00\x1a\vkube-system\"\x00*$31a7a864-bd75-11e8-831a-
   005056a2e1282\x008\x00B\b\b\xef‘\xdd\x05\x10\x00z\x00\x12\x18\n\x0eadmin_password\x12\x06admin\
   x12\x18\n\x0eadmin_username\x12\x06admin\x1a\x06Opaque\x1a\x00\"\x00"
   "Lease" : 0
   "More" : false
   "Count" : 1
   ```

A saída de comando indica que o nome do usuário administrador e a senha padrão são armazenados como texto sem formatação.

Criptografando todos os segredos

Os segredos são criptografados em uma operação de gravação. Portanto, ao atualizar um segredo, o segredo é criptografado.

1. Antes de criptografar, faça backup de todos os segredos em um arquivo.

   ```
   kubectl get secrets --all-namespaces -o json > mysecrets.json
   ```

2. Criptografe todos os segredos que estiverem no armazenamento do etcd.

   ```
   kubectl get secrets --all-namespaces -o json | kubectl replace -f -
   ```

Verificando se os dados estão criptografados

Leia o segredo de etcd para verificar se o segredo está criptografado.

```
etcdctl3 get -w fields /registry/secrets/kube-system/platform-auth-idp-credentials
```

A saída assemelha-se ao conteúdo a seguir:

```
"ClusterID" : 16723368499499280303
"MemberID" : 15696589318412288662
"Revision" : 858394
"RaftTerm" : 5
"Key" : "/registry/secrets/kube-system/platform-auth-idp-credentials"
"CreateRevision" : 4595
"ModRevision" : 858338
"Version" : 8
"Value" : "k8s:enc:aescbc:v1:key1:簫\x1f\xcb\x02Oz\xcdv\xfb\x13=\xab\xaf~\xa9翎
\xe5\xda\xf5u\xc3PY\x9f\x0es\x9cO\xfc\x1d\x8a\x821TLfe\xf5\x87{z\x99\x98\x0ex\xa7\xb5\"H\xc7N\xe8\xb
1\x1cq\x82\xd7\x17\xc4\xcaь<[\x9d\xef]n\x0f{\x87\x10\xd4%\xe7\x8eMm…\x9eIf\xa9\x19\xb1\x9c\xd2(X8\xf
7,\xacF~\xa5A\xee\xed5scto\r\xf7\xde\x01'\xc4E\x97\x16ζak\xba\\\x1c$\x06\x9d\x0f\n\xddkQ}-
\xecd\x1cKﻉ\x8ca_\xa5\x9f$\xff\xd9P\xdbb@\x10\xd2\xe5\x02\xf9\xeeVj\x19j\xc0\x02-
X\xbe\x84\x05;\xad\xe9\xb9\x17\x92(\xfe\x047\xa99\U000fa669L\xe0\xdc\xc3\xd35\xd1;i\xcc>\x15\x94(\xe
6\xb4=>X\x03\xc7"
"Lease" : 0
"More" : false
"Count" : 1
```

A saída de comando indica que o nome do usuário e a senha do administrador padrão agora estão criptografados.

Rotando uma chave de decriptografia

É possível mudar o segredo sem incorrer em um tempo de inatividade em seu cluster.

1. Gerar uma nova chave

   - Para o Linux® e o macOS, execute o comando a seguir:

     ```
     head -c 32 /dev/urandom | base64
     ```

     A saída pode ser semelhante ao conteúdo a seguir:

     ```
     qM4BeDF2CcpNJqTIOzGwkqPaeWm5XgLt6FQJM0KF4ao=
     ```

2. Inclua a chave como uma segunda entrada de chave em seu arquivo de configuração de criptografia que está localizado na pasta `/etc/cfc/conf/`. Inclua a mesma segunda chave em todos os nós principais.

   ```
   kind: EncryptionConfig
   apiVersion: v1
   resources:
   - resources:
     - secrets
     providers:
     - aescbc:
         keys:
         - name: key1
           secret: 7WU+F6l4JQwgiGO99CAJwHanlb2pPxwfyZpcnsPhp8k=
         - name: key2
           secret: qM4BeDF2CcpNJqTIOzGwkqPaeWm5XgLt6FQJM0KF4ao=
     - identity: {}
   ```

3. Em todos os nós principais, reinicie o processo `kube-apiserver` para assegurar que cada nó principal possa decriptografar usando a nova chave.

   ```
   docker stop $(docker ps | grep k8s_apiserver_k8s-master | gawk '{print $1}')
   ```

4. Torne a nova chave a primeira entrada na matriz de chaves, para que ela seja usada para criptografia.

   ```
   kind: EncryptionConfig
   apiVersion: v1
   ```

```
resources:
 - resources:
   - secrets
   providers:
   - aescbc:
       keys:
       - name: key2
         secret: qM4BeDF2CcpNJqTIOzGwkqPaeWm5XgLt6FQJM0KF4ao=
       - name: key1
         secret: 7WU+F6l4JQwgiGO99CAJwHanlb2pPxwfyZpcnsPhp8k=
   - identity: {}
```

5. Em todos os nós principais, reinicie o processo `kube-apiserver` para assegurar que cada nó principal agora criptografa usando a nova chave.

   ```
   docker stop $(docker ps | grep k8s_apiserver_k8s-master | gawk '{print $1}')
   ```

6. Criptografe todos os segredos existentes com a nova chave.

   ```
   kubectl get secrets --all-namespaces -o json | kubectl replace -f -
   ```

7. Leia os novos segredos criptografados do etcd.

   ```
   etcdctl3 get -w fields /registry/secrets/kube-system/platform-auth-idp-credentials
   ```

   A saída se assemelha ao código a seguir:

   ```
   "ClusterID" : 16723684994992803O3
   "MemberID" : 15696589318412288662
   "Revision" : 861046
   "RaftTerm" : 5
   "Key" : "/registry/secrets/kube-system/platform-auth-idp-credentials"
   "CreateRevision" : 4595
   "ModRevision" : 860965
   "Version" : 9
   "Value" :
   "k8s:enc:aescbc:v1:key2:\x15=M\x99y~\x14\x8ee\x82\x85\xf0\x8d}N\x81\xac\xa8b'\xe6\xb7\xcd\xeb\x
   ea\x93;\xaf\x0e\xafV\x16\xf3'G\x10/U\x905\xd0@Z\xe8f\r3\xe0q|\"\x11\xb4\b\x1e\x9b|h\f\x02+\xdd\
   xe6\xa0/\xc0\x12>\x8a\x8f\xa1\x8ac\x8c\xb0\xf0\x18\xc31\ns\xf9+\xbbv\xd5d\xae\x8c\x99z)\x04\xdb
   '\xab\u07b8ᵘ\xcd&!\x0fZ;\x8c\xdc\xd0p\x95LQSz'A\x04n\xbdZ\x84\xb1D\xda\a\x86\xa1\xe7\x064\x1c@\
   x1d\bt\xa6\x97\x0e*\x03\x0e/Vq\x18\x9e\x85\x83\xcaQ\xe6ˇ\x86\x823F\x96
   \xe5z\x89Ы\xad5\xe7\x9b\xe1*\xed¸\x84b\xb8WSa\xe0ƞ\x90\xd3\xdc\x142s\xdb\xce[BE\xb9\xe9\\\xe8\x
   e0\xb2_\x89@,\xb8j"
   "Lease" : 0
   "More" : false
   "Count" : 1
   ```

   A saída indica que o segredo foi criptografado usando a nova chave `k8s:enc:aescbc:v1:key2`.

8. Verifique se o segredo pode ser decriptografado ao usar a nova chave.

   1. Obtenha os dados criptografados.

      ```
      kubectl -n kube-system get secret platform-auth-idp-credentials -o yaml
      ```

      A saída se assemelha ao código a seguir:

      ```
      apiVersion: v1 data: admin_password: YWRtaW4=
      admin_username: YWRtaW4= kind: Secret metadata:
      creationTimestamp: 2018-09-21T08:06:07Z name:
      platform-auth-idp-credentials namespace: kube-system
      resourceVersion: "18766" selfLink:
      /api/v1/namespaces/kube-system/secrets/platform-auth-idp-credentials
      uid: 31a7a864-bd75-11e8-831a-005056a2e128 type: Opaque
      ```

   2. Decriptografe os dados.

      ```
      echo YWRtaW4 = | base64 -d
      ```

      O seguinte é uma saída de amostra:

      ```
      admin
      ```

      A saída indica que o segredo pode ser decriptografado.

9. Remova a chave de decriptografia antiga do arquivo de configuração. Conclua esta etapa somente depois de fazer backup do etcd e de atualizar todos os segredos.

```
kind: EncryptionConfig
apiVersion: v1
resources:
 - resources:
   - secrets
   providers:
   - aescbc:
       keys:
       - name: key2
         secret: qM4BeDF2CcpNJqTIOzGwkqPaeWm5XgLt6FQJM0KF4ao=
   - identity: {}
```

Decriptografando todos os dados

Para desativar a criptografia, atualize o arquivo de configuração `encryption-config.yaml`, que está na pasta `/etc/cfc/conf/`.

1. Coloque o provedor de identidade como a primeira entrada no arquivo de configuração.

```
kind: EncryptionConfig
apiVersion: v1
resources:
 - resources:
   - secrets
   providers:
   - identity: {}
   - aescbc:
       keys:
       - name: key2
         secret: qM4BeDF2CcpNJqTIOzGwkqPaeWm5XgLt6FQJM0KF4ao=
```

2. Em todos os nós principais, reinicie o processo `kube-apiserver`.

```
docker stop $(docker ps | grep k8s_apiserver_k8s-master | gawk '{print $1}')
```

3. Force todos os segredos a serem decriptografados.

```
kubectl get secrets --all-namespaces -o json | kubectl replace -f -
```

4. Verifique se os segredos foram decriptografados.

```
etcdctl3 get -w fields /registry/secrets/kube-system/platform-auth-idp-credentials
```

   A saída se assemelha ao código a seguir:

```
"ClusterID" : 16723368499499280303
"MemberID" : 15696589318412288662
"Revision" : 862012
"RaftTerm" : 5
"Key" : "/registry/secrets/kube-system/platform-auth-idp-credentials"
"CreateRevision" : 4595
"ModRevision" : 861920
"Version" : 10
"Value" : "k8s\x00\n\f\n\x02v1\x12\x06Secret\x12\xa4\x01\nf\n\x1dplatform-auth-idp-
credentials\x12\x00\x1a\vkube-system\"\x00*$31a7a864-bd75-11e8-831a-
005056a2e1282\x008\x00B\b\b\xef'\xdd\x05\x10\x00z\x00\x12\x18\n\x0eadmin_password\x12\x06admin\
x12\x18\n\x0eadmin_username\x12\x06admin\x1a\x06Opaque\x1a\x00\"\x00"
"Lease" : 0
"More" : false
"Count" : 1
```

   A saída de comando indica que o nome do usuário administrador e a senha padrão são armazenados como texto sem formatação.

Reconfigurando Kubelet em um cluster ativo

A configuração dinâmica do Kubelet está disponível como Beta no Kubernetes 1.11.

É possível usar esse recurso no IBM® Cloud Private para mudar a configuração de cada Kubelet em um cluster do Kubernetes em tempo real. Para isso, implemente um ConfigMap e configure cada Nó para usar o ConfigMap.

Para obter mais informações, consulte Reconfigurar o Kubelet do nó em um Cluster em Tempo Real. Certifique-se de ler os avisos.

Siga estas etapas para reconfigurar o Kubelet em um nó em tempo real em seu cluster:

Nota: no IBM Cloud Private, o `--dynamic-config-dir` é configurado como `/etc/cfc/kubelet/kubelet-dynamic-config` por padrão.

1. Gerar um arquivo com a configuração atual
2. Editar o arquivo de configuração
3. Enviar por push o arquivo de configuração para o plano de controle
4. Definir o nó para usar a nova configuração
5. Verifique as atualizações

Antes de iniciar, assegure-se de que a CLI kubectl esteja configurada. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

Gere um arquivo com a configuração atual

Gere um arquivo de configuração que contenha a configuração atual de um nó. Escolha uma das duas maneiras a seguir:

- Use o arquivo `/etc/cfc/kubelet/kubelet-service-config` para criar o primeiro ConfigMap do Kubelet.
- Acesse o terminal `configz` do servidor Kubelet por meio do proxy kubectl. Para obter mais informações, consulte Gerar o arquivo de configuração para obter mais informações.

O arquivo `/etc/cfc/kubelet/kubelet-service-config` é usado nas seções a seguir para atualizar a configuração do Kubelet.

Editar o arquivo de configuração

Conclua estas etapas para editar o arquivo de configuração:

1. Copie `/etc/cfc/kubelet/kubelet-service-config` para `./kubelet-dynamic-config`.

   ```
   cp /etc/cfc/kubelet/kubelet-service-config ./kubelet-dynamic-config
   ```

2. Edite o arquivo `./kubelet-dynamic-config` conforme necessário. Por exemplo, é possível incluir reservas de recursos atualizando as definições de configuração a seguir:

   ```
   systemReserved:
    cpu: "500m"
    memory: "1500Mi"
    ephemeral-storage: "1Gi"
   kubeReserved:
    cpu: "500m"
    memory: "1500Mi"
    ephemeral-storage: "1Gi"
   ```

Para obter uma lista de parâmetros disponíveis, consulte a seção *KubeletConfiguration contém a configuração para o Kubelet* do arquivo de amostra do Kubernetes.

Push the configuration file to the control plane

Create the ConfigMap by pushing the configuration file to the control plane.

```
kubectl -n kube-system create configmap my-node-config --from-file=kubelet=kubelet-dynamic-config --
append-hash -o yaml
```

O resultado se parecerá com o código a seguir:

```
apiVersion: v1
data:
  kubelet: |
    {...}
kind: ConfigMap
metadata:
```

```
  creationTimestamp: 2017-09-14T20:23:33Z
  name: my-node-config-gkt4c2m4b2
  namespace: kube-system
  resourceVersion: "119980"
  selfLink: /api/v1/namespaces/kube-system/configmaps/my-node-config-gkt4c2m4b2
  uid: 946d785e-998a-11e7-a8dd-42010a800006
```

The ConfigMap is created in the `kube-system` namespace. The `--append-hash` option appends a short checksum of the ConfigMap contents to the name.

Set the node to use the new configuration

Store the node name or IP address, and the ConfigMap name to two variables. Consulte os comandos de exemplo a seguir:

```
NODE_NAME=10.10.25.11

CONFIG_MAP_NAME=my-node-config-gkt4c2m4b2
```

Execute o comando a seguir para atualizar a configuração do nó:

```
kubectl patch node ${NODE_NAME} -p "{\"spec\":{\"configSource\":{\"configMap\":
{\"name\":\"${CONFIG_MAP_NAME}\",\"namespace\":\"kube-system\",\"kubeletConfigKey\":\"kubelet\"}}}}"
```

Note: You can write a script to update the configuration of multiple nodes.

Verify the updates

Retrieve the node information and check `Node.Status.Config`:

```
kubectl get node ${NODE_NAME} -o yaml
```

For more information, see Observe that the Node begins using the new configuration.

Revert to the local default configuration

Reconfigure o nó para usar a configuração padrão.

1. Edite o nó:

   ```
   kubectl edit node ${NODE_NAME}
   ```

2. Remova o campo `Node.Spec.ConfigSource`.

3. Verifique a atualização.

   ```
   kubectl get node ${NODE_NAME} -o yaml
   ```

The `Node.Status.Config` is empty because you reset all the configuration sources to `nil` when you removed the `Node.Spec.ConfigSource` field. The local default configuration is now the `assigned`, `active`, and `lastKnownGood` configuration.

Guia de segurança

Aprenda a gerenciar a segurança e o acesso à sua plataforma.

- Certificados no IBM Cloud Private
- Integração de autenticação e conexão única
- Integração de autorização, administração e cumprimento
- Estudos de caso
- Guia de adoção do IAM
- Isolamento no IBM Cloud Private
- Configurações de cluster
- Guia de adoção do Key Management Service (KMS)
- Consultor de Vulnerabilidade
- Guia de adoção de criação de log de auditoria
- Ativando e desativando o modo FIPS

Certificados no IBM Cloud Private

Vários certificados são criados e usados em todo o IBM® Cloud Private. Informações adicionais sobre o gerenciamento de certificados criados pelo instalador podem ser encontradas nos seguintes documentos:

- Atualizando certificados
- Substituindo certificados
- Restaurando certificados

Criando um Certificado

Todos os certificados necessários pelos serviços que são executados no IBM Cloud Private são criados durante a instalação do IBM Cloud Private. Os certificados são criados e gerenciados pelo instalador do IBM Cloud Private ou pelo gerenciador de certificados do IBM Cloud Private (cert-manager).

IBM Cloud Private Certificados

Esses são os certificados que são criados automaticamente durante a instalação do IBM Cloud Private.

Criado e gerenciado pelo Instalador

Os certificados estão localizados no diretório `<install_directory>/cluster/cfc-certs/`. Todos os certificados possuem um tamanho de chave de 2.048 bits, exceto para a CA Raiz do ICP, que possui um tamanho de chave de 4.096 bits.

Certificados que podem ser atualizados após a instalação

Componente	Função	Certificado	Arquivo Chave	Pasta	Emitido / Assinado por	Duração (anos)
CA Raiz do ICP	CA Raiz do ICP	`ca.crt`	`ca.key`	root-ca/	Itself	10
etcd	CCA etcd	`ca.pem`	`ca-key.pem`	etcd/	Itself	2
etcd	autenticação de etcd client-server	`server.pem`	`server-key.pem`	etcd/	CCA etcd	10
etcd	Autenticação de cliente etcd	`client.pem`	`client-key.pem`	etcd/	CCA etcd	2
etcd	Autenticação de peer etcd	`member-<master/etcd ip>.pem`	`member-<master/etcd ip>-key.pem`	etcd/	CCA etcd	2
Proxy Frontal do Kubernetes	CA do Kubernetes Front Proxy	`front-proxy-ca.pem`	`front-proxy-ca-key.pem`	frente/	Itself	10
Proxy Frontal do Kubernetes	Servidor de Proxy Frontal do Kubernetes	`front-proxy-server.pem`	`front-proxy-server-key.pem`	frente/	CA do Kubernetes Front Proxy	2
Proxy Frontal do Kubernetes	Cliente de Proxy Frontal do Kubernetes	`front-proxy-client.pem`	`front-proxy-client-key.pem`	frente/	CA do Kubernetes Front Proxy	2
Kubernetes	TLS para o Servidor de API do Kubernetes	`server.cert`	`server.key`	kubernetes/	CA Raiz do ICP	2
Kubernetes	CLI do Kubectl	`kubecfg.crt`	`kubecfg.key`	kubernetes/	CA Raiz do ICP	2
Kubernetes	Kubelet	`kubelet-<master ip>.crt`	`kubelet-<master ip>.key`	kubernetes/	CA Raiz do ICP	2
Kubernetes	Servidor de API do Kubelet Client for Kubernetes	`kubelet-client.crt`	`kubelet-client.key`	kubernetes/	CA Raiz do ICP	2
Kubernetes	Kubernetes Proxy Client	`kube-proxy.crt`	`kube-proxy.key`	kubernetes/	CA Raiz do ICP	2
Kubernetes	Kubernetes Scheduler	`kube-scheduler.crt`	`kube-scheduler.key`	kubernetes/	CA Raiz do ICP	2
Kubernetes	Kubernetes Control Manager	`kube-controller-manager.crt`	`kube-controller-manager.key`	kubernetes/	CA Raiz do ICP	2
Helm-Tiller	Servidor Helm-Tiller	`tiller.crt`	`tiller.key`	leme /	CA Raiz do ICP	2

Componente	Função	Certificado	Arquivo Chave	Pasta	Emitido / Assinado por	Duração (anos)
Helm-Tiller	CLI do Helm	`admin.crt`	`admin.key`	leme /	CA Raiz do ICP	2
IPSec	IPSec (Strong Swan ou Libre Swan)	`ipsec-mesh.crt`	`ipsec-mesh.key`	ipsec /	CA Raiz do ICP	2

Acessando o Certificado de CA Raiz do ICP

O Certificado de CA Raiz do ICP é armazenado dentro do Segredo do Kubernetes `ibmcloud-cluster-ca-cert` no namespace `kube-public`. O certificado pode ser importado em seus armazenamentos confiáveis do cliente para acessar as APIs do IBM Cloud Private Platform.

Para recuperar e decodificar o certificado, execute o comando a seguir:

```
kubectl get secret -n kube-public ibmcloud-cluster-ca-cert -o jsonpath='{.data.ca\.crt}' | base64 --decode
```

Criado e gerenciado pelo gerenciador de certificados do IBM Cloud Private

(cert-manager) Os serviços internos a seguir usam o cert-manager para criar e gerenciar seus certificados.

- IAM
- MongoDB
- Key Management Service
- Terminal da Web
- Monitoramento
- Helm
- Servidor de Métricas
- Criação de log da auditoria
- Image Manager (Docker Registry) e ingresso de gerenciamento

Para obter mais informações sobre o cert-manager, consulte Usando o gerenciador de certificados do IBM Cloud Private.

Atualizando certificados

É possível atualizar certificados que são criados pelo instalador após a instalação do IBM Cloud Private.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

É possível atualizar certificados para os componentes a seguir:

Componente	Diretório
CA raiz	`cluster/cfc-certs/root-ca`
Etcd	`cluster/cfc-certs/etcd`
Proxy Frontal	`cluster/cfc-certs/front-proxy`
Kubernetes	`cluster/cfc-certs/kubernetes`
IPSec	`cluster/cfc-certs/ipsec`
Helm	`cluster/cfc-certs/helm`

Antes de iniciar:

Verifique se seu cluster do IBM Cloud Private está em execução.

Nota: Ao executar o comando para atualizar um certificado, os serviços de gerenciamento relacionados podem ficar indisponíveis por um curto período de tempo. O comando não tem nenhum impacto nos aplicativos em seu cluster.

Conclua as seguintes etapas para fazer backup dos certificados existentes:

1. Efetue login no nó de inicialização.
2. Mude para o diretório do cluster e crie um backup de todos os certificados executando os seguintes comandos:

```
cd <installation_directory>/cluster/
cp -r cfc-certs cfc-certs.bak
```

Atualizando certificados de autoridade de certificação raiz

1. Atualize a autoridade de certificação (CA) antiga: Exclua o certificado da autoridade de certificação antiga `ca.crt` executando o seguinte comando:

   ```
   rm -rf <cluster_dir>/cfc-certs/root-ca
   ```

2. Exclua os certificados e chaves que estão relacionados à antiga CA. Por exemplo:

   ```
   rm -rf <cluster_dir>/cfc-certs/kubernetes
   rm -rf <cluster_dir>/cfc-certs/helm
   rm -rf <cluster_dir>/cfc-certs/ipsec
   ```

3. Execute o seguinte comando para atualizar os certificados `root-ca`:

   - Para o IBM Cloud Private:

     - Para o Linux, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
       replace-certificates --tags "root-ca-certs"
       ```

     - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
       replace-certificates --tags "root-ca-certs"
       ```

     - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
       replace-certificates --tags "root-ca-certs"
       ```

   - Para o IBM Cloud Private-CE, execute o seguinte comando:

     ```
     sudo docker run --net=host -t -e LICENSE=accept \
     -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
     replace-certificates --tags "root-ca-certs"
     ```

     Nota: Ao atualizar a autoridade de certificação raiz, todos os certificados que são assinados pela autoridade de certificação raiz são atualizados automaticamente. Esses certificados incluem o Kubernetes, o IPSec e o Helm.

4. Reinicie manualmente os serviços recarregando-os. Todos os tokens padrão e serviços relacionados devem ser recarregados. Para obter informações adicionais, consulte Recarregando serviços.

5. Recrie manualmente o segredo de pull da imagem. Para obter informações adicionais, consulte Recriar segredo de pull da imagem

6. Atualize todos os certificados do cert-manager que usam a CA. Para obter informações adicionais, consulte Atualizando certificados do Cert-Manager.

7. Reinicie o Docker

   ```
   sudo systemctl restart docker
   ```

8. Execute o comando a seguir para reiniciar o Kubelet em seu nó principal.

   ```
   sudo systemctl restart kubelet
   ```

Atualizando certificados etcd

1. Exclua os certificados antigos executando o seguinte comando:

   ```
   rm -rf <cluster_dir>/cfc-certs/etcd
   ```

2. Execute o seguinte comando para substituir os certificados `etcd`:

- Para o IBM Cloud Private:
  - Para o Linux, execute o seguinte comando:

    ```
    sudo docker run --net=host -t -e LICENSE=accept \
    -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
    replace-certificates --tags "etcd-certs"
    ```

  - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

    ```
    sudo docker run --net=host -t -e LICENSE=accept \
    -v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
    replace-certificates --tags "etcd-certs"
    ```

  - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

    ```
    sudo docker run --net=host -t -e LICENSE=accept \
    -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
    replace-certificates --tags "etcd-certs"
    ```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

  ```
  sudo docker run --net=host -t -e LICENSE=accept \
  -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
  replace-certificates --tags "etcd-certs"
  ```

Atualizando certificados de proxy frontal

1. Exclua os certificados antigos executando o seguinte comando:

   ```
   rm -rf <cluster_dir>/cfc-certs/front
   ```

2. Execute o seguinte comando para substituir os certificados de `proxy frontal`:
   - Para o IBM Cloud Private:
     - Para o Linux, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
       replace-certificates --tags "front-proxy-certs"
       ```

     - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
       replace-certificates --tags "front-proxy-certs"
       ```

     - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
       replace-certificates --tags "front-proxy-certs"
       ```

   - Para o IBM Cloud Private-CE:

     ```
     sudo docker run --net=host -t -e LICENSE=accept \
     -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
     replace-certificates --tags "front-proxy-certs"
     ```

Atualizando certificados do kubernetes

1. Exclua os certificados antigos executando o seguinte comando:

   ```
   rm -rf <cluster_dir>/cfc-certs/kubernetes
   ```

2. Execute o seguinte comando para substituir os certificados do `kubernetes`:
   - Para o IBM Cloud Private:
     - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "kubernetes-certs"
```

- Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "kubernetes-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "kubernetes-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "kubernetes-certs"
```

Atualizando certificados ipsec

1. Exclua os certificados antigos executando o seguinte comando:

```
rm -rf <cluster_dir>/cfc-certs/ipsec
```

2. Execute o seguinte comando para substituir os certificados `ipsec`:

- Para o IBM Cloud Private:
  - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "ipsec-certs"
```

  - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "ipsec-certs"
```

  - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "ipsec-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "ipsec-certs"
```

Atualizando certificados do helm

1. Exclua os certificados antigos executando o seguinte comando:

```
rm -rf <cluster_dir>/cfc-certs/helm
```

2. Execute o seguinte comando para substituir os certificados do `helm`:

- Para o IBM Cloud Private:
  - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "helm-certs"
```

  - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "helm-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "helm-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "helm-certs"
```

Recarregando serviços

Recarregando o token padrão

Após a substituição da autoridade de certificação raiz, deve-se excluir o token padrão de todos os namespaces e reiniciar os serviços relacionados. É possível usar o seguinte código para excluir o token padrão de todos os namespaces e reiniciar os serviços relacionados.

```
kubectl get secret --no-headers --all-namespaces -o=custom-
columns=NAME:.metadata.name,NAMESPACE:.metadata.namespace | grep default-token | while read token;
do
    secret_name=$(echo $token | awk '{print $1}')
    secret_namespace=$(echo $token | awk '{print $2}')
    echo "------------------------------------------------------------------------"
    echo "|                       Token: ${secret_name}"
    echo "|                   Namespace: ${secret_namespace}"
    echo "------------------------------------------------------------------------"
    echo "Deleteing default token ..."
    kubectl -n ${secret_namespace} delete secret ${secret_name} &>/dev/null
    echo "Reloading services ..."
    kubectl -n ${secret_namespace} get po --field-
selector=status.phase!=Completed,status.phase!=Succeeded,status.phase!=Unknow --no-headers -
o=custom-columns=NAME:.metadata.name | while read pod; do
    secret_used=$(kubectl -n ${secret_namespace} get po ${pod} -oyaml | egrep 'secretName: default-
token|secretName: calico-node-token' &>/dev/null || echo no && echo yes)
    if [[ "$secret_used" == "yes" ]]; then
        echo "    - Restarting pod ${pod} ..."
        kubectl -n ${secret_namespace} delete po ${pod} --grace-period=0 --force &>/dev/null
    fi
    done
    echo
done
```

Recriar segredo de pull da imagem

- Para o IBM Cloud Private:
  - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
image-pull-secret
```

  - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
image-pull-secret
```

  - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
image-pull-secret
```

- Para o IBM Cloud Private-CE:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
image-pull-secret
```

Atualizando certificados do cert-manager

Após a substituição da autoridade de certificação raiz, deve-se atualizar todos os certificados que são assinados pela autoridade de certificação raiz e reiniciar os serviços que usam esses certificados. O cert-manager cria o ClusterIssuer padrão a partir da autoridade de certificação raiz, portanto, todos os certificados emitidos pelo cert-manager e assinados pelo ClusterIssuer padrão também devem ser atualizados.

O código a seguir exclui os Segredos do Kubernetes associados a cada certificado do cert-manager para atualizar o certificado e reiniciar os serviços que usam o certificado.

```
kubectl get cert --all-namespaces -o custom-columns=:spec.secretName,:metadata.namespace --no-
headers -l certmanager.k8s.io/issuer-name=icp-ca-issuer -l certmanager.k8s.io/issuer-
kind=ClusterIssuer | while read secret ; do
    name=$(echo $secret | awk '{print $1}')
    namespace=$(echo $secret | awk '{print $2}')
    kubectl delete secret $name -n $namespace
    echo "Secret $name was deleted."
done
```

Substituindo certificados

É possível substituir determinados certificados que são criados pelo instalador no ambiente IBM Cloud Private.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

É possível substituir certificados para os componentes a seguir:

Componente	Diretório
CA raiz	`cluster/cfc-certs/root-ca`

Antes de iniciar

Verifique se seu cluster do IBM Cloud Private está em execução.

Conclua as seguintes etapas para fazer backup dos certificados existentes:

1. Efetue login no nó de inicialização.
2. Mude para o diretório do cluster e crie um backup de todos os certificados executando os seguintes comandos:

```
cd <installation_directory>/cluster/
cp -r cfc-certs cfc-certs.bak
```

Substituindo certificados de autoridade de certificação raiz

1. Atualize a antiga Autoridade de Certificação (CA). Copie o certificado de autoridade de certificação preparado `ca.crt`, a chave CA `ca.key` e a chave CA de formato PKCS1 `ca.key.p1` para o diretório `<cluster_dir>/cfc-certs/root-ca/` para substituir a antiga CA. Por exemplo:

```
cp <certificate_location>/ca.crt <cluster_dir>/cfc-certs/root-ca
cp <certificate_location>/ca.key <cluster_dir>/cfc-certs/root-ca
cp <certificate_location>/ca.key.p1 <cluster_dir>/cfc-certs/root-ca
```

2. Exclua os certificados e chaves que estão relacionados à antiga CA. Por exemplo:

```
rm -rf <cluster_dir>/cfc-certs/kubernetes
rm -rf <cluster_dir>/cfc-certs/helm
rm -rf <cluster_dir>/cfc-certs/ipsec
```

3. Substitua os certificados `root-ca`.
    - Para o IBM Cloud Private:
        - Para o Linux® x86_64, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "root-ca-certs"
```

- Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "root-ca-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "root-ca-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "root-ca-certs"
```

Nota: Ao substituir a autoridade de certificação raiz, todos os certificados assinados pela autoridade de certificação raiz são atualizados automaticamente. Esses certificados incluem o Kubernetes, o IPSec e o Helm.

4. Reinicie manualmente os serviços recarregando-os. Todos os tokens padrão e serviços relacionados devem ser recarregados. Para obter informações adicionais, consulte Recarregando serviços.
5. Recrie manualmente o segredo de pull da imagem. Para obter informações adicionais, consulte Recriar segredo de pull da imagem.
6. Atualize todos os certificados do cert-manager que usam a CA. Para obter informações adicionais, consulte Atualizando certificados do cert-manager.
7. Execute o seguinte comando para reiniciar o Docker.

```
sudo systemctl restart docker
```

8. Execute o comando a seguir para reiniciar o Kubelet em seu nó principal.

```
sudo systemctl restart kubelet
```

Recarregando serviços

Recarregando o token padrão

Após a substituição da autoridade de certificação raiz, o token padrão deve ser excluído de todos os namespaces, e os serviços relacionados devem ser reiniciados. O código a seguir exclui o token padrão de todos os namespaces e reinicia os serviços relacionados facilmente.

```
kubectl get secret --no-headers --all-namespaces -o=custom-
columns=NAME:.metadata.name,NAMESPACE:.metadata.namespace | grep default-token | while read token;
do
    secret_name=$(echo $token | awk '{print $1}')
    secret_namespace=$(echo $token | awk '{print $2}')
    echo "------------------------------------------------------------------------"
    echo "|                     Token: ${secret_name}"
    echo "|                 Namespace: ${secret_namespace}"
    echo "------------------------------------------------------------------------"
    echo "Deleteing default token ..."
    kubectl -n ${secret_namespace} delete secret ${secret_name} &>/dev/null
    echo "Reloading services ..."
    kubectl -n ${secret_namespace} get po --field-
selector=status.phase!=Completed,status.phase!=Succeeded,status.phase!=Unknow --no-headers -
o=custom-columns=NAME:.metadata.name | while read pod; do
    secret_used=$(kubectl -n ${secret_namespace} get po ${pod} -oyaml | egrep 'secretName: default-
token|secretName: calico-node-token' &>/dev/null || echo no && echo yes)
    if [[ "$secret_used" == "yes" ]]; then
        echo "    - Restarting pod ${pod} ..."
        kubectl -n ${secret_namespace} delete po ${pod} --grace-period=0 --force &>/dev/null
    fi
    done
```

```
    echo
done
```

Recriar segredo de pull da imagem

- Para o IBM Cloud Private:
  - Para o Linux x86_64, execute este comando:

    ```
    sudo docker run --net=host -t -e LICENSE=accept \
    -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
    image-pull-secret
    ```

  - Para o Linux on Power (ppc64le), execute este comando:

    ```
    sudo docker run --net=host -t -e LICENSE=accept \
    -v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
    image-pull-secret
    ```

  - Para o Linux on IBM Z and LinuxONE, execute este comando:

    ```
    sudo docker run --net=host -t -e LICENSE=accept \
    -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
    image-pull-secret
    ```

- Para o IBM Cloud Private-CE:

  ```
  sudo docker run --net=host -t -e LICENSE=accept \
  -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
  image-pull-secret
  ```

Atualizando certificados do cert-manager

Após a substituição da autoridade de certificação raiz, todos os certificados assinados pela autoridade de certificação raiz devem ser atualizados e os serviços que usam esses certificados devem ser reiniciados. O cert-manager cria o ClusterIssuer padrão a partir da autoridade de certificação raiz, portanto, todos os certificados emitidos pelo cert-manager e assinados pelo ClusterIssuer padrão também devem ser atualizados.

O código a seguir exclui os Segredos do Kubernetes associados a cada certificado do cert-manager para atualizar o certificado e reiniciar os serviços que usam o certificado.

```
kubectl get cert --all-namespaces -o custom-columns=:spec.secretName,:metadata.namespace --no-
headers -l certmanager.k8s.io/issuer-name=icp-ca-issuer -l certmanager.k8s.io/issuer-
kind=ClusterIssuer | while read secret ; do
    name=$(echo $secret | awk '{print $1}')
    namespace=$(echo $secret | awk '{print $2}')
    kubectl delete secret $name -n $namespace
    echo "Secret $name was deleted."
done
```

Restaurando certificados

É possível restaurar certificados que são criados pelo instalador depois de atualizar ou substituí-los.

É possível restaurar certificados para os componentes a seguir:

Componente	Diretório
CA raiz	cluster/cfc-certs/root-ca
Etcd	cluster/cfc-certs/etcd
Proxy frontal	cluster/cfc-certs/front
Kubernetes	cluster/cfc-certs/kubernetes
IPsec	cluster/cfc-certs/ipsec
Helm	cluster/cfc-certs/helm

Antes de iniciar:

- Assegure-se de que seu cluster do IBM Cloud Private esteja em execução.

- Certifique-se de ter um backup de todos os certificados que você deseja restaurar.

Restaurando certificados de autoridade de certificação raiz

1. Copie o diretório de backup raiz cfc-certs para o diretório `cfc-certs`. Por exemplo:

   ```
   cp -rf cfc-certs.bak/root-ca cfc-certs/
   ```

2. Execute o seguinte comando para restaurar os certificados `root-ca`:

   - Para o IBM Cloud Private:

     - Para o Linux, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
       replace-certificates --tags "root-ca-certs"
       ```

     - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
       replace-certificates --tags "root-ca-certs"
       ```

     - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
       replace-certificates --tags "root-ca-certs"
       ```

   - Para o IBM Cloud Private-CE:

     ```
     sudo docker run --net=host -t -e LICENSE=accept \
     -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
     replace-certificates --tags "root-ca-certs"
     ```

     Nota: Ao restaurar a autoridade de certificação raiz, todos os certificados que são assinados pela autoridade de certificação raiz também serão atualizados automaticamente. Esses certificados incluem o Kubernetes, o IPSec e o Helm.

3. Os certificados de autoridade de certificação raiz mudaram. Deve-se recarregar todos os tokens padrão e o serviço relacionado. Para obter informações adicionais, consulte Recarregando serviços.

4. Recrie manualmente o segredo de pull da imagem. Para obter informações adicionais, consulte Recriar segredo de pull da imagem.

5. Atualize todos os certificados do cert-manager que usam a CA. Para obter informações adicionais, consulte Atualizando certificados do Cert-Manager.

6. Reiniciar Docker.

   ```
   sudo systemctl restart docker
   ```

7. Execute o comando a seguir para reiniciar o Kubelet no nó principal:

   ```
   sudo systemctl restart kubelet
   ```

Restaurando certificados etcd

1. Copie o diretório de backup etcd cfc-certs para o diretório `cfc-certs`. Por exemplo:

   ```
   cp -rf cfc-certs.bak/etcd cfc-certs/
   ```

2. Execute o seguinte comando para restaurar os certificados `etcd`:

   - Para o IBM Cloud Private:

     - Para o Linux, execute o seguinte comando:

       ```
       sudo docker run --net=host -t -e LICENSE=accept \
       -v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
       ```

```
replace-certificates --tags "etcd-certs"
```

- Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "etcd-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "etcd-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "etcd-certs"
```

Restaurando certificados de proxy frontal

1. Copie o diretório de backup do proxy frontal cfc-certs para o diretório `cfc-certs`. Por exemplo:

```
cp -rf cfc-certs.bak/front cfc-certs/
```

2. Execute o seguinte comando para restaurar os certificados de `proxy frontal`:

- Para o IBM Cloud Private:

  - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "front-proxy-certs"
```

  - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "front-proxy-certs"
```

  - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "front-proxy-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "front-proxy-certs"
```

Restaurando certificados Kubernetes

Nota: os certificados Kubernetes dependem da autoridade de certificação raiz. Se a autoridade de certificação raiz tiver mudado, não será possível restaurar os certificados do Kubernetes. Siga as etapas em Restaurando certificados de autoridade de certificação raiz para restaurar todos os certificados relacionados.

1. Copie o diretório de backup `cfc-certs` do Kubernetes no diretório `cfc-certs`. Por exemplo:

```
cp -rf cfc-certs.bak/kubernetes cfc-certs/
```

2. Execute o comando a seguir para restaurar os certificados do Kubernetes:

- Para o IBM Cloud Private:

  - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
```

```
replace-certificates --tags "kubernetes-certs"
```

- Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "kubernetes-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "kubernetes-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "kubernetes-certs"
```

Restaurando certificados IPsec

Nota: Os certificados IPsec dependem da autoridade de certificação raiz. Se a autoridade de certificação raiz tiver mudado, não será possível restaurar os certificados IPsec. Siga as etapas em Restaurando certificados de autoridade de certificação raiz para restaurar todos os certificados relacionados.

1. Copie o diretório de backup `cfc-certs` do IPsec no diretório `cfc-certs`. Por exemplo:

```
cp -rf cfc-certs.bak/ipsec cfc-certs/
```

2. Execute o seguinte comando para restaurar os certificados `ipsec`:

- Para o IBM Cloud Private:

- Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "ipsec-certs"
```

- Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "ipsec-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "ipsec-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "ipsec-certs"
```

Restaurando certificados Helm

Nota: Os certificados do Helm dependem da autoridade de certificação raiz. Se a autoridade de certificação raiz tiver mudado, não será possível restaurar os certificados do Helm. Siga as etapas em Restaurando certificados de autoridade de certificação raiz para restaurar todos os certificados relacionados.

1. Copie o diretório de backup do helm cfc-certs para o diretório `cfc-certs`. Por exemplo:

```
cp -rf cfc-certs.bak/helm cfc-certs/
```

2. Execute o seguinte comando para restaurar os certificados do `helm`:

- Para o IBM Cloud Private:

- Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
replace-certificates --tags "helm-certs"
```

- Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
replace-certificates --tags "helm-certs"
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
replace-certificates --tags "helm-certs"
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
replace-certificates --tags "helm-certs"
```

Recarregando serviços

Recarregando tokens padrão

Depois de substituir a autoridade de certificação raiz, deve-se excluir o token padrão de todos os namespaces e reiniciar os serviços relacionados. É possível usar o seguinte código para excluir o token padrão de todos os namespaces e reiniciar os serviços relacionados.

```
kubectl get secret --no-headers --all-namespaces -o=custom-
columns=NAME:.metadata.name,NAMESPACE:.metadata.namespace | grep default-token | while read token;
do
    secret_name=$(echo $token | awk '{print $1}')
    secret_namespace=$(echo $token | awk '{print $2}')
    echo "------------------------------------------------------------------------"
    echo "|                        Token: ${secret_name}"
    echo "|                   Namespace: ${secret_namespace}"
    echo "------------------------------------------------------------------------"
    echo "Deleteing default token ..."
    kubectl -n ${secret_namespace} delete secret ${secret_name} &>/dev/null
    echo "Reloading services ..."
    kubectl -n ${secret_namespace} get po --field-
selector=status.phase!=Completed,status.phase!=Succeeded,status.phase!=Unknow --no-headers -
o=custom-columns=NAME:.metadata.name | while read pod; do
    secret_used=$(kubectl -n ${secret_namespace} get po ${pod} -oyaml | egrep 'secretName: default-
token|secretName: calico-node-token' &>/dev/null || echo no && echo yes)
    if [[ "$secret_used" == "yes" ]]; then
        echo "    - Restarting pod ${pod} ..."
        kubectl -n ${secret_namespace} delete po ${pod} --grace-period=0 --force &>/dev/null
    fi
    done
    echo
done
```

Recrie segredos de pull de imagem

- Para o IBM Cloud Private:
  - Para o Linux, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee \
image-pull-secret
```

  - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee \
image-pull-secret
```

- Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee \
image-pull-secret
```

- Para o IBM Cloud Private-CE, execute o seguinte comando:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 \
image-pull-secret
```

Atualizando certificados do cert-manager

Após a substituição da autoridade de certificação raiz, deve-se atualizar todos os certificados que são assinados pela autoridade de certificação raiz e reiniciar os serviços que usam esses certificados. É possível usar o seguinte código para excluir os Segredos do Kubernetes associados a cada certificado do cert-manager para atualizar o certificado e reiniciar os serviços que usam o certificado.

```
kubectl get cert --all-namespaces -o custom-columns=:spec.secretName,:metadata.namespace --no-
headers -l certmanager.k8s.io/issuer-name=icp-ca-issuer -l certmanager.k8s.io/issuer-
kind=ClusterIssuer | while read secret ; do
    name=$(echo $secret | awk '{print $1}')
    namespace=$(echo $secret | awk '{print $2}')
    kubectl delete secret $name -n $namespace
    echo "Secret $name was deleted."
done
```

Usando o Gerenciador de certificados do IBM Cloud Private (cert-manager)

É possível usar o cert-manager do IBM Cloud Private para criar e montar um certificado para um Deployment, StatefulSet ou DaemonSet do Kubernetes. Também é possível criar e incluir um certificado em um Ingresso do Kubernetes.

Issuer, *ClusterIssuer* e *Certificate* são tipos de recursos do Kubernetes que foram introduzidos para suportar a geração e o gerenciamento de ciclo de vida de certificados. Para obter mais informações sobre cert-manager, consulte a Documentação da comunidade do cert-manager.

Consulte a lista a seguir para saber como o cert-manager do IBM Cloud Private funciona:

- O Issuer assina novos certificados e pares de chaves.
- O certificado representa um certificado X.509 e um par de chaves para o TLS ou autenticação.
- O certificado é armazenado como um Segredo do Kubernetes.
- O certificado é renovado automaticamente.

Primeiro, crie um Emissor e, em seguida, crie um certificado que será assinado pelo Emissor. O gerenciador de certificado IBM Cloud Private gera um certificado X.509 e um par de chaves e os armazena em um Segredo do Kubernetes.

Para obter mais informações sobre o Gerenciador de Certificados e outras ferramentas de configuração, consulte a documentação do produto a seguir:

- Criando seus próprios Emissores autoassinados e de CA
- Criando certificados cert-manager do IBM Cloud Private
- Visualizando recursos do cert-manager do IBM Cloud Private
- Atualizando certificados do (cert-manager) do IBM Cloud Private
- Incluindo certificados usando o Vault Issuer
- Incluindo certificados usando o Emissor Acme
- Incluindo certificados usando o algoritmo ECDSA para criptografia

Para obter informações sobre como atualizar, substituir e restaurar certificados criados e gerenciados pelo instalador, consulte Certificados no IBM Cloud Private

Integração de autenticação e conexão única

É possível integrar cargas de trabalho ao Identity and Access Management (IAM) e configurar a conexão única (SSO).

Integração é a configuração de cargas de trabalho no IAM do IBM Cloud Private. É possível configurar suas cargas de trabalho para obter os requisitos de serviço de autenticação e autorização. Como parte deste processo, as cargas de trabalho devem se tornar conhecidas pelo IAM como um serviço, registrando-se com os serviços de autenticação e autorização.

Integração de uma carga de trabalho

A primeira etapa para a integração da carga de trabalho no serviço de autenticação do IBM Cloud Private é registrar-se como um cliente do serviço de autenticação baseado no Liberty. O registro é uma etapa importante porque é por meio desse registro que o serviço de autenticação do IBM Cloud Private conhece as informações a seguir sobre a carga de trabalho:

- Um cliente autorizado do serviço de autenticação do IBM Cloud Private
- O serviço de autenticação sabe para onde redirecionar as solicitações para esse cliente após uma autenticação bem-sucedida

A maioria das cargas de trabalho de conteúdo possui uma página de login própria e deseja redirecionar para sua página de painel específica do serviço após um login bem-sucedido.

O processo de registro do cliente OpenID Connect *(OIDC)* requer acesso a um segredo que está no namespace `kube-system`. É possível visualizar o `OAUTH2_CLIENT_REGISTRATION_SECRET` a partir do segredo do Kubernetes `platform-oidc-credentials`. Os métodos a seguir estão disponíveis para obtenção do segredo do OAuth e para o registro automático:

- Método de registro de cliente automatizado 1
- Método de registro de cliente automatizado 2

Para integrar manualmente, que é obter o segredo e registrar o cliente, use o `cloudctl` ou o IAM.

Integrar usando o cloudctl

Para obter mais informações sobre os comandos IAM `cloudctl`, consulte Comandos iam da CLI do IBM Cloud Private (iam).

1. Instale o `cloudctl`. Para obter mais informações, consulte Instalando a CLI do IBM Cloud Private.

2. Construa a carga útil do registro do cliente.
   A seguir está um conteúdo de amostra a partir do arquivo `registration.json`:

```
{
"token_endpoint_auth_method":"client_secret_basic",
"client_id": "<WLP_CLIENT_ID>",
"client_secret": "<WLP_CLIENT_SECRET>",
"scope":"openid profile email",
"grant_types":[
   "authorization_code",
      "client_credentials",
      "password",
      "implicit",
      "refresh_token",
      "urn:ietf:params:oauth:grant-type:jwt-bearer"
],
 "response_types":[
   "code",
    "token",
    "id_token token"
],
 "application_type":"web",
 "subject_type":"public",
 "post_logout_redirect_uris":[
   "https://<ICP_PROXY_IP>:<PORT_WHERE_SERVICE_RUNS>"   ],
"preauthorized_scope":"openid profile email general",
"introspect_tokens":true,
"trusted_uri_prefixes":[
   "https://<ICP_ENDPOINT>:<port>", "https://<ICP_PROXY_IP>"    ],
"redirect_uris":["https://<ICP_PROXY_IP>:<PORT_WHERE_SERVICE_RUNS>/auth/liberty/callback"]
}
```

3. Crie a versão customizada de seu conteúdo do `regisration.json` editando o conteúdo de amostra.
   O `wlp_client_id` e o `wlp_client_secret` podem ser gerados pelo serviço de conteúdo. Sua saída pode ser semelhante ao código a seguir:

```
  wlp_client_id: {{ randAlphaNum 32 | b64enc | quote }}
wlp_client_secret: {{ randAlphaNum 32 | b64enc | quote }}
```

4. Registre um cliente com o serviço de autenticação do IBM Cloud Private.

```
cloudctl iam oauth-client-register -f registration.json
```

Integrar usando o IAM

1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

2. Obtenha o segredo do OAUTH:

```
OAUTH2_CLIENT_REGISTRATION_SECRET=$(kubectl -n kube-system get secret platform-oidc-credentials
-o yaml | grep OAUTH2_CLIENT_REGISTRATION_SECRET | awk '{ print $2}' | base64 --decode)
```

3. Construa a carga útil do registro do cliente.
   A seguir está um conteúdo de amostra a partir do arquivo `registration.json`:

```
{
"token_endpoint_auth_method":"client_secret_basic",
"client_id": "<WLP_CLIENT_ID>",
"client_secret": "<WLP_CLIENT_SECRET>",
"scope":"openid profile email",
"grant_types":[
   "authorization_code",
      "client_credentials",
      "password",
      "implicit",
      "refresh_token",
      "urn:ietf:params:oauth:grant-type:jwt-bearer"
],
 "response_types":[
   "code",
    "token",
    "id_token token"
],
 "application_type":"web",
 "subject_type":"public",
 "post_logout_redirect_uris":[
   "https://<ICP_PROXY_IP>:<PORT_WHERE_SERVICE_RUNS>"   ],
"preauthorized_scope":"openid profile email general",
"introspect_tokens":true,
"trusted_uri_prefixes":[
   "https://<ICP_ENDPOINT>:<port>", "https://<ICP_PROXY_IP>"    ],
"redirect_uris":["https://<ICP_PROXY_IP>:<PORT_WHERE_SERVICE_RUNS>/auth/liberty/callback"]
}
```

4. Crie a versão customizada de seu conteúdo do `regisration.json` editando o conteúdo de amostra.
   O `wlp_client_id` e o `wlp_client_secret` podem ser gerados pelo serviço de conteúdo. Sua saída pode ser semelhante ao código a seguir:

```
  wlp_client_id: {{ randAlphaNum 32 | b64enc | quote }}
wlp_client_secret: {{ randAlphaNum 32 | b64enc | quote }}
```

5. Registre um cliente com o serviço de autenticação do IBM Cloud Private. Execute o comando de API a seguir a partir de qualquer nó que tenha acesso ao nó principal:

```
curl -i -k -X POST -u oauthadmin:$OAUTH2_CLIENT_REGISTRATION_SECRET -H "Content-Type:
application/json" --data @platform-oidc-registration.json https://icp-
ip:port/idauth/oidc/endpoint/OP/registration
```

Aplicação de autenticação por cargas de trabalho

Depois de registrar o serviço de conteúdo como um cliente do serviço de autenticação do IBM Cloud Private, é possível usar os terminais de autenticação do OIDC do IBM Cloud Private para cumprir a autenticação. O IBM Cloud Private suporta dois protocolos de autenticação: OIDC e Security Assertion Markup Language (SAML). O serviço de autenticação baseado em OIDC é o serviço padrão no IBM Cloud Private. Se necessário, é possível configurar um servidor SAML para fornecer autenticação federada.

O OIDC e o SAML são usados para conexão única com o IBM Cloud Private, mas para propósitos diferentes.

O IBM Cloud Private é um provedor de identidade OIDC e fornece serviços de autenticação e autorização para a console de gerenciamento e APIs do IBM Cloud Private. Ele funciona junto com um ou mais provedores Lightweight Directory Access Protocol (LDAP) para autenticar o ID do usuário e a senha com o serviço LDAP. Ele fornece um token de acesso para solicitações subsequentes para serviços do IBM Cloud Private. O IBM Cloud Private é um provedor de identidade por meio do LDAP.

É possível configurar o IBM Cloud Private como um provedor de serviços SAML para permitir autenticação federada com um provedor de identidade SAML 2.0 externo. Ao efetuar login na console de gerenciamento, seu navegador é redirecionado para a página de login de terceiros e o OIDC emite um token de acesso para você.

Os seguintes terminais podem ser usados para cumprir a autenticação para os serviços de autenticação baseados em OIDC e em SAML:

Terminais OIDC para autenticação

O IBM Cloud Private fornece autenticação baseada em OIDC por meio do servidor WebSphere Liberty. O microsserviço de autenticação do IAM gerencia a autenticação OIDC. Esta autenticação é suportada pelo servidor OIDC baseado no Liberty para fornecer autenticação local e baseada no diretório LDAP.

As APIs OIDC padrão a seguir são suportadas pelo IBM Cloud Private:

- `https://icp-ip:port/idprovider/v1/info`
- `https://icp-ip:port/idprovider/v1/auth/identitytoken`
- `https://icp-ip:port/idprovider/v1/auth/token` (ou) `https://icp-ip:port/idauth/oidc/endpoint/OP/token`
- `https://icp-ip:port/idprovider/v1/auth/authorize`
- `https://icp-ip:port/idprovider/v1/auth/userInfo`
- `https://icp-ip:port/idprovider/v1/auth/introspect`

  Nota: A seguir está a diferença entre os dois terminais do token:

  - `https://icp-ip:port/idprovider/v1/auth/token` retorna um token criptografado e executa mais operações específicas do IBM Cloud Private antes de chamar o terminal a seguir. Em seguida, criptografa o token antes de retorná-lo.
  - `https://icp-ip:port/idauth/oidc/endpoint/OP/token` é o terminal do token OIDC do Liberty padrão.

  Os ingressos estão disponíveis para ambos os terminais. No entanto, use o prefixo.

Da mesma forma, `https://icp-ip:port/oidc/endpoint/OP/token` retorna um token criptografado e pode ser usado para todas as APIs do IBM Cloud Private. `https://icp-ip:port/idauth/oidc/endpoint/OP/token` retorna um token não criptografado e não pode ser usado para APIs do IBM Cloud Private; pode ser usado por um serviço de conteúdo para autenticação ou por um serviço que requer um provedor OIDC padrão.

Autenticação de IU - Implementação do OAUTH Dance

OAuth Dance é um processo de autenticação que identifica usuários que usam OAuth. Ele segue um processo de duas etapas:

1. Chamada para o terminal `/authorize` com `client_id`, `scope`, `redirect_uri` e `grant_type`. O terminal `/authorize` apresenta uma página de login na qual é possível inserir seu nome de usuário e senha. Em uma autenticação bem-sucedida, é retornado um código. A seguir estão os parâmetros padrão:

   - `scope`: openid+email+profile
   - `client_id`: O ID exclusivo que é usado para reconhecer um serviço com
   - `response_type`: code
   - `redirect_uri`: https://ip:port/callback. A URL para a qual o usuário é redirecionado após a autenticação bem-sucedida. Em um OAuth dance, a chamada para o terminal `/authorize` está com o valor `response_type` igual ao código.

   A seguir está um exemplo do terminal de autorização que chama a console de gerenciamento da página de login do IBM Cloud Private:

   ```
   GET https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/authorize?
   client_id=client_id&scope=openid&redirect_uri=redirect_uri&response_type=code&state=state
   ```

2. Depois que o usuário é autenticado, o servidor OIDC retorna um código. O serviço que é executado no `redirect_uri` e atende nesta URL recupera o código (https://ip:port/callback?code=xvcbdb) e chama o terminal `/token` para recuperar o

token. A seguir estão os parâmetros padrão:

- `client_id`: O ID do cliente de serviço
- `client_secret`: O segredo do cliente de serviço
- `scope`: openid
- `redirect_uri`: https://ip:port/callback
- `grant_type`: authorization_code

O terminal `/authorize` da lista de APIs do OIDC pode ser usado pelo serviço de conteúdo para seu login baseado na interface com o usuário. Esta API requer que o ID e o segredo do cliente sejam transmitidos para o serviço de autenticação do IBM Cloud Private para descobrir para onde o usuário deve ser redirecionado após uma autenticação bem-sucedida. Esses URIs são os `redirect_uris` especificados durante o processo de registro do cliente.

O `redirect_uris` é o retorno de chamada que o servidor Liberty pode chamar para concluir o OAUTH Dance. Esta URL de retorno de chamada do serviço de conteúdo deve obter o código de autorização e, em seguida, enviar a solicitação `/token` para obter os tokens de acesso válidos. Com esses tokens configurados na sessão, o serviço de conteúdo pode encaminhar a solicitação para a página desejada em seu serviço a partir da autenticação bem-sucedida.

Autenticação da CLI

O IBM Cloud Private suporta autenticação baseada na CLI. Se você estiver executando aplicativos sidecar e desejar usar os recursos de segurança do IBM Cloud Private para autenticação, é possível usar as seguintes APIs executando os comandos Curl:

Autenticar usando seu nome de usuário e senha:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
-d "grant_type=password&client_id=<ID>&client_secret=
<secret>&username=admin&password=admin&scope=openid" \
https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/identitytoken --insecure
```

Em que `<Cluster Master Host>:<Cluster Master API Port>` é definido em Terminais principais.

O seguinte é uma saída de amostra:


```
{
  "access_token":
"38400d87f39a7c328a4605265eb601bebd9426e2ef6f1b51a449da6a9cb08e03543857ac4ffbd7d2c259867c89324563c5a
89f026683aca9262858ae7ffb1e635242eabab3d579793e8f9da09070708dccf2a8d660f3be06550f02af681d2fa64562fb9
dc3df1b19839a5d3933311f89348634fa6908fa7d2d50584ffd36f9dc298a3411d3f5abad5c7f45283428ecf0de249eac553
4136c31317493f85363126bfe9a6f582403c34a3dab96e3e7bba83c263f1a4ff8d8609fca888852e097e3bc382b822576a53
e55e6753c57f79d5703cf6b6bed4b015702ce3ce1636fd834944231fa77eb90079bca398be511f22fd58792a3766a100af10
f274e6b9d75a2be2fe6ab18a3ce2ed0c8da7542e0b79f08e32a9ddced6a389572e6247230e1b62adf5fb0ee6549c06f99b85
afc7cccd7a51012dea5df40fc27a934be37e9465ddb46a4f43ec542faecb4e6dd062189392b802b8a0ad8c38a00a14f7b962
5bcdab251b87cd478c0e5d3c79f8887797da767f209f5fb2b3d44c8b051f49c2ed680a14cd15388b545ca573540184bb27be
28378dbe0ecbe2d71c0ac3d7365fce5f1948ead1576f444f70c87d7ba89352b0d2b795a11ccc5ad06441c4143a3e78f80316
c72110ba7062159f249719c664818befd6514b1526498729fe624852128495a5fa9c57ba8c9386a1040e0bb8013e93a75172
2de6e85966994cefce4c43066",
  "token_type": "Bearer",
  "expires_in": 43199,
  "scope": "openid",
  "refresh_token": "ryJlHRTJu0ZWgpDm9Ci11YenaPUk2ehZ51p1gAmL2w5VAThuff",
  "id_token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXNoIjoiNjZrYjBqMTY1NDBuN3ZhZXczem4iLCJyZWFsbU5hbWUiOi
JjdXN0b21SZWFsbSIsInVuaXF1ZVNlY3VyaXR5TmFtZSI6ImFkbWluIiwiaXNzIjoiaHR0cHM6Ly9teWNsdXN0ZXIuaWNwOjk0ND
Mvb2lkYy9lbmRwb2ludC9PUCIsImF1ZCI6IjZhNTVlMWEzZmY1Mjc5NjY2YTBiNmI4NzcxYTViMzEwIiwiZXhwIjoxNTI5MzQzOT
M4LCJpYXQiOjE1MjkzMTUxMzgsInN1YiI6ImFkbWluIiwidGVhbVJvbGVNYXBwaW5ncyI6W119.OHZTG7I5SjTk3uHIJsk7zzg5u
eQM5fEU9nC11jSvpRw-tm1T-OBqjKHPQ_g-uhmFuuym3hvQcEB-
wRQi4NMB_d580eeXHYYl_NiawunkHIl7AISQQetc7HS4U7ZXx3Mc2EmvqyVyo0zSYowGfT6D_X36O_E6Riz-
_rrGvc1nrzOdGa8IjJIi_GncSs5IFNUQxtRA9ZwdtIbQcRrSs9B3hPH8sJqUnaZnOjAkctJA8zQY0eV3IAZ4lFc01_hT5DrOdtAi
SAQBoakttxbY8iqEaNHAc07wUiN6J4rcgtJE2ZwOZth1D_39KyD5nbRbNO8HJh6hYFcBplFGwp9FDZb27A"
}
```


Esta API retorna um `access_token`, `id_token` e um `refresh_token`. O `access_token` é usado para autorização de componentes de segurança do IBM Cloud Private que incluem equipe, usuário, grupo de usuários, ID de serviço, serviço, chave de API e outros componentes. O `id_token` é usado para autorização de componentes do Kubernetes que incluem pods, daemonsets, serviços e outros componentes. O `refresh_token` pode ser usado para recuperar novos tokens, caso os tokens tenham expirado.

Recuperar um novo token

```
curl -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
-d "grant_type=refresh_token&client_id=<ID>&client_secret=
<secret>&scope=openid&refresh_token=ryJlHRTJu0ZWgpDm9Ci11YenaPUk2ehZ51p1gAmL2w5VAThuff" \
https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/token --insecure
```

O seguinte é uma saída de amostra:

```
{
  "access_token":
"77f3ea9695e50d147a3081990c331f8ce9baa0b6d02ac4e970c886eabccd7aa7e7f12e1897ceacbdf6bdaf0881ed5a725f2
14209eb20b9415c2fcf4ad1afb90412a247aeab6ab0e026e08013b8f2b773b5bdb2d8d3c1247e9e7ebeaa8c9c9c66c1e85ca
f78105e35e934a28f21619bef2ff17cebe75792da86b4a65c19973713559569e92ae6aa86ddb8ee48991c6ced9caf41ae6c3
b88f67fcaacf8c2c6af82018b5f55a4e35c1b9026438b690a606de0314bdced35eab21642b4b6c33c5241db457f2564840b9
d32c255d0bfa9e4fda176416f7481c205ee98912790a11134597ce7245264669568fd69153a8e2f240df9edb4df3b219e213
c3cfb0366713802a9a525fe85c9ec2a8c54ba61b5d845054ff23eb466c990c15dcb025ef320f36bb21ec0d0a412bcdecafba
57da6b239891e22c139a7d4057f84fd741215ed5567c3f4b824d9bbfe92d56b77fe1712d35cea60e12f5207b727e3cc658db
1b8b5002780049a5faefd8ccc2ccee9100472dfff58978ee3e7303547dc4ea03025275e58ec4e3da8e6ae91939bfb092f1ce
78fe2d91124c179f55bda4027957093090c4f47037771e9cacf227867063c909e9aee3bf87140426052821116c6484037822
a41f05a0fa565276b5ff1a8a654d3d5d119f6a665469a7591e4ec197d6a90bd586b8b95e227b9869b8654c23c10f78fc6a3f
cbbe6d543638f379736193643",
  "token_type": "Bearer",
  "expires_in": 43199,
  "scope": "openid",
  "refresh_token": "5QM3H8fmGjxhPRyYlQ77s4Z5APOHVk5276ItT8q41e2xKNMxF6"
}
```

Validar tokens

A qualquer momento, se desejar validar os tokens recebidos usando as APIs anteriores, é possível usar informações sobre o usuário ou chamada de introspecção. Veja as APIs a seguir:

Obter informações sobre o usuário

```
export ACCESS_TOKEN=<Your access token>
curl -k -X POST --header "Authorization: Bearer $ACCESS_TOKEN" \
https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/userInfo
```

O seguinte é uma saída de amostra:

```
{
  "sub":"admin",
  "iss":"https://mycluster.icp:9443/oidc/endpoint/OP"
}
```

Terminal de chamada de introspecção

```
export TOKEN=<your token>
export CLIENT_ID=<client_id>
export CLIENT_SECRET=<client_secret>
```

Obter o cabeçalho de autorização básico usando os seguintes comandos:

```
BASIC_AUTH_HEADER=`echo -n "$CLIENT_ID:$CLIENT_SECRET" | base64 -w 0`
curl -H "Authorization: Basic $BASIC_AUTH_HEADER" -d "token=$TOKEN" https://<Cluster Master Host>:
<Cluster Master API Port>/idprovider/v1/auth/introspect
```

Revogar um token

Se desejar revogar um token específico em vez de esperar sua expiração, é possível usar a seguinte API:

```
export TOKEN=<Your access token here>
export CLIENT_ID=<client_id here>
export CLIENT_SECRET=<client_secret here>
```

Obter o cabeçalho de autorização básico usando os seguintes comandos:

```
BASIC_AUTH_HEADER=`echo -n "$CLIENT_ID:$CLIENT_SECRET" | base64 -w 0`
curl -k -X POST -H "Authorization: Basic $BASIC_AUTH_HEADER" -d
"token_type_hint=access_token&token=$TOKEN" \
https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/revoke
```

O seguinte é uma saída de amostra:

```
{}
```

- Integração de política de serviço para o Helm
- API PDP bulk AuthZ

Método de registro de cliente automatizado 1

O IBM Cloud Private usa o OpenID Connect (OIDC) para permitir que cargas de trabalho integradas usem a autenticação integrada no IBM Cloud Private. É possível obter o segredo do OAuth depois de implementar um gráfico Helm.

É possível registrar-se com o OIDC e obter o segredo após a implementação do gráfico Helm. O registro durante uma implementação não é recomendado, pois requer que o instalador do gráfico ou a conta do serviço do aplicativo tenha autoridade de administrador do cluster e também acesso entre namespaces.

Como parte da implementação do gráfico do Helm, um contêiner, que inclui a lógica que é usada para registro com o OIDC, é implementado.

A seguir está um arquivo `users-config.yaml` de amostra:

```
{{/************************************************************* {COPYRIGHT-TOP} ****
* Licensed Materials - Property of IBM
*
* "Restricted Materials of IBM"
*
*  5737-H89, 5737-H64
*
* © Copyright IBM Corp. 2015, 2018  All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication, or
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
************************************************************** {COPYRIGHT-END} ****/}}
{{- $compName := "cem-users" -}}
{{- include "sch.config.init" (list . "cem.sch.chart.config.values") -}}
{{- $configMapName := include "sch.names.fullCompName" (list . $compName) -}}
kind: ConfigMap
metadata:
  name: {{ $configMapName }}
  namespace: {{ .Release.Namespace }}
  labels:
{{ include "sch.metadata.labels.standard" (list .) | indent 4 }}
    origin: helm-cem
apiVersion: v1
data:
  oidcPayload.json: |-
    {
      "token_endpoint_auth_method":"client_secret_basic",
      "client_id": "${client_id}",
      "client_secret": "${client_secret}",
      "scope":"openid profile email",
      "grant_types":[
        "authorization_code",
           "client_credentials",
           "password",
           "implicit",
           "refresh_token",
           "urn:ietf:params:oauth:grant-type:jwt-bearer"
      ],
      "response_types":[
        "code",
         "token",
         "id_token token"
      ],
      "application_type":"web",
      "subject_type":"public",
      "post_logout_redirect_uris":[
        "https://{{ .Values.global.masterIP }}:{{ .Values.global.masterPort }}/console/logout"
      ],
      "preauthorized_scope":"openid profile email general",
      "introspect_tokens":true,
```

```
      "trusted_uri_prefixes":[
        "https://{{ .Values.global.masterIP }}:{{ .Values.global.masterPort }}"
      ],
      "redirect_uris":[
         "https://{{ .Values.global.masterIP }}:{{ .Values.global.masterPort
}}/auth/liberty/callback",
         "https://{{ .Values.global.ingress.domain }}/{{ .Values.global.ingress.prefix }}",
         "https://{{ .Values.global.ingress.domain }}/{{ .Values.global.ingress.prefix
}}users/api/authprovider/v1/icp/return"
      ]
    }
  oidc_reg.sh: |-
    #!/bin/bash
    #validate no: of args
    if [[ "$1" == "?" ]]; then
      echo "USAGE: oidc_reg.sh { OIDC_CREDENTIALS }"
      echo
      echo "Run as cluster admin from your master node to register credentials for Cloud App
Management."
      echo
      exit 1
    else
      echo "Registering IBM Cloud Event Management identity ..."
    fi
    OIDC_CREDENTIALS="$1"
    OIDC_CREDENTIALS="$(base64 --decode <<< $OIDC_CREDENTIALS)"
    if [ -z "$OIDC_CREDENTIALS" ] ; then
       echo "✖ Error: OIDC_CREDENTIALS not valid!" >&2;
       echo
       exit 1
    fi
    echo
    #replace variables in OIDC_PAYLOAD and write to a temp file
    sed -e "s/\${client_id}/$AUTH_ICP_CLIENT_ID/" -e "s/\${client_secret}/$AUTH_ICP_CLIENT_SECRET/"
/etc/oidc/oidcPayload.json > /tmp/OIDC_PAYLOAD_TEMP
    #Check registration
    echo "Checking registration..."
    REG_CHECK_RESULT="$(curl -k -X GET -u oauthadmin:$OIDC_CREDENTIALS -H "Content-Type:
application/json" https://{{ .Values.global.masterIP
}}:8443/idauth/oidc/endpoint/OP/registration/$AUTH_ICP_CLIENT_ID)"
    echo
    #echo REG_CHECK_RESULT:
    #echo $REG_CHECK_RESULT
    #echo
    if [[ $REG_CHECK_RESULT = *"access_denied"* ]]; then
      echo "✖ Authentication failed, must be a admin"
      echo
    fi
    if [[ $REG_CHECK_RESULT = *"client_id_issued_at"* ]]; then
      echo "✔ Client exists..."
      #Update registration
      echo "Updating registration..."
      curl -k -X PUT -u oauthadmin:$OIDC_CREDENTIALS -H "Content-Type: application/json" --data
@/tmp/OIDC_PAYLOAD_TEMP https://{{ .Values.global.masterIP
}}:9443/oidc/endpoint/OP/registration/$AUTH_ICP_CLIENT_ID >/dev/null
      echo
    fi
    if [[ $REG_CHECK_RESULT = *"invalid_client"* ]]; then
      echo "✖ Client does not exist..."
      #Create registration
      echo "Registering client..."
      curl -k -X POST -u oauthadmin:$OIDC_CREDENTIALS -H "Content-Type: application/json" --data
@/tmp/OIDC_PAYLOAD_TEMP https://{{ .Values.global.masterIP }}:9443/oidc/endpoint/OP/registration
>/dev/null
      echo
    fi
    #Cleanup - delete the temp payload file with variables replaced
    trap "{ rm -f /tmp/OIDC_PAYLOAD_TEMP; }" EXIT
    echo
    echo Done.
```

A seguir está um arquivo `users.yaml` de amostra:

```
{{/******************************************************** {COPYRIGHT-TOP} ****
* Licensed Materials - Property of IBM
*
* "Restricted Materials of IBM"
*
*  5737-H89, 5737-H64
*
* © Copyright IBM Corp. 2015, 2018  All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication, or
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
******************************************************** {COPYRIGHT-END} ****/}}
{{- $compName := "cem-users" -}}
{{- include "sch.config.init" (list . "cem.sch.chart.config.values") -}}
{{- $deploymentName := include "sch.names.fullCompName" (list . $compName) -}}
{{- $cdbConfigTemplateName := include "sch.names.volumeClaimTemplateName" (list . "config"
$deploymentName) -}}
{{- $rootData := fromYaml (include "root.data" .) -}}
{{- $rootMetering := $rootData.metering -}}
{{- $configMapName := include "sch.names.fullCompName" (list . $compName) -}}
{{- $serviceAccountName := include "sch.names.fullCompName" (list . $compName) -}}
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: {{ $deploymentName }}
  namespace: {{ .Release.Namespace }}
  labels:
{{ include "sch.metadata.labels.standard" (list . $compName) | indent 4 }}
    origin: helm-cem
spec:
  replicas: {{ .Values.cemusers.clusterSize }}
  selector:
    matchLabels:
      release: {{ .Release.Name }}
      app: {{ include "sch.names.appName" (list .) | quote }}
      component: {{ $compName | quote }}
  template:
    metadata:
      labels:
{{ include "sch.metadata.labels.standard" (list . $compName) | indent 8 }}
        origin: helm-cem
      annotations:
        checksum/cemusers-config: {{ include (print $.Template.BasePath "/config/cem-users-
config.yaml") . | sha256sum }}
{{- include "sch.metadata.annotations.metering" (list . $rootMetering) | indent 8 }}
    spec:
{{ include "ingress-host-alias" . | indent 6 }}
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          #If you specify multiple nodeSelectorTerms associated with nodeAffinity types,
          #then the pod can be scheduled onto a node if one of the nodeSelectorTerms is satisfied.
          #
          #If you specify multiple matchExpressions associated with nodeSelectorTerms,
          #then the pod can be scheduled onto a node only if all matchExpressions can be satisfied.
          #
          #valid operators: In, NotIn, Exists, DoesNotExist, Gt, Lt
            nodeSelectorTerms:
            - matchExpressions:
              - key: beta.kubernetes.io/arch
                operator: In
                values:
                {{- if .Values.arch }}
                  - {{ .Values.arch }}
                {{- else }}
                  - {{ template "arch" . }}
                {{- end }}
      initContainers:
      - name: create-random-secrets
        image: "{{ .Values.global.image.repository }}/hdm-cem-users:{{
.Values.commonimages.cemusers.image.tag }}"
        command: ["node"]
        args: ["create-secrets.js"]
        env:
        - name: RELEASE
```

```
          value: '{{ template "releasename" . }}'
      - name: waitforcouchdb
        image: "{{ .Values.global.image.repository }}/hdm-cem-users:{{
.Values.commonimages.cemusers.image.tag }}"
        command: ["sh", "-c", "i=1;until getent hosts {{ template "releasename" . }}-couchdb.{{
.Release.Namespace }}.svc; do echo waiting for couchdb $i;i=$((i+1)); sleep 2; done;"]
      - name: waitforredis
        image: "{{ .Values.global.image.repository }}/hdm-cem-users:{{
.Values.commonimages.cemusers.image.tag }}"
        command: ["sh", "-c", "i=1;until getent hosts {{ template "releasename" . }}-redis-master-
svc.{{ .Release.Namespace }}.svc; do echo waiting for redis $i;i=$((i+1)); sleep 2; done;"]
      containers:
      - name: cem-users
        image: "{{ .Values.global.image.repository }}/hdm-cem-users:{{
.Values.commonimages.cemusers.image.tag }}"
        ports:
        - containerPort: 6002
          protocol: TCP
        livenessProbe:
          tcpSocket:
            port: 6002
          initialDelaySeconds: 120
          periodSeconds: 30
          timeoutSeconds: 20
        readinessProbe:
          tcpSocket:
            port: 6002
          initialDelaySeconds: 20
          timeoutSeconds: 20
        env:
        - name: LICENSE
          value: {{ .Values.license | default "not accepted" }}
        - name: ENV_ICP
          value: "1"
        - name: PORT
          value: "6002"
        - name: BASEURL
          value: '{{ include "cem.services.cemusers" . }}'
{{ include "cloudeventmanagement.cemusers.env" . | indent 8 }}
        - name: VCAP_APPLICATION
          value: '{}'
        - name: INGRESS_PREFIX
          value: '{{ .Values.global.ingress.prefix }}'
        - name: INGRESS_DOMAIN
          value: '{{ .Values.global.ingress.domain }}'
        resources:
{{ include "ibmcemprod.comp.size.data" (list . "cemusers" "resources") | indent 10 }}
        terminationMessagePath: "/dev/termination-log"
        imagePullPolicy: IfNotPresent
        volumeMounts:
        - name: {{ $cdbConfigTemplateName }}
          mountPath: /etc/oidc
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext:
        runAsUser: 1000
      serviceAccountName: {{ $serviceAccountName }}
      volumes:
        - name: {{ $cdbConfigTemplateName }}
          configMap:
            name: {{ $configMapName }}
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
```

Depois que o gráfico do Helm é implementado com sucesso, um administrador de cluster pode executar o comando `kubectl` a seguir para executar a lógica de registro e obter o segredo:

```
  kubectl exec -n {{ .Release.Namespace }} -t `kubectl get pods -l release={{ .Release.Name }} -n {{
.Release.Namespace }} | grep "{{ .Release.Name }}-ibm-cem-cem-users" | grep "Running" | head -n 1 |
awk '{print $1}'` bash -- "/etc/oidc/oidc_reg.sh" "`echo $(kubectl get secret platform-oidc-
credentials -o yaml -n kube-system | grep OAUTH2_CLIENT_REGISTRATION_SECRET: | awk '{print $2}')`"
```

Método de registro de cliente automatizado 2

O IBM Cloud Private usa o OpenID Connect (OIDC) para permitir que cargas de trabalho integradas usem a autenticação integrada no IBM Cloud Private. É possível obter o segredo do OAuth ao executar um script.

O administrador de cluster pode executar o script a seguir para obter o `OAUTH2_CLIENT_REGISTRATION_SECRET`:

```
#!/bin/bash

#*============================================================================
#*
#* Licensed Materials - Property of IBM
#* IBM Cost And Asset Management
#* Copyright IBM Corporation 2017. All Rights Reserved.
#*
#*============================================================================

set -o nounset
set -o pipefail
set -o errexit

# requires `kubectl`
# requires `jq` (sudo apt-get install jq -y)
# Usage
# ./oidc_regn.sh [-g GATEWAY_URL] [-n path/to/jsonfile] [-p auth/provider/host]
print_usage() {
  echo "usage: ./oidc_regn.sh [-g GATEWAY_URL] [-n path/to/jsonfile] [-p auth/provider/host]"
  echo "  where:"
  echo "        GATEWAY_URL: the API gateway full URL"
  echo "        path/to/jsonfile: Path to openid sample sso json"
  echo "        auth provider host url"
  echo ""
  echo "example: ./oidc_regn.sh -g https//myminikube.info:30091 -n openid_sso.json -p ip"
  echo ""
  exit 1
}

while getopts ':g:n:p:' flag; do
  case "${flag}" in
    g) host="${OPTARG}"
      if [ -z "${host+x}" ];
    then
    echo Enter the API gateway full URL:
    read -r host
    fi
    ;;
    n) filepath="${OPTARG}"
      if [ -z "${filepath+x}" ];
    then
    echo Enter the filepath
    read -r filepath
    fi
    ;;
    p) aph="${OPTARG}"
      if [ -z "${aph+x}" ];
    then
    echo Enter the aph
    read -r aph
    fi
    ;;
    *) print_usage
       exit 1 ;;
  esac
done

read_variable()
{
    var=$1
    value=$(python -c "import json;
with open('$filepath') as json_file:
    data = json.load(json_file)
    print (data['$var'])")
    echo $value
```

```
}

sudo apt-get install jq -y
WLP_CLIENT_ID=$(read_variable _clientID)
WLP_CLIENT_SECRET=$(read_variable _clientSecret)
kubectl get configmaps -n kube-system registration-json -o jsonpath='{.data.*}' | jq '.redirect_uris
+= ["'$host'/auth/sso/callback"]' > registration.json
jq '.client_id = "'$WLP_CLIENT_ID'"' registration.json > reg.json && mv reg.json registration.json
jq '.client_secret = "'$WLP_CLIENT_SECRET'"' registration.json > reg.json && mv reg.json
registration.json
cat registration.json

OAUTH2_CLIENT_REGISTRATION_SECRET=$(kubectl -n kube-system get secret platform-oidc-credentials -o
yaml | grep OAUTH2_CLIENT_REGISTRATION_SECRET | awk '{ print $2}' | base64 --decode)

regn_resp=$(curl -kvv -w 'RESP_CODE:%{response_code}' -S -X POST -u
oauthadmin:$OAUTH2_CLIENT_REGISTRATION_SECRET -H "Content-Type: application/json" -d
@registration.json https://$aph:8443/oidc/endpoint/OP/registration)
if [[ "$regn_resp" == *"RESP_CODE:201"* ]]; then
    echo "Successfully registered oidc client"
else
    echo "Error registering oidc client"
    echo $regn_resp
    exit 1
fi
echo Platform UI environment variables:
echo WLP_CLIENT_ID=$WLP_CLIENT_ID
echo WLP_CLIENT_SECRET=$WLP_CLIENT_SECRET
echo PLATFORM_AUTH_SERVICE_URL=https://$aph:8443/idauth
echo cfcRouterUrl=https://$aph:8443
echo PLATFORM_IDENTITY_PROVIDER_URL=https://$aph:8443/idprovider
echo OAUTH2_CLIENT_REGISTRATION_SECRET=$OAUTH2_CLIENT_REGISTRATION_SECRET
```

Integração, administração e cumprimento de autorização

A integração de autorização envolve a integração de políticas de serviço e de tipos de serviço para controle de acesso preciso.

Integração da autorização

Integração de políticas de serviço

Para usar os serviços do Identity and Access Management (IAM) para cumprir autorização, um serviço de conteúdo define suas políticas de serviço; que é um mapeamento de arquivo JSON de quais terminais podem ser acessados com qual função para o serviço. Para obter mais informações, consulte o exemplo de integração de política de serviço.

Nota: quando as políticas de serviço são inseridas no IBM Cloud Private, como com o mapeamento da função de ação e quando as APIs são acessadas por meio da porta 8443 do ingresso de gerenciamento do IBM Cloud Private, elas são restringidas automaticamente pelo Access Control Gateway do IAM. O Access Control Gateway verifica o acesso do usuário com relação à sua função para qualquer acesso a essas APIs.

Integrando tipo e instâncias de serviço

É obrigatório integrar o tipo de serviço para que o serviço de conteúdo possa fazer com que o IAM reconheça seu tipo de serviço e instâncias de serviço, a fim de que eles estejam disponíveis para cumprimento de autorização por meio da IU ou da API.

No IBM Cloud Private, os tipos de serviço e as instâncias de serviço são descobertos automaticamente para a administração do controle de ação baseado na função (RBAC) na equipe. Para que a autodescoberta ocorra, quando o gráfico da carga de trabalho é implementado e a política de serviço é definida, conforme explicado na etapa anterior, é importante que o "chartName" seja especificado. Por exemplo: "chartName": "sample-api".

Administração de autorização

O controle de acesso preciso pode ser obtido para uma carga de trabalho de conteúdo nos níveis a seguir:

- Tipo de serviço
- Instância de serviço
- Tipo de recurso

- Instância de recurso

Durante a administração da equipe, os gráficos que são identificados com o atributo `chartName` no momento da instalação são listados na página Recursos para tipos de serviço. As liberações ou as instâncias de serviço também são autodescobertas. Isso permite que os usuários escolham o tipo de gráfico/serviço e/ou a instância de liberação/serviço para aplicar controles de acesso baseados em função. O controle de acesso preciso no nível do tipo de recurso e da instância de recurso é possível, mas eles são entradas de texto livres.

Verificação de autorização

Quando o RBAC da equipe é definido no IBM Cloud Private usando o padrão anterior, a autorização pode ser verificada no nível da API. Consulte APIs do PDP AuthZ para obter exemplos de solicitações e respostas de API que mostram como a política ou a verificação de acesso pode ser feita por meio das APIs.

Configurar casos de uso não orientados pelo usuário

É possível conduzir casos de uso não orientados pelo usuário para o propósito das cargas de trabalho de conteúdo usando o conceito service-id e api-key no IBM Cloud Private. Os detalhes da administração do service-id, do api-key e das políticas associadas são discutidos mais detalhadamente na respectiva seção própria. A seguir há casos de uso de amostra que as cargas de trabalho têm e que precisam ser orientadas por uma entidade não de usuário:

- Os IDs de serviço podem ser usados para autorizar aplicativos usando cargas de trabalho de conteúdo.
- As políticas de serviço ligadas aos IDs de serviço são usadas para autorizar o acesso aos recursos.

Integração de política de serviço

A integração da política de serviço pode ser feita por meio de comandos da API.

A amostra a seguir exibe como a API de serviço e os mapeamentos de função de ação podem ser especificados no formato JSON para um serviço, neste caso, `sample-api`:

1. Crie o arquivo JSON das funções da ação da API de serviço:

   A amostra a seguir é um mapeamento de função de ação da API de serviço para os métodos GET, PUT, POST, DELETE. Observe o chartName, actions.id, e actions.roles:

   action_role_sampleapi.json:

```
{
    "chartName": "sample-api",
    "displayName":
    {
        "default": "sampleapi"
    }, "actions": [ {
        "id": "GET /sample-api/api/v1/repos",
        "displayName":
        {
            "default": "sampleapi.repos.get.allorsingular"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor", "crn:v1:icp:private:iam::::role:Viewer" ]
    },
    {
        "id": "PUT /sample-api/api/v1/repos",
        "displayName":
        {
            "default": "sampleapi.repos.put.updaterepo"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator" ]
    },
    {
        "id": "POST /sample-api/api/v1/repos",
        "displayName":
        {
            "default": "sampleapi.repos.post.addrepo"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator" ]
```

```
    },
    {
        "id": "DELETE /sample-api/api/v1/repos",
        "displayName":
        {
            "default": "sampleapi.repos.delete.removerepo"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator" ]
    }], "enabled": true, "supportedAttributes": [ {
        "chave": "cadeia"
    }], "supportedRoles": [ {
        "id": "crn:v1 :icp:private:iam ::::role :ClusterAdministrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Administrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Operator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Editor"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Viewer"
    }]
}
```

2. Crie ou atualize as funções de ação da API de serviço:

   PUT /acms/v1/services/SERVICE_NAME

```
 Entrada: Substitua o seguinte no comando curl:

 ${ACCESS_TOKEN} ---> User access token
 ${MASTER_NODE_IP} ---> Master node ipaddress or VIP ipaddress in HA
 ${SERVICE_NAME} ---> New service name
 ${API_ACTION_ROLES_JSON_FILE}  ---> API action roles json file name e.g.,
action_role_helmapi.json

 export API_ACTION_ROLES_JSON_FILE=action_role_helmapi.json
 export SERVICE_NAME=helmapi-service

 curl -k -X PUT -H 'Content-Type: application/json' -H 'Accept: application/json' -H
"Authorization: Bearer ${ACCESS_TOKEN}" -d @${API_ACTION_ROLES_JSON_FILE}
"https://${MASTER_NODE_IP}:8443/iam-pap/acms/v1/services/${SERVICE_NAME}"

 Response:
 {"name":"sampleapi-service","displayName":{"default":"sampleapi"},"actions":[{"id":"GET
/sample-api/api/v1/repos","displayName":
{"default":"sampleapi.repos.get.allorsingular"},"roles":
["crn:v1:icp:private:iam::::role:ClusterAdministrator","crn:v1:icp:private:iam::::role:Administ
rator","crn:v1:icp:private:iam::::role:Operator","crn:v1:icp:private:iam::::role:Editor","crn:v
1:icp:private:iam::::role:Viewer"]},{"id":"PUT /sample-api/api/v1/repos","displayName":
{"default":"sampleapi.repos.put.updaterepo"},"roles":
["crn:v1:icp:private:iam::::role:ClusterAdministrator"]},{"id":"POST /sample-
api/api/v1/repos","displayName":{"default":"sampleapi.repos.post.addrepo"},"roles":
["crn:v1:icp:private:iam::::role:ClusterAdministrator"]},{"id":"DELETE /sample-
api/api/v1/repos","displayName":{"default":"sampleapi.repos.delete.removerepo"},"roles":
["crn:v1:icp:private:iam::::role:ClusterAdministrator"]},"platformExtensions":
{"supportedAttributes":[{"key":"accountId"},{"key":"serviceName"}],"supportedRoles":
[]},"links":{"href":"https://9.30.255.32:8443/acms/v1/services/sampleapi-
service","link":"self"}}
```

APIs PDP AuthZ

Solicitações de API para verificar a autorização.

- Verificar Autorização
  - Verificação de autorização PDP para APIs de ação
    - Arquivos JSON de exemplo de funções de definição de serviço
    - Executando APIs PDP
  - Verificação de autorização PDP para recurso

Verificar Autorização

No IBM® Cloud Private, a verificação de autorização de ponto de decisão de política (PDP) é feita para um conjunto de ações ou de APIs de ação que são integradas ao Identity and Access Management (IAM) e aos recursos, como namespaces, que são configurados para equipes.

Verificação de autorização PDP para APIs de ação

Para APIs de ação (por exemplo, `GET /shop/dashboard`), a função de usuário é buscada a partir das equipes. A equipe pode ou não ter acesso ao namespace. A função de usuário é validada com funções de definição de serviço integradas para validar a autorização para uma API de ação específica.

A seguir estão exemplos dos formatos de API de ação:

- `GET /shop/dashboard`
- `POST /shop/edit`
- `feature.cluster.manage`
- `feature.topic.write`
- `action.view`
- `feature.update`

Essas funções de definição de serviço devem ser integradas ao IAM para que elas possam validar qualquer API para autorização.

As funções de definição de serviço podem ser integradas usando o gráfico `security-onboarding` durante a instalação do IBM Cloud Private ou usando as APIs de integração de serviço após a instalação do IBM Cloud Private. Para obter mais informações sobre as APIs de integração de serviço, consulte APIs de integração de serviço e RBAC.

Arquivos JSON de exemplo de funções de definição de serviço

A seguir estão exemplos de funções de definição de serviço:

Exemplo 1: action_roles_shop.json

```
{
    "name": "shop",
    "displayName":
    {
        "default": "shop"
    }, "actions": [ {
        "displayName":
        {
            "default": "shop-view"
        },
        "id": "GET /shop/dashboard",
        "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
            "crn:v1:icp:private:iam::::role:Administrator",
            "crn:v1:icp:private:iam::::role:Editor",
            "crn:v1:icp:private:iam::::role:Viewer"
        ]
    },
    {
        "displayName":
        {
            "default": "shop-edit"
        },
        "id": "POST /shop/edit",
        "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
            "crn:v1:icp:private:iam::::role:Administrator"
        ]
    }], "supportedRoles": [ {
        "id": "crn:v1 :icp:private:iam ::::role :ClusterAdministrator"
    },
    {
```

```
        "id": "crn:v1 :icp:private:iam ::::role :Administrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Editor"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Viewer"
    }],
    "enabled": true
}
```

Exemplo 2: action_roles_feature.json

```
{
    "name": "feature",
    "displayName":
    {
        "default": "Actions for feature"
    },
    "chartName": "ibm-feature-dev",
    "actions": [
    {
        "id": "feature.cluster.read",
        "displayName":
        {
            "default": "Cluster read"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor", "crn:v1:icp:private:iam::::role:Viewer" ]
    },
    {
        "id": "feature.cluster.operate",
        "displayName":
        {
            "default": "Cluster operate"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator" ]
    },
    {
        "id": "feature.topic.read",
        "displayName":
        {
            "default": "Topic read"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor", "crn:v1:icp:private:iam::::role:Viewer" ]
    },
    {
        "id": "feature.topic.write",
        "displayName":
        {
            "default": "Topic write"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor" ]
    },
    {
        "id": "feature.topic.manage",
        "displayName":
        {
            "default": "Topic manage"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator" ]
    }], "supportedRoles": [ {
        "id": "crn:v1 :icp:private:iam ::::role :ClusterAdministrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Administrator"
    },
```

```
    {
        "id": "crn:v1 :icp:private:iam ::::função :Operator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Editor"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Viewer"
    }],
    "enabled": true
}
```

Para `Verificação de autorização PDP para APIs de ação`, a carga útil de entrada PDP deve conter valor para apenas o atributo "action".

Carga útil de entrada PDP de amostra 1

```
{
    "action": "GET /shop/dashboard",
    "subject":
    {
        "id": "", "type": ""
    }, "resource": {
        "crn": "",
        "attributes":
        {
            "serviceName": "", "accountId": "" }
    }
}
```

Carga útil de entrada PDP de amostra 2

```
{
    "action": "feature.topic.write",
    "subject":
    {
        "id": "", "type": ""
    }, "resource": {
        "crn": "",
        "attributes":
        {
            "serviceName": "", "accountId": "" }
    }
}
```

Executando APIs PDP

Para executar essas APIs, deve-se incluir um cabeçalho de autorização em sua solicitação. É necessário um token de acesso para incluir no cabeçalho de autorização. Para obter o token de acesso, consulte Preparando para executar os comandos da API do componente ou de gerenciamento.

`<Cluster Master Host>:<Cluster Master API Port>` são usados para acessar as APIs. Os parâmetros são definidos nos Terminais mestres.

Comando curl de amostra que usa a carga útil de entrada PDP de amostra 1

A seguir está um comando curl de amostra que usa a carga útil de entrada de PDP de amostra 1:

```
curl -k -X POST --header "Content-Type: application/json" --header "Accept: application/json" --
header "Authorization: bearer ${ACCESS_TOKEN}" -d '{
  "action": "GET /shop/dashboard",
  "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "",
    "attributes": {
      "serviceName": "", "accountId": "" }
  }
}' "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pdp/v1/authz"
```

A seguir estão as respostas de amostra:

- Se o usuário não tiver uma função em nenhuma equipe, será exibida uma resposta `Deny`.

```
{"decision":"Deny","obligations":[{"actions":["GET /shop/dashboard"],"crns":
[""],"decision":"Deny","max-age":86400,"obligationId":"bf69c79fbe636dc8"}]}
```

- Se o usuário tiver uma função que seja necessária para uma ação em qualquer equipe, será exibida uma resposta `Permit`.

```
{"decision":"Permit","obligations":[{"actions":["GET /shop/dashboard"],"crns":
[""],"decision":"Permit","max-age":86400,"obligationId":"bf69c79fbe636dc8"}]}
```

Comando curl de amostra que usa a carga útil de entrada PDP de amostra 2

A seguir está um comando curl de amostra que usa a carga útil de entrada PDP de amostra 2:

```
curl -k -X POST --header "Content-Type: application/json" --header "Accept: application/json" --
header "Authorization: bearer ${ACCESS_TOKEN}" -d '{
  "action": "feature.topic.write",
  "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "",
    "attributes": {
      "serviceName": "", "accountId": "" }
  }
}' "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pdp/v1/authz"
```

A seguir estão as respostas de amostra:

- Se o usuário não tiver uma função em nenhuma equipe, será exibida uma resposta `Deny`.

```
{"decision":"Deny","obligations":[{"actions":["feature.topic.write"],"crns":
[""],"decision":"Deny","max-age":86400,"obligationId":"bf69c79fbe636dc8"}]}
```

- Se o usuário tiver uma função que seja necessária para uma ação em qualquer equipe, será exibida uma resposta `Permit`.

```
{"decision":"Permit","obligations":[{"actions":["feature.topic.write"],"crns":
[""],"decision":"Permit","max-age":86400,"obligationId":"bf69c79fbe636dc8"}]}
```

Verificação de autorização PDP para recurso

Para recursos, como um namespace, a verificação de autorização deve enviar o CRN do recurso para a API de verificação de autorização.

O usuário deve ter uma função e acesso aos recursos nas equipes. Somente então, a API retorna uma resposta `Permit`. Caso contrário, a API retorna uma resposta `Deny`.

A carga útil de entrada PDP deve conter o valor para os atributos `action` e `resource.crn`.

Se a ação não for integrada e se o valor do atributo `action` contiver palavras, como `create`, `read`, `update` ou `delete`, as funções padrão serão consideradas. A tabela a seguir tem as funções padrão para uma ação:

Tabela 1. Ação e funções padrão

Ação	Funções suportadas
criar	ClusterAdministrator, Administrator, Operator
leitura	ClusterAdministrator, Administrator, Operator, Editor, Viewer
atualizar	ClusterAdministrator, Administrator, Operator, Editor
excluir	ClusterAdministrator, Administrator

Carga útil de entrada PDP de amostra

```
{
  "action": "action.read",
  "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/default:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
}
```

Comando curl de amostra

```
curl -k -X POST --header "Content-Type: application/json" --header "Accept: application/json" --
header "Authorization: bearer ${ACCESS_TOKEN}" -d '{
  "action": "action.read",
  "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/default:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
}' "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pdp/v1/authz"
```

Respostas de amostra

- Se o usuário não tiver uma função e acesso aos recursos nas equipes, o comando retornará uma resposta `Deny`.

  ```
  {"decision":"Deny","obligations":[{"actions":["action.read"],"crns":
  ["crn:v1:icp:private:k8:mycluster:n/default:::"],"decision":"Deny","max-
  age":86400,"obligationId":"bf69c79fbe636dc8"}]}
  ```

- Se o usuário tiver uma função e acesso aos recursos em qualquer equipe, o comando retornará uma resposta `Permit`.

  ```
  {"decision":"Permit","obligations":[{"actions":["action.read"],"crns":
  ["crn:v1:icp:private:k8:mycluster:n/default:::"],"decision":"Permit","max-
  age":86400,"obligationId":"bf69c79fbe636dc8"}]}
  ```

Verificação de autorização PDP usando a API authz_bulk

Esta solicitação de API pode ser usada para verificar a autorização para uma ou mais entradas PDP usando uma única API.

```
Input:
${MASTER_NODE_IP} ---> Master node ipaddress
${ACCESS_TOKEN} ---> User access token
payload:
{"inputArray":[{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/kube-public:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
},{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/default:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
},{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/kube-system:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
},{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/services:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
}]}


Command:
curl -k -X POST --header "Content-Type: application/json" --header "Accept: application/json" --
header "Authorization: bearer $ACCESS_TOKEN" -d '{"inputArray":[{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
```

```
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/kube-public:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
},{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/default:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
},{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/kube-system:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
},{
  "action": "audit.view", "subject": {
    "id": "", "type": ""
  }, "resource": {
    "crn": "crn:v1:icp:private:k8:mycluster:n/services:::", "attributes": {
      "serviceName": "", "accountId": "" }
  }
}]}' "https://$MASTER_NODE_IP:8443/iam-pdp/v1/authz_bulk"

Response:
{
    "responses": [
        {
            "input": {
                "action": "audit.view", "resource": {
                    "attributes": {
                        "serviceName": "", "accountId": ""
                    },
                    "crn": "crn:v1:icp:private:k8:mycluster:n/kube-public:::"
                }, "subject": {
                    "type": "user",
                    "id": "abc",
                    "groupIds": [
                        "cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com" ]
                }
            }, "output": {
                "obligations": [ {
                        "decision": "Permit",
                        "obligationId": "4a42af5b19174ee9",
                        "crns": [
                            "crn:v1:icp:private:k8:mycluster:n/kube-public:::"
                        ], "actions": [
                            "audit.view"
                        ], "max-age": 86400 }
                ], "decision": "Permit" }
        },
        {
            "input": {
                "action": "audit.view", "resource": {
                    "attributes": {
                        "serviceName": "", "accountId": ""
                    },
                    "crn": "crn:v1:icp:private:k8:mycluster:n/default:::"
                }, "subject": {
                    "type": "user",
                    "id": "abc",
                    "groupIds": [
                        "cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com" ]
                }
            }, "output": {
                "obligations": [ {
                        "decision": "Deny",
                        "obligationId": "29059b42665fa0e9",
                        "crns": [
                            "crn:v1:icp:private:k8:mycluster:n/default:::"
                        ], "actions": [
                            "audit.view"
```

```
                    ], "max-age": 86400 }
                ], "decision": "Deny" }
        },
        {
            "input": {
                "action": "audit.view", "resource": {
                    "attributes": {
                        "serviceName": "", "accountId": ""
                    },
                    "crn": "crn:v1:icp:private:k8:mycluster:n/kube-system:::"
                }, "subject": {
                    "type": "user",
                    "id": "abc",
                    "groupIds": [
                        "cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com" ]
                }
            }, "output": {
                "obligations": [ {
                        "decision": "Deny",
                        "obligationId": "41e1470939e395a1",
                        "crns": [
                            "crn:v1:icp:private:k8:mycluster:n/kube-system:::"
                        ], "actions": [
                            "audit.view"
                        ], "max-age": 86400 }
                ], "decision": "Deny" }
        },
        {
            "input": {
                "action": "audit.view", "resource": {
                    "attributes": {
                        "serviceName": "", "accountId": ""
                    },
                    "crn": "crn:v1:icp:private:k8:mycluster:n/services:::"
                }, "subject": {
                    "type": "user",
                    "id": "abc",
                    "groupIds": [
                        "cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com" ]
                }
            }, "output": {
                "obligations": [ {
                        "decision": "Permit",
                        "obligationId": "5d7994e5023edeaf",
                        "crns": [
                            "crn:v1:icp:private:k8:mycluster:n/services:::"
                        ], "actions": [
                            "audit.view"
                        ], "max-age": 86400 }
                ], "decision": "Permit" }
        }
    ]
}
```

Verificar autorização para o ID de serviço

A verificação de autorização para o ID de serviço funciona da mesma maneira que a verificação de autorização para APIs de ação ou recurso. Deve-se fornecer o token de chave de API do ID de serviço em vez do token de acesso do usuário. Para obter mais informações sobre como gerar o token de chave de API do ID de serviço, consulte as APIs a seguir:

1. Criar um ID de serviço
2. Criar uma API chave
3. Gerar um token OpenID Connect (OIDC)

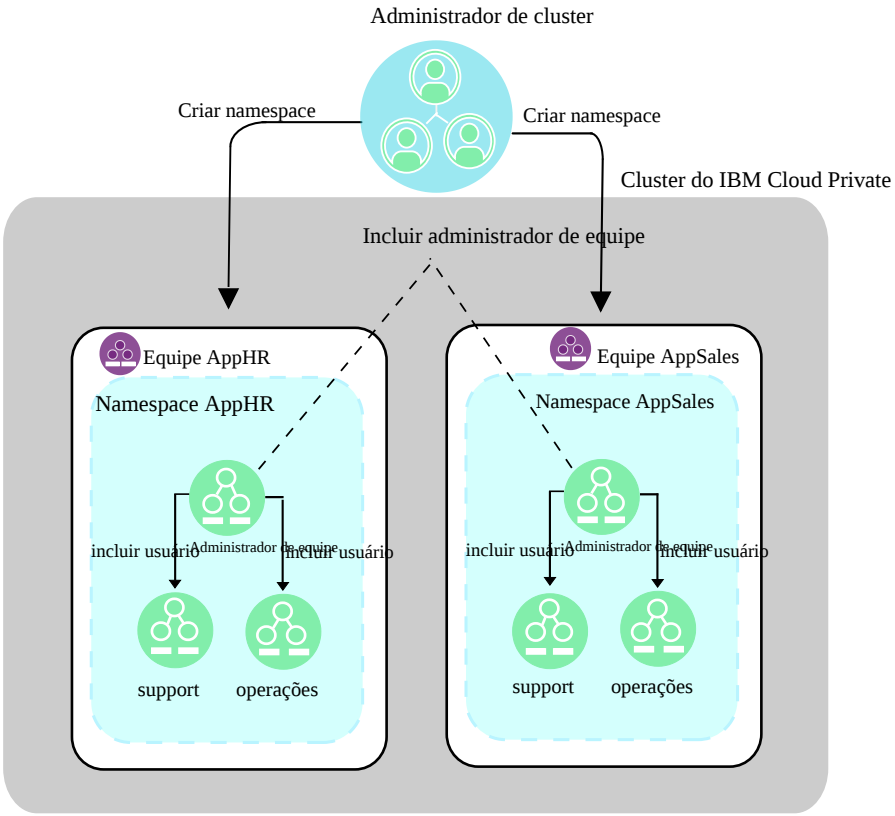
A função de ID de serviço é buscada a partir de equipes e de políticas de ID de serviço para uma verificação de autorização de API de ação e recurso. O ID do serviço deve ter acesso ao recurso nas equipes ou para as políticas de ID de serviço para verificação de autorização de recurso. Para obter mais informações sobre como criar políticas de ID de serviço, consulte APIs de gerenciamento de política de serviço.

Estudos de caso

A plataforma IBM Cloud Private permite usar recursos de cluster entre aplicativos em modelos compartilhados e dedicados. Há vários tipos de requisitos de configuração de diferentes usuários do IBM Cloud Private que requerem isolamento em vários níveis. Esta seção destaca cenários que são suportados no IBM Cloud Private que ajudam a atingir o isolamento em vários níveis da pilha do cluster; como aplicativo, gerenciamento de usuário, infraestrutura e rede.

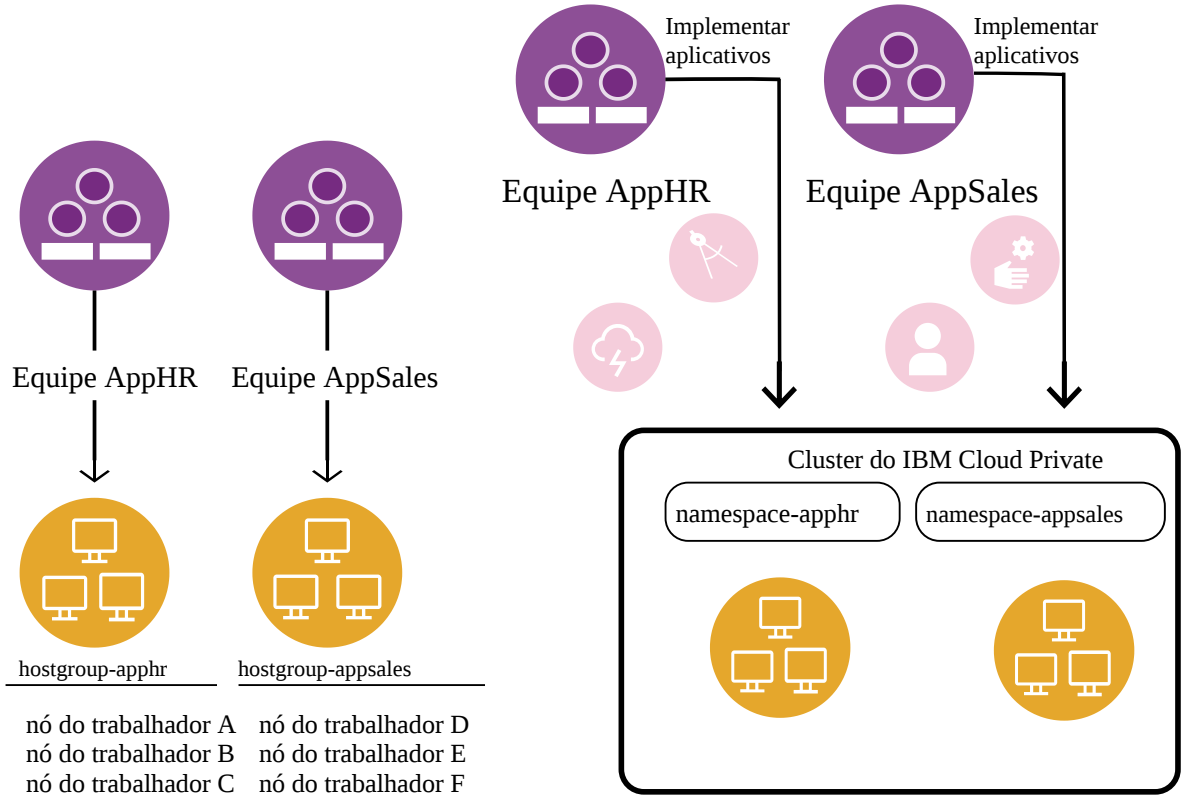
Funções de usuário e isolamento de namespace

O usuário deseja usar um único cluster do IBM Cloud Private para múltiplas equipes do projeto do aplicativo com isolamento completo. O usuário possui uma equipe de Operações que gerencia um único cluster com a alta disponibilidade configurada. Os membros da equipe de Operações definem namespaces individuais para as equipes de projeto separadas. Eles designam a função de administrador do cluster para uma ou mais pessoas por equipe do projeto, que, em seguida, definem os membros e as funções da equipe para esse namespace. Os membros da equipe de Operações que possuem a função do administrador do cluster são responsáveis por configurar o armazenamento compartilhado e monitorar todos os namespaces.



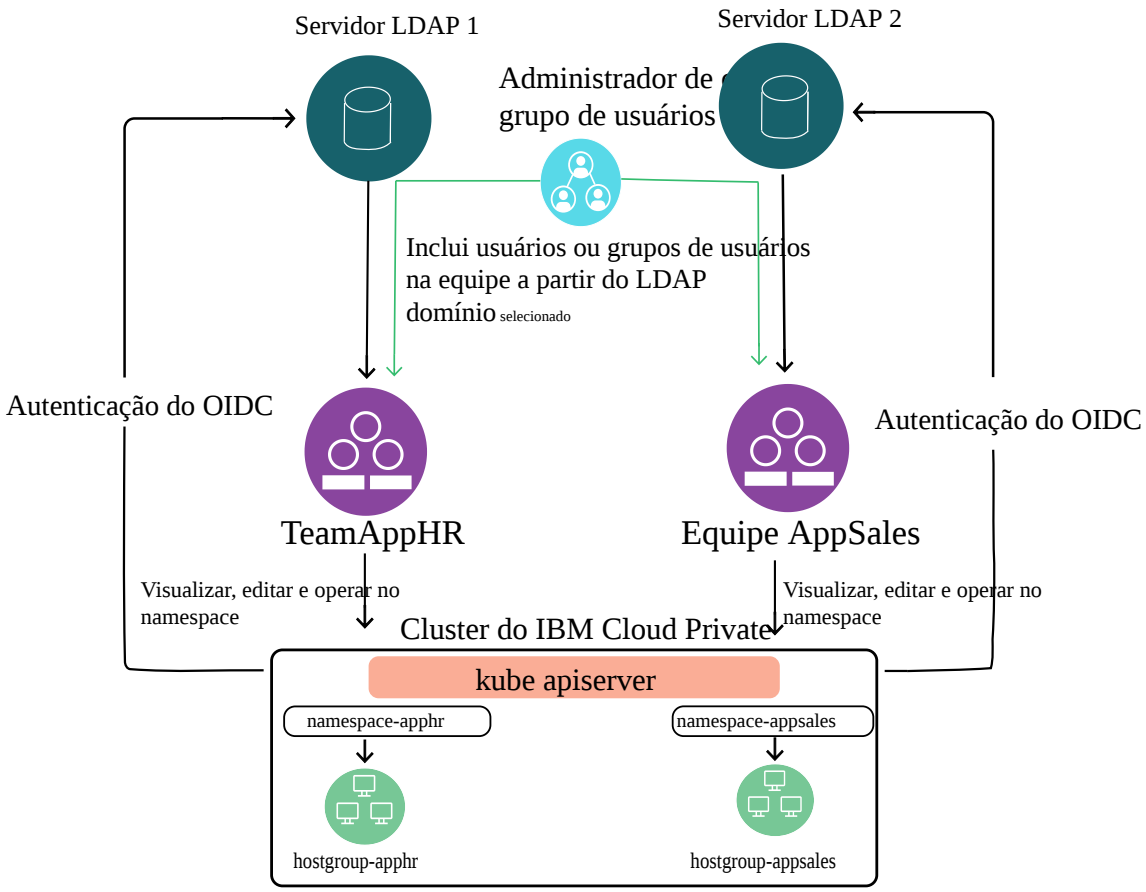
Cargas de trabalho isoladas para nós

O usuário tem múltiplas equipes em sua organização e tem namespaces dedicados para cada equipe. As implementações de aplicativos e cargas de trabalho de qualquer equipe devem acontecer somente dentro do namespace designado à equipe. Cada equipe possui um grupo de nós planejado, ou seja, servidores físicos ou máquinas virtuais, que são incluídos no cluster, em que pode haver muitos outros nós a partir de várias equipes configurados e gerenciados como parte do mesmo cluster do IBM Cloud Private. Todas as implementações feitas por uma equipe específica devem ser hospedadas somente no grupo de nós ao qual a equipe foi designada.



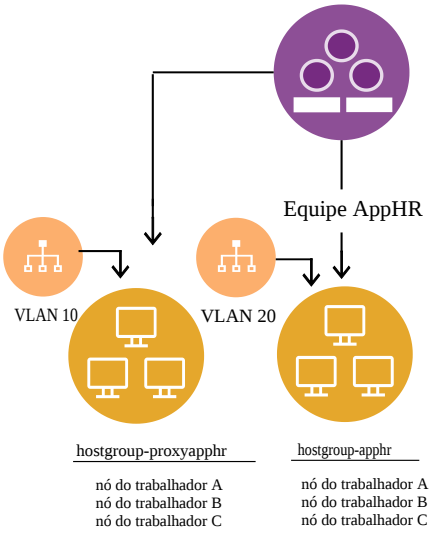
Múltiplos clusters suportados pelo LDAP

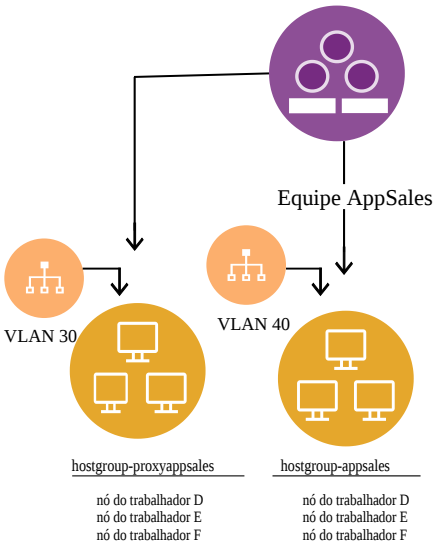
Há múltiplos servidores LDAP, como o OpenLDAP e o Active Directory, que são configurados para vários departamentos em uma organização. Cada departamento é isolado para seu próprio grupo de usuários como equipes. Cada equipe tem namespaces dedicados, alcançando ambientes de namespace isolados para os vários departamentos que acessam o mesmo cluster do IBM Cloud Private.

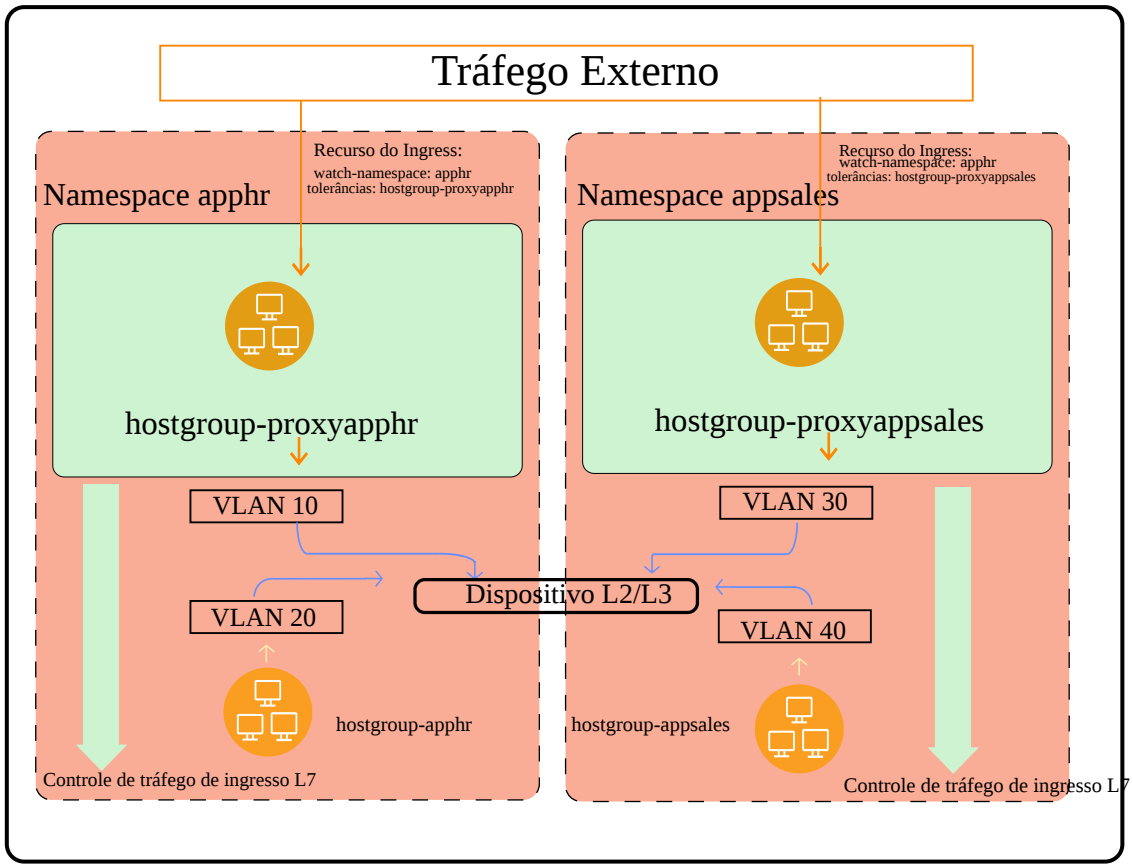


Cargas de trabalho com isolamento de rede controlado

As cargas de trabalho no site do usuário são executadas em uma infraestrutura de rede compartilhada. O usuário é executado em um cluster no qual múltiplas equipes em suas organizações hospedam cargas de trabalho em seus namespaces isolados dedicados. O tráfego de rede para os aplicativos no namespace de um cluster não deve ter nenhuma interferência no tráfego de rede em outros namespaces e todo o tráfego deve ser confinado no namespace da implementação. Além disso, os nós são agrupados e designados com intervalos de sub-rede de rede local virtual (VLAN) dedicada.


Equipe AppHR
VLAN 10
VLAN 20
hostgroup-proxyapphr
hostgroup-apphr
nó do trabalhador A
nó do trabalhador B
nó do trabalhador C
nó do trabalhador A
nó do trabalhador B
nó do trabalhador C


Equipe AppSales
VLAN 30
VLAN 40
hostgroup-proxyappsales
hostgroup-appsales
nó do trabalhador D
nó do trabalhador E
nó do trabalhador F
nó do trabalhador D
nó do trabalhador E
nó do trabalhador F



Isolamento de pod

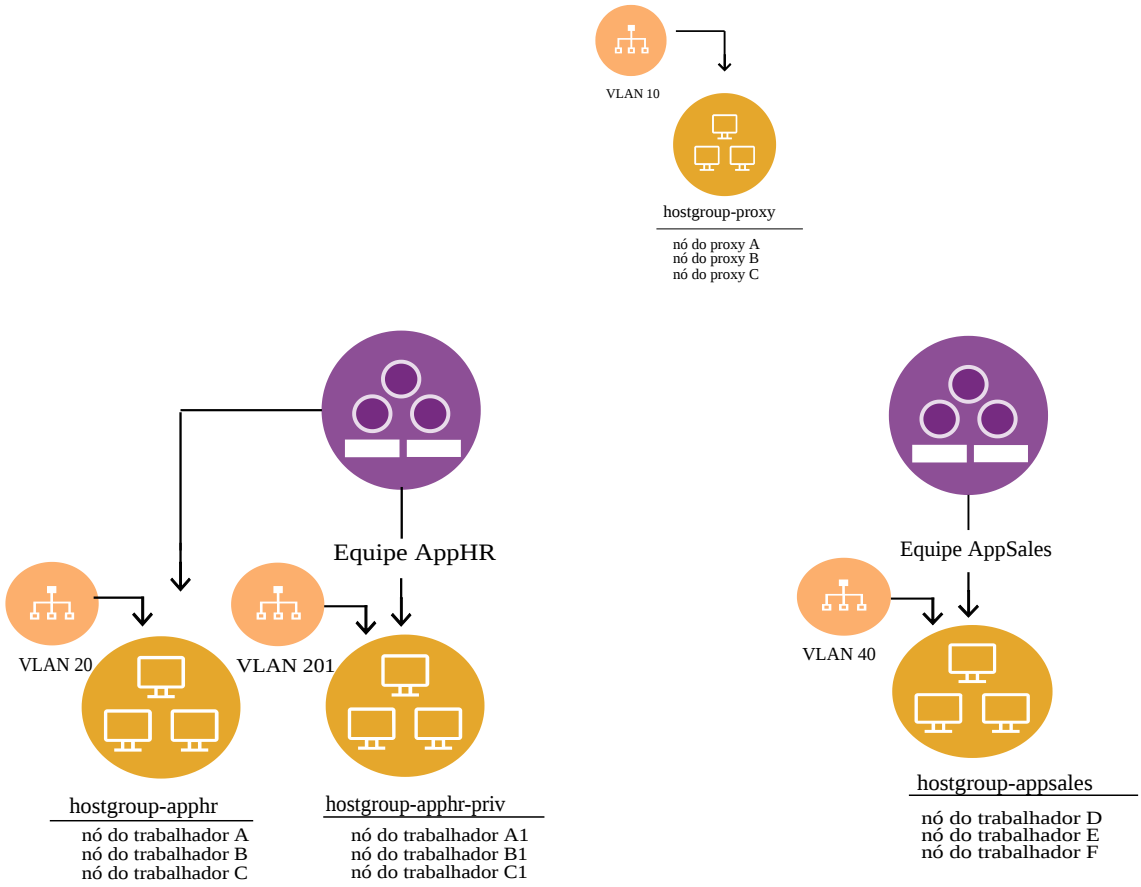
Os contêineres de imagem são executados em pods em um conjunto compartilhado de nós do trabalhador. Esses pods podem ser autocontidos, que não acessam nenhum recurso do nó do host, ou os pods podem solicitar acesso privilegiado aos recursos do host. Ao solicitar acesso privilegiado para o host, o pod pode fazer com que outros pods no mesmo host falhem.

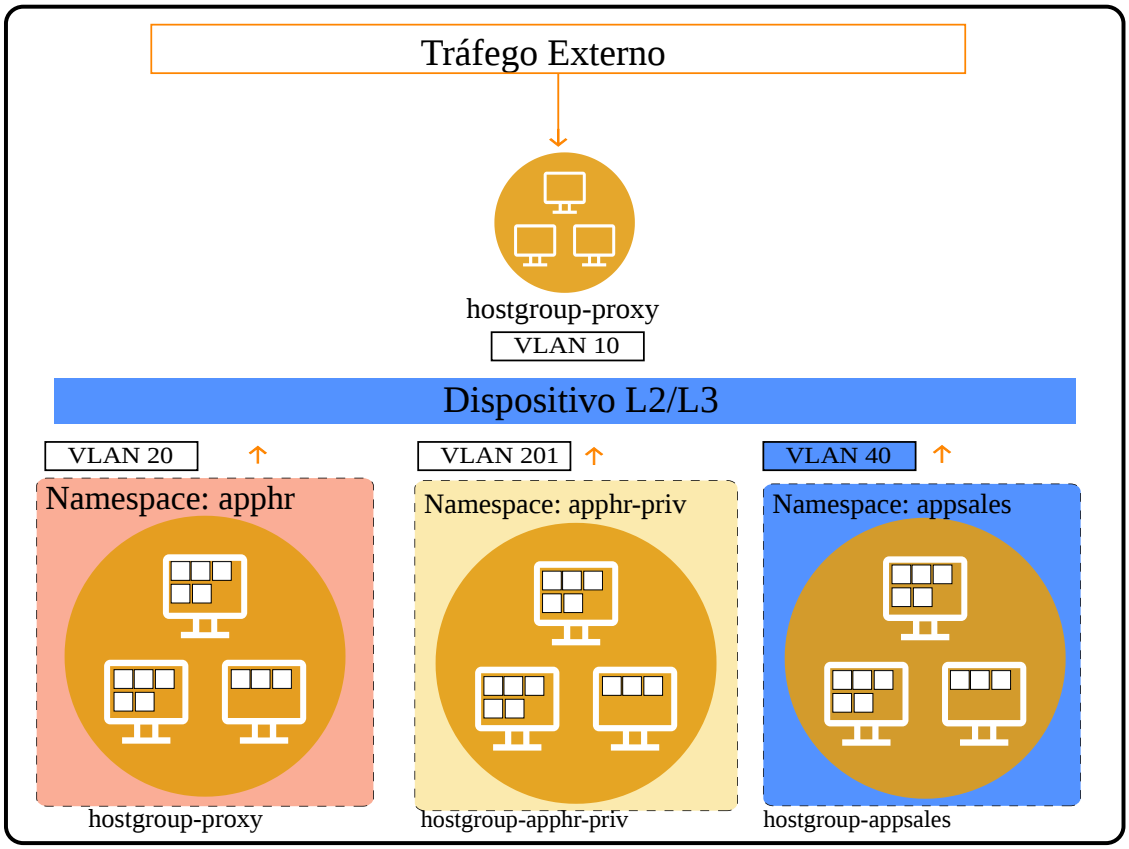
Isole pods com diferentes requisitos de contexto de segurança para grupos de hosts específicos e redes para aumentar a estabilidade, o desempenho e a segurança.

O isolamento de ingresso do proxy não pode ser usado para segregação de rede ao usar o isolamento de pod, já que múltiplos namespaces são usados. Use VLANs ou políticas de rede para segregar as redes do nó do trabalhador umas das outras.

Isolamento de pod e de rede com VLANs

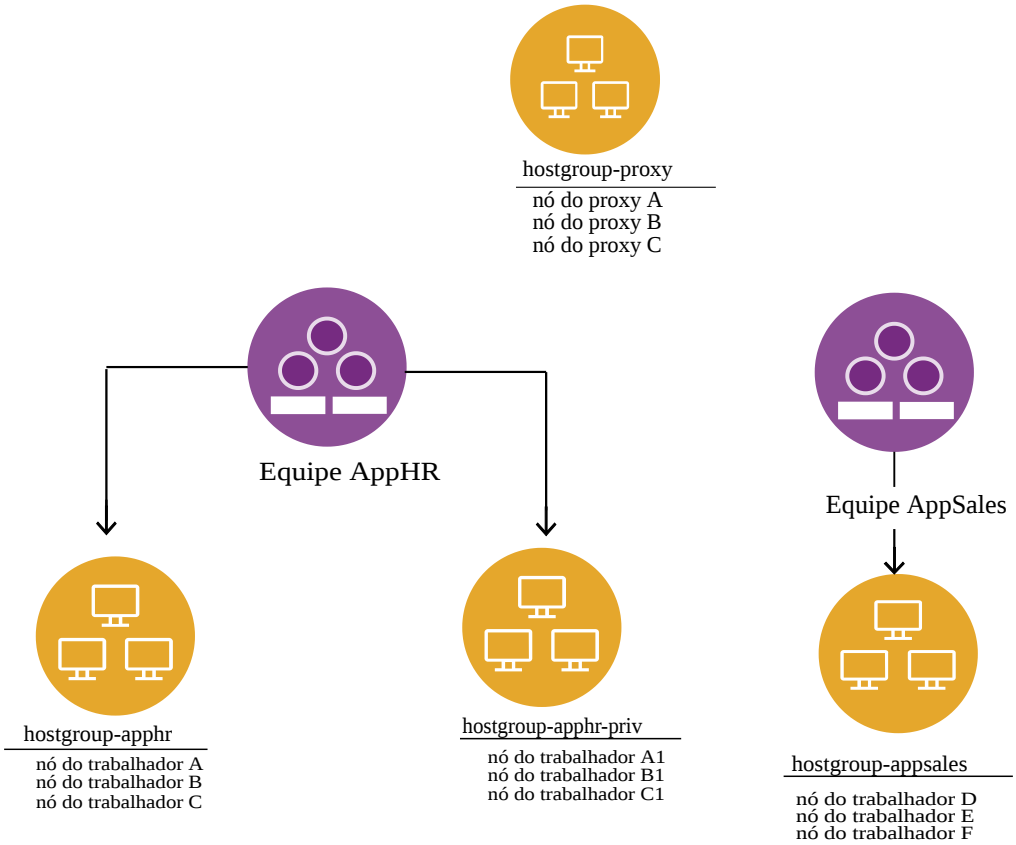
As VLANs de rede no nível de infraestrutura podem ser usadas para isolar o tráfego em um aplicativo de outro e em grupos de hosts. Para implementar o isolamento de pod e de rede usando VLANs, separe namespaces em duas ou mais categorias de privilégio de pod, que variam da menos privilegiada para a mais privilegiada.

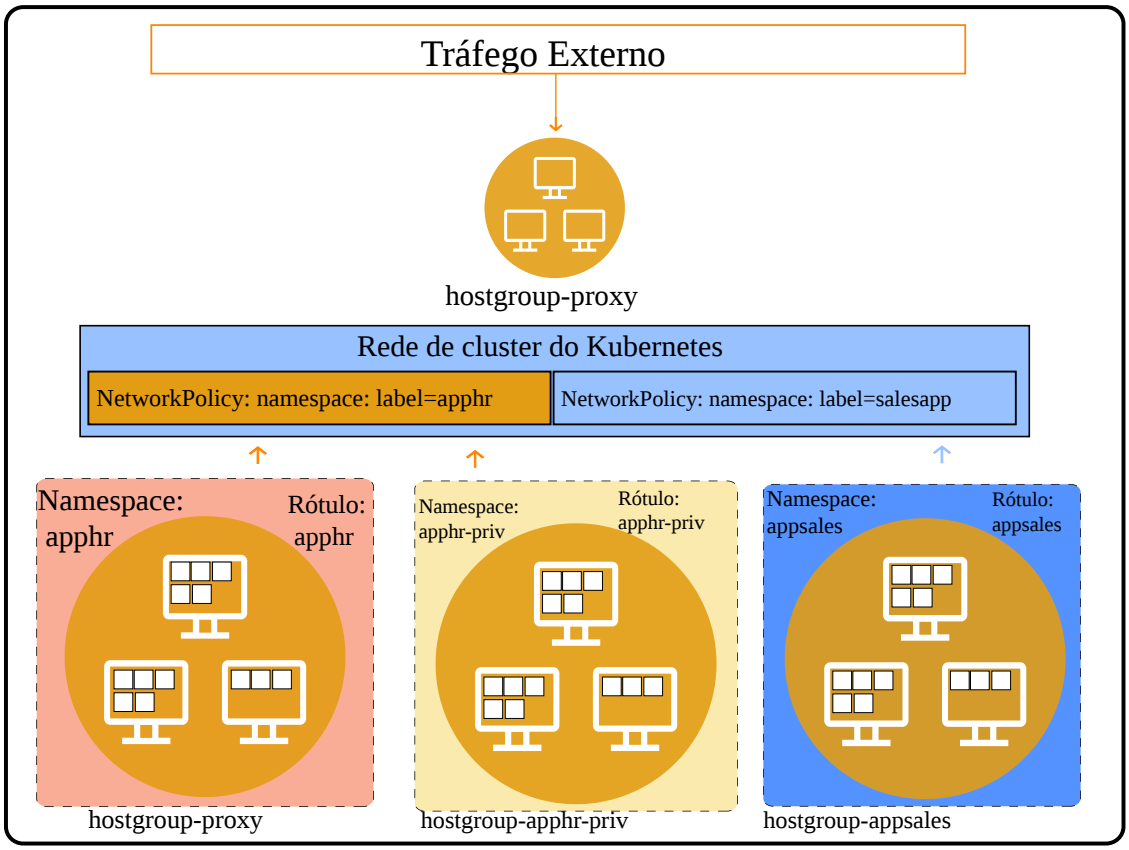




Isolamento de pod e de rede usando políticas de rede

As políticas de rede do Kubernetes podem ser usadas para isolar o tráfego entre namespaces ou pods usando rótulos. Para implementar o isolamento de pod e de rede usando políticas de rede, separe namespaces em duas ou mais categorias de privilégio de pod que variam da menos privilegiada para a mais privilegiada e use um rótulo de namespace comum para permitir o tráfego de ingresso e de egresso entre namespaces no mesmo aplicativo.





Guia de adoção do IAM

Este tópico cobre os conceitos do Identity and Access Management (IAM) no IBM Cloud Private e discute como os usuários e as cargas de trabalho de conteúdo podem alavancar os serviços IAM do IBM Cloud Private para autenticação e autorização. Ele também cobre os recursos do IAM e a especificação de API relacionada.

Em um alto nível, o IBM Cloud Private fornece suporte de autenticação por meio da especificação do OpenID Connect (OIDC). O IBM Cloud Private também suporta autenticação federada baseada em Security Assertion Markup Language (SAML). A estrutura de autorização se alinha com o Cloud IAM com alguma customização específica do IBM Cloud Private, como agrupamento de entidades do usuário e do recurso sob uma construção de equipe.

Os detalhes de autenticação e autorização são explorados nos tópicos a seguir:

- Uma visualização do usuário para usar os serviços de autenticação e autorização do IBM Cloud Private
- Uma visualização de carga de trabalho de conteúdo para usar os serviços de autenticação e autorização IBM Cloud Private
- IAM para serviço para comunicação de serviço
- IAM para IBM Cloud Private with OpenShift

Para a adoção do IAM, detalhes de configuração podem ser necessários, como o endereço IP e o número da porta de seu cluster. Para obter informações sobre o cluster, consulte ConfigMap de configuração de cluster.

Para resolução de problemas, consulte Resolução de problemas do IAM.

Para APIs, consulte APIs do IAM.

IAM para usuários da plataforma do IBM Cloud Private

O Identity and Access Management (IAM) para usuários da plataforma inclui a autenticação que inclui o OIDC e o SAML e a autorização que inclui controle de acesso baseado na função, gerenciamento de usuários e Naming Resource Naming (CRN).

Autenticação

O IBM® Cloud Private usa o WebSphere Liberty OpenID Connect (OIDC) 1.0 para autenticação. Ele chama os terminais OIDC padrão `/authorize` e `/token` para iniciar uma movimentação OAuth. O OpenID no Liberty pode ser configurado com o Lightweight Directory Access Protocol (LDAP), após o qual um usuário LDAP pode autenticar-se no IBM Cloud Private usando os mesmos terminais do OpenID. Para autenticação baseada em conexão única (SSO), o OIDC é configurado com o Security Assertion Markup Language (SAML) para interagir com sua origem da identidade corporativa.

Protocolos de autenticação suportados

O IBM Cloud Private suporta os dois protocolos de autenticação a seguir:

1. Autenticação baseada em OIDC
2. Autenticação federada baseada em SAML

O OIDC e o SAML são usados para SSO com o IBM Cloud Private, mas para propósitos diferentes.

O IBM Cloud Private é um provedor de identidade OIDC que fornece serviços de autenticação e autorização para a console de gerenciamento e APIs do IBM Cloud Private. Ele funciona junto com um ou mais provedores LDAP para autenticar o ID do usuário e a senha com o serviço LDAP e para fornecer um token de acesso para solicitações subsequentes para serviços do IBM Cloud Private. O IBM Cloud Private é um provedor de identidade por meio do LDAP.

O IBM Cloud Private pode ser configurado como um provedor de serviços SAML, que permite autenticação federada com um provedor de identidade SAML 2.0 externo. Ao configurar a SSO, o IBM Cloud Private redireciona o navegador da console de gerenciamento para a página de login de terceiros, e o OIDC emite um token de acesso.

O serviço de autenticação baseado em OIDC é o serviço de autenticação padrão no IBM Cloud Private. Se necessário, é possível configurar um servidor SAML para fornecer autenticação federada.

Autenticação baseada em OIDC

Deve-se configurar e conectar um diretório LDAP ao cluster do IBM Cloud Private e fornecer o nível de acesso de administrador de cluster ou de administrador. Para obter mais informações, consulte Configurando a conexão LDAP. Deve-se configurar a conexão LDAP antes de criar uma equipe e incluir usuários na equipe. Somente usuários LDAP que são designados a uma equipe podem efetuar login na console de gerenciamento.

Para obter mais informações sobre como criar uma equipe, consulte Criar equipes. Para obter mais informações sobre como incluir usuários em uma equipe, consulte Incluir usuários em uma equipe.

Terminal de autorização

Se você deseja que seu aplicativo chame a página de login da console de gerenciamento do IBM Cloud Private, deve-se usar o seguinte terminal:

```
https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/authorize?
client_id=client_id&scope=openid&redirect_uri=redirect_uri&response_type=code&state=state
```

APIs do terminal Token

Há dois terminais do token:

1. `https://<Cluster Master Host>:<Cluster Master API Port>/idprovider/v1/auth/token`: este retorna um token criptografado e executa mais operações específicas do IBM Cloud Private antes de fazer uma chamada para o segundo terminal. Após a chamada, ele criptografa o token antes de retorná-lo. Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.
2. `https://<Cluster Master Host>:<Cluster Master API Port>/idauth/oidc/endpoint/OP/token`: este é o terminal do token OIDC do OOTB Liberty.

Ingressos estão disponíveis para ambos os terminais, no entanto, o uso do prefixo é recomendado. O uso do prefixo também é recomendado para os terminais a seguir: `https://<Cluster Master Host>:<Cluster Master API Port>/oidc/endpoint/OP/token` e `https://<Cluster Master Host>:<Cluster Master API Port>/idauth/oidc/endpoint/OP/token`.

Nota: o terminal número 1 é recomendado, porque ele retorna o token criptografado e funciona em todas as APIs do IBM Cloud Private. O token decriptografado que é retornado por `/oidc/endpoint/OP/token` não funcionará com APIs do IBM Cloud Private. Ele pode ser usado por um serviço de conteúdo para autenticação ou por um que requer um provedor OIDC OOTB.

Autenticação baseada em SAML

O IBM Cloud Private pode ser configurado para usar a autenticação baseada em SAML a partir de um servidor SAML corporativo. Para obter mais informações, consulte Configurando a conexão única.

Nota:

- O IBM Cloud Private não suporta login de SSO usando a CLI.
- Depois de configurar o SAML, é possível efetuar login como o usuário administrador padrão ou como um usuário LDAP que é incluído em uma equipe.

Suporte do Lightweight Directory Access Protocol (LDAP)

O IBM Cloud Private pode ser configurado com um único ou com múltiplos servidores LDAP para autenticação e autorização. O IBM Cloud Private suporta os tipos de LDAP a seguir:

- IBM Tivoli Directory Server
- IBM Lotus Domino
- IBM SecureWay Directory Server
- Novell eDirectory
- Sun Java™ System Directory Server
- Netscape Directory Server
- Microsoft Active Directory
- Customizado

Configurando a Conexão Única

Para obter mais informações, consulte Configurando a conexão única.

Configurando LDAPs

Com o IBM Cloud Private, você é capaz de autenticar-se em múltiplos LDAPs. É possível incluir múltiplas entradas de diretório na configuração do LDAP no arquivo `server.xml`. O Liberty resolve automaticamente o nome do domínio do login e é autenticado com relação ao diretório LDAP de destino. Os usuários e grupos de usuários do IBM Cloud Private são associados a um diretório corporativo durante o momento da integração de usuário e de grupo de usuários por meio da importação. Quando a nova entrada de diretório LDAP é criada, o nome de domínio também é incluído como uma nova entrada. No momento do login, é possível especificar o domínio com relação ao qual sua autenticação deve ser validada.

É possível ter uma combinação de tipos de diretório, como AD, Tivoli e OpenLDAP. O controle de acesso baseado na função (RBAC) é cumprido no domínio LDAP. Os administradores de cluster têm acesso a todos os domínios do LDAP, enquanto que os administradores da equipe estão restritos apenas aos domínios aos quais eles estão autorizados.

Para obter mais informações, consulte Configurando a conexão LDAP.

Mudando configurações de cache LDAP

Para obter mais informações, consulte Mudando configurações de cache LDAP.

Mudando configurações de procura LDAP

Para obter mais informações, consulte Mudando configurações de procura LDAP.

Mudando a propriedade de configuração do Logjam

Para obter mais informações, consulte Mudando a propriedade de configuração de Logjam.

Resolução de Problemas do LDAP

Para obter informações adicionais, consulte Resolução de problemas de LDAP.

LDAP sobre SSL

Para obter mais informações, consulte Configurando a conexão LDAP.

Autorização

Verifique os tópicos a seguir para obter mais informações sobre autorização para usuários da plataforma:

IBM Cloud Private e controle de acesso baseado na função (RBAC) do Kubernetes

O RBAC é cumprido no IBM Cloud Private por meio de equipes. Uma equipe é uma entidade que agrupa usuários e recursos. Os recursos podem ser recursos do tipo Kubernetes (como namespace, pod e broker) ou um tipo não Kubernetes, como gráfico Helm, instância do BD e conexão em nuvem. A designação de recursos para a equipe acontece por meio de CRNs de recurso. Os serviços responsáveis devem expor os CRNs de recursos por meio de uma API, para que eles se tornem disponíveis no diálogo `Incluir recurso` da equipe. Os recursos do Kubernetes, como os namespaces, são expostos por meio da API `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/teams/resources?resourceType=namespace`. É possível buscar os recursos que estão conectados a um usuário específico por meio de suas equipes usando a API `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/teams/resources`.

Para obter mais informações, consulte controle de acesso baseado na função e equipes.

Especificação de CRN

O IBM Cloud Private segue a convenção de CRN: `crn:version:cname:ctype:service-name:region:scope:service-instance:resource-type:resource-instance`.

As definições de atributo CRN a seguir são específicas para os serviços da plataforma IBM Cloud Private:

- **cname**: `icp`
- **ctype**: `private`
- **service-name**: um serviço registrado do IAM, por exemplo: `k8`, `security`, `iam`.
- **region**: o cluster-id do conjunto de clusters durante o momento da instalação. Isso é configurado como o parâmetro `cluster_name` do arquivo de configuração `config.yaml`. O padrão, quando não especificado, é `mycluster`. Esse valor não muda durante a vida do cluster.

Outros atributos são específicos do serviço.

O IBM Cloud Private define as construções a seguir para aplicativos com base em:

1. Plataforma do Kubernetes
2. Cargas de trabalho
3. Plataforma IBM Cloud Private Cloud Foundry

CRNs para recursos do Kubernetes

A `region` / `cluster-id` é configurada para o parâmetro `cluster_name` do arquivo de configuração `config.yaml`. O padrão, quando não especificado, é `mycluster'.

```
crn:v1:icp:private:k8:mycluster::::
crn:v1:icp:private:k8:mycluster:n/namespace-id:::
crn:v1:icp:private:k8:cluster-id:n/namespace-id::deployment:deployment-name
crn:v1:icp:private:k8:cluster-id:n/namespace-id::pod:pod-name
crn:v1:icp:private:k8:cluster-id:n/namespace-id::service:service-name
crn:v1:icp:private:k8:cluster-id:::persistent-volume:volume-name
```

CRNs para recursos do catálogo de serviços

```
crn:v1:icp:private:k8:mycluster:::clusterservicebroker:csb123
crn:v1:icp:private:k8:mycluster:::clusterserviceclass:csc123 (to be changed to:
crn:v1:icp:private:k8:mycluster:sc/csc123::::)
crn:v1:icp:private:k8:mycluster:sc/csc123::clusterserviceplan:csp123
crn:v1:icp:private:k8:cluster-id:n/namespace-id::serviceinstance:instanceid
crn:v1:icp:private:k8:cluster-id:n/namespace-id::servicebinding:bindingid
```

CRNs para recursos relacionados à segurança

```
crn:v1:icp:private:security:mycluster::User:userId
crn:v1:icp:private:security:mycluster::Directory:directoryId
```

```
crn:v1:icp:private:security:mycluster::UserGroup:groupId
crn:v1:icp:private:security:mycluster::Team:teamId
```

CRNs para funções do IAM do IBM Cloud Private

```
crn:v1:icp:private:iam::::role:ClusterAdministrator
crn:v1:icp:private:iam::::role:Administrator
crn:v1:icp:private:iam::::role:Operator
crn:v1:icp:private:iam::::role:Editor
crn:v1:icp:private:iam::::role:Viewer
```

CRNs para recursos de criação de log e de monitoramento

```
crn:v1:icp:private:logging:mycluster::xxxx:yyyy
crn:v1:icp:private:monitoring:mycluster::xxxx:yyyy
```

CRNs para recursos do Helm

```
crn:v1:icp:private:helm-catalog:cluster-id:c/chart-name:::
crn:v1:icp:private:helm-catalog:cluster-id:c/chart-name::DBaaS
crn:v1:icp:private:helm-catalog:cluster-id:r/repo-name::repoid
crn:v1:icp:private:helm-catalog:cluster-id:r/repo-name::repo-name
crn:v1:icp:private:k8:cluster-id:n/namespace-id::release:release-name
```

Fluxos de eventos

```
crn:v1:icp:private:eventstreams:cluster-id:n/namespace-id:::
crn:v1:icp:private:eventstreams:cluster-id:n/namespace-id:r/eventstream-instanceid::
crn:v1:icp:private:eventstreams:cluster-id:n/namespace-id:r/eventstream-instanceid:resourceType:
crn:v1:icp:private:eventstreams:cluster-id:n/namespace-id:r/eventstream-
instanceid:resourceType:resourceId
```

CRNs para cargas de trabalho definidas pelo usuário

```
crn:v1:icp:private:user-defined:cluster-id:u/mongoaaS-id:MongoDB:Mongodb-id crn:v1:icp:private:user-
defined:cluster-id:u/mySQLaaS-id:MySQLDB:Mysqldb-id
```

CRNs para o Key Management Service

```
crn:v1:icp:private:kms:cluster-id:n/kube-system:::
crn:v1:icp:private:kms:cluster-id:n/kube-system::key:key-id
crn:v1:icp:private:kms:cluster-id:n/kube-system:instance-id::
crn:v1:icp:private:kms:cluster-id:n/kube-system:instance-id:key:
crn:v1:icp:private:kms:cluster-id:n/kube-system:instance-id:key:key-id
```

Exemplos de CRN para cargas de trabalho hospedadas na plataforma IBM Cloud Private

```
- crn:version:cname:ctype:service-name:region:scope:service-instance:resource-type:resource-instance
- scope - repo, pod, user-defined
```

- Um Operador configura um repositório atrás de um monocular e descobre gráficos Helm, alguns dos quais são brokers, como DBaaS: `crn:v1:icp:private:helm-catalog:cluster-id:r/repo-name:helm-charts:DBaaS`
- Há dois cenários com o Helm:
  1. Implementação de um gráfico de middleware típico (não X-aaS modelo), que, em seguida, cria uma liberação. Uma liberação é um artefato em execução na plataforma: `crn:v1:icp:private:k8:cluster-id:n/namespace-id::release:release-name`
  2. A implementação de um gráfico pode criar serviços. Um operador implementa um gráfico DBaaS, que, em seguida, cria o MongoaaS e o MySQLaaS no catálogo de serviços:

     ```
     crn:v1:icp:private:user-defined:cluster-id::u/mongo-aas::
     crn:v1:icp:private:user-defined:cluster-id::u/mysql-aas::
     ```

- Quando um Desenvolvedor efetua login, ele vê esses serviços:

  ```
  rn:v1:icp:private:helm-catalog::r/repo-name:catalog-name:mongoaaS:mongoaaS-id
  crn:v1:icp:private:helm-catalog::r/repo-name:catalog-name:mySQLaaS:mySQLaaS-id
  ```

- Quando um Desenvolvedor pode implementar o MongoDB a partir do serviço e usá-lo:

```
crn:v1:icp:private:user-defined:cluster-id:u/mongoaaS-id:MongoDB:Mongodb-id
crn:v1:icp:private:user-defined:cluster-id:u/mySQLaaS-id:MySQLDB:Mysqldb-id
```

CRN para aplicativo que está hospedado na plataforma IBM Cloud Private Cloud Foundry

```
crn:version:cname:ctype:service-name:region:scope:service-instance:resource-type:resource-instance
scope - account, org, space
```


Configurando a Conexão Única

Configure a conexão única (SSO) entre o IBM® Cloud Private e sua origem da identidade corporativa.

A SAML (Security Assertion Markup Language), uma linguagem de marcações baseada em XML, é um padrão aberto para a troca de informações de identidade, autenticação e autorização entre um provedor de identidade (o servidor SAML corporativo) e um provedor de serviços (o cluster IBM Cloud Private).

O provedor de identidade emite asserções de autenticação juntamente com um perfil SSO de SAML. O provedor de serviços recebe essas asserções e o perfil.

O fluxo de SSO pode ser resumido conforme a seguir:

1. Um usuário tenta acessar um serviço no IBM Cloud Private por meio de um navegador da web.
2. O IBM Cloud Private verifica se há um token de autenticação presente.
3. Se não houver um token de autenticação presente, o IBM Cloud Private redireciona a solicitação de autenticação para o servidor SAML corporativo do usuário.
4. O servidor SAML corporativo apresenta uma página de login para o usuário.
5. Se o usuário efetuar login com êxito, o servidor SAML o redirecionará, juntamente com a resposta SAML, para o IBM Cloud Private.
6. O IBM Cloud Private gera um token de autenticação e concede acesso ao serviço solicitado pelo usuário.

Configurando a SSO no IBM Cloud Private

Arquivos de metadados são utilizados para a comunicação entre o cluster IBM Cloud Private e o servidor SAML corporativo.

Pré-requisitos

- Deve-se configurar um nome completo do domínio (FQDN) para acessar o cluster. É possível configurar esse FQDN no arquivo `config.yaml` durante a instalação do IBM Cloud Private. Ou, pode-se seguir as instruções em Customizando a URL de acesso do cluster para incluir uma URL customizada após a instalação do IBM Cloud Private.
- O diretório LDAP (Lightweight Directory Access Protocol) conectado com o cluster IBM Cloud Private deve incluir usuários que podem usar a solicitação SSO. Ou, é possível conectar o IBM Cloud Private com o mesmo servidor LDAP que o servidor SAML corporativo utiliza para autenticação.

Configurando a SSO

Para configurar a SSO, conclua a seguinte sequência de etapas:

1. Ative o SAML.
2. Exporte os metadados do IBM Cloud Private para o servidor SAML corporativo. Após a conclusão dessa tarefa, um arquivo de metadados do IBM Cloud Private é transferido por download.
3. Importe os metadados enviados pelo servidor SAML corporativo.
4. Verifique se o SAML foi configurado com êxito.

É possível usar a interface de programação de aplicativos (API) ou a interface da linha de comandos (CLI) para configurar a SSO no IBM Cloud Private.

Configurando a SSO usando as APIs

Para configurar a SSO usando as APIs, consulte APIs de conexão única.

Configurando a SSO usando a CLI

Pré-requisitos

Instale a CLI do IBM Cloud Private. Para obter mais informações, consulte Instalando a CLI do IBM Cloud Private.

Os comandos a seguir estão disponíveis para configurar e gerenciar a SSO no cluster do IBM Cloud Private.

- Ativar a SAML (Security Assertion Markup Language)
- Exportar arquivo de metadados
- Import metadata file
- Verificar status de configuração de SSO
- Desativar SAML

Ativar SAML

Ative a SSO.

```
cloudctl iam saml-enable
```

Exportar arquivo de metadados

Ao executar o comando, um arquivo de metadados é transferido por download do IBM Cloud Private e salvo com o nomes de arquivo especificado. Faça upload desse arquivo para o servidor SAML corporativo.

```
cloudctl iam saml-export-metadata -- file < file_name> .xml
```

Um arquivo de metadados de amostra assemelha-se ao código a seguir:

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" \
entityID="https://travistest.rtp.raleigh.ibm.com:8443/ibm/saml20/defaultSP"><md:SPSSODescriptor
AuthnRequestsSigned="true" \
WantAssertionsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"> \
<md:KeyDescriptor use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data> \
<ds:X509Certificate>MIID9zCCAd8CCQDIJbZgmPut9DANBgkqhkiG9w0BAQsFADBjMQswCQYDVQQGEwJVUzERMA8GA1UE
.
.
btEmEMpzbGQy8Lb190tLeLZNW2zrBWbRmxzShn9ekS58aEbeD6PBTzWsKXsgYhZWWXw=</ds:X509Certificate> \
</ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:KeyDescriptor use="encryption"> \
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:X509Data> \
<ds:X509Certificate>MIID9zCCAd8CCQDIJbZgmPut9DANBgkqhkiG9w0BAQsFADBjMQswCQYDVQQGEwJVUzERMA8GA1UE
.
.
btEmEMpzbGQy8Lb190tLeLZNW2zrBWbRmxzShn9ekS58aEbeD6PBTzWsKXsgYhZWWXw=</ds:X509Certificate>
</ds:X509Data> \
</ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" \
Location="https://travistest.rtp.raleigh.ibm.com:8443/ibm/saml20/defaultSP/slo"/>
<md:AssertionConsumerService \
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" \
Location="https://travistest.rtp.raleigh.ibm.com:8443/ibm/saml20/defaultSP/acs" index="0"
isDefault="true"/>\
</md:SPSSODescriptor></md:EntityDescriptor>
```

Importar arquivo de metadados

Ao executar o comando, você faz upload do arquivo de metadados recebido do servidor SAML corporativo para o IBM Cloud Private.

```
cloudctl iam saml-upload-metadata -- file < file_name> .xml
```

Um arquivo de metadados de amostra assemelha-se ao código a seguir:

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20">
```

```
<md:IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>MIIDhTCCAm2gAwIBAgIEOxmOOjANBgkqhkiG9w0BAQsFADBzMQswCQYDVQQGEwJVUz\
.
.
3YZ25IwGyzN5KK7XR1avMCk9GG0BbpjpqU29Wx3tWpqsh+Kl016Kc=</X509Certificate>
</X509Data>
</KeyInfo>
</md:KeyDescriptor>
<md:KeyDescriptor use="encryption">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>MIIDhTCCAm2gAwIBAgIEOxmOOjANBgkqhkiG9w0BAQsFADBzMQswCQYDVQQGEwJVUzELMAkGA\
.
.
GyzN5KK7XR1avMCk9GG0BbpjpqU29Wx3tWpqsh+Kl016Kc=</X509Certificate>
</X509Data>
</KeyInfo>
<md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
</md:KeyDescriptor>
<md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/soap" index="0" isDefault="true"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/slo"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/slo"/>
<md:ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/mnids"/>
<md:ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/mnids"/>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFormat>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/login"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/login"/>
</md:IDPSSODescriptor>
<md:Organization>
<md:OrganizationName xml:lang="en">IBM</md:OrganizationName>
<md:OrganizationDisplayName xml:lang="en">IBM</md:OrganizationDisplayName>
<md:OrganizationURL xml:lang="en"/>
</md:Organization>
<md:ContactPerson contactType="technical">
<md:Company>IBM</md:Company>
<md:GivenName/>
<md:SurName/>
<md:EmailAddress/>
<md:TelephoneNumber/>
</md:ContactPerson>
</md:EntityDescriptor>
```

Verificar status de configuração de SSO

Verifique se a SSO está configurada corretamente. O comando retorna `true` somente quando o SAML está ativado e o arquivo de metadados recebido do servidor SAML corporativo é transferido por upload para o IBM Cloud Private.

```
cloudctl iam saml-status
```

Desativar SAML

Desativar SSO.

```
cloudctl iam saml-disable
```

Configurando a conexão LDAP

Configure uma conexão LDAP (Lightweight Directory Access Protocol) para o cluster do IBM® Cloud Private.

Deve-se conectar um diretório LDAP com seu cluster do IBM Cloud Private. Em seguida, é possível incluir usuários de seu diretório LDAP em seu cluster.

Os tipos de LDAP a seguir são suportados:

- IBM Tivoli Directory Server
- IBM Lotus Domino
- IBM SecureWay Directory Server
- Novell eDirectory
- Sun Java™ System Directory Server
- Netscape Directory Server
- Microsoft Active Directory
- Customizado

Nota: é possível configurar uma política de bloqueio de conta ao configurar o servidor LDAP. Uma política de bloqueio de conta fornece mais segurança restringindo o acesso à conta se diversas tentativas de login falharem.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Conectando-se ao seu diretório LDAP

Siga essas etapas para configurar sua conexão LDAP.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Identidade e acesso**.
3. Clique em **Criar conexão** . A página "Conexão LDAP" é exibida.

4. Insira os detalhes a seguir para configurar sua conexão LDAP.
   Nota: é possível configurar várias instâncias de conexão LDAP para o mesmo servidor LDAP. Entretanto, o `DN Base` e o `Nome` da conexão devem ser exclusivos.

conexão LDAP

Insira informações de conexão.

- **Nome**: um nome exclusivo para a conexão LDAP. Formato: 1 a 50 caracteres alfanuméricos; caracteres especiais que são permitidos: - _
- **Tipo**: um tipo de diretório LDAP ao qual você está se conectando. Selecione na lista. Formato: 1 a 255 caracteres alfanuméricos; espaço em branco é permitido; nenhum caractere especial é permitido.

- **URL**: o nome de domínio do diretório LDAP ou endereço IP e o número da porta LDAP. O nome de domínio deve iniciar com `ldap://`. URL de exemplo: `ldap://corpldap.abc.com:389` ou `ldap://10.10.10.1:389`.

  Para LDAP sobre SSL (LDAPS), deve-se usar o nome de domínio e a URL deve começar com `ldaps://`. URL de exemplo: `ldaps://corpldap.abc.com:636`.

  Nota: se não for possível se conectar ao seu servidor LDAPS usando o nome do host, inclua o endereço IP e o nome do host do servidor LDAPS em seu DNS local. O nome do host do servidor LDAPS deve ser resolvido a partir de seu nó principal do IBM Cloud Private.

Autenticação LDAP

Insira informações sobre autenticação.

- **DN Base**: o nome distinto da base de procura. Exemplo: dc=abc,dc=com. Formato: 1 a 255 caracteres alfanuméricos; caracteres especiais que são permitidos: `= . ,` -
- **DN de Ligação**: o usuário que tem permissão para procurar o DN base. Exemplo: cn=admin,dc=abc,dc=com. Esse parâmetro é opcional. Se nenhum usuário for especificado no parâmetro `Bind DN`, a conexão LDAP será estabelecida sem autenticação. Formato: 0 a 255 caracteres alfanuméricos; espaço em branco é permitido; caracteres especiais que são permitidos: `= . ,` -

- **Senha do DN de ligação**: a senha do usuário que é mencionado no `DN de ligação`. Esse parâmetro não será necessário se você não especificar um usuário no DN de ligação. É permitido um máximo de 255 caracteres.

É possível clicar em **Testar conexão** para verificar se os detalhes da conexão LDAP são válidos.

Filtros LDAP

Insira informações sobre os filtros de procura. Para filtros LDAP padrão por tipo de LDAP, consulte Filtros LDAP padrão por tipo de LDAP.

- **Filtro de grupo**: a cláusula de filtro para procurar grupos. Formato: 1 a 255 caracteres alfanuméricos; caracteres especiais que são permitidos: espaço em branco, `= ; . , & % () {} <> |`
- **Mapa de ID do grupo**: o filtro para mapear um nome do grupo para uma entrada LDAP. Formato: 1 a 255 caracteres alfanuméricos; caracteres especiais que são permitidos: espaço em branco, `* : = ; . , & % () {}`
- **Mapa de ID de membro do grupo**: o filtro para mapear um usuário para um grupo. Formato: 1 a 255 caracteres alfanuméricos; caracteres especiais que são permitidos: espaço em branco, `* : = ; . , & % () {}`
- **Filtro de usuário**: a cláusula de filtro para procurar usuários. Formato: 1 a 255 caracteres alfanuméricos; caracteres especiais que são permitidos: espaço em branco, `= ; . , & % () {} <> |`
- **Mapa de ID do usuário**: o filtro para mapear um nome de usuário para uma entrada LDAP. Formato: 1 a 255 caracteres alfanuméricos; caracteres especiais que são permitidos: espaço em branco, `* : = ; . , & % () {}`

5. Clique em **Conectar**.

O cluster do IBM Cloud Private agora está conectado ao seu diretório LDAP.

Nota: Se estiver usando uma conexão LDAPS, os certificados SSL (Secure Sockets Layer) que são necessários para sua conexão LDAPS são configurados automaticamente quando você se conecta ao seu diretório. No entanto, é preciso reiniciar manualmente o pod `auth-idp`. Conclua estas etapas em seu nó principal:

1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Obtenha os pods `auth-idp`.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

3. Exclua os pods `auth-idp`.

   ```
   kubectl -n kube-system delete pods <pod_name>
   ```

4. Espere a reinicialização dos pods.

   Se sua conexão LDAPS não for bem-sucedida, é possível tentar configurar a conexão manualmente. Para obter informações adicionais, consulte Configurando o LDAP sobre SSL.

Em seguida, é possível incluir seus usuários LDAP e grupos de usuários no cluster do IBM Cloud Private. Para obter mais informações sobre como incluir usuários, consulte Incluir usuários em uma equipe e Incluir grupos em uma equipe.

Filtros LDAP padrão por tipo de LDAP

Tabela 1. Lista de filtros LDAP padrão do IBM Tivoli Directory Server

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)(objectclass=groupOfUniqueNames))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	groupOfUniqueNames:uniquemember
userFilter	sequência	(&(emailAddress=%v)(objectclass=person))
userIdMap	sequência	*:uid

Tabela 2. Lista de filtros LDAP padrão do Servidor Customizado

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)(objectclass=groupOfUniqueNames))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	groupOfUniqueNames:uniquemember
userFilter	sequência	(&(uid=%v)(objectclass=ePerson))
userIdMap	sequência	*:uid

Tabela 3. Lista de filtros LDAP padrão do Microsoft Active Directory

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)(objectclass=group))

Nome do Atributo	Tipo de Dados	Valor Padrão
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	memberOf:member
userFilter	sequência	(&(sAMAccountName=%v)(objectclass=user))
userIdMap	sequência	user:sAMAccountName

Tabela 4. Lista de filtros LDAP padrão do IBM Lotus Domino

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)(objectclass=dominoGroup))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	dominoGroup:member
userFilter	sequência	(&(uid=%v)(objectclass=Person))
userIdMap	sequência	person:uid

Tabela 5. Lista de filtros LDAP padrão do IBM SecureWay Directory Server

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	groupOfNames:member;groupOfUniqueNames:uniqueMember
userFilter	sequência	(&(uid=%v)(objectclass=ePerson))
userIdMap	sequência	*:uid

Tabela 6. Lista de filtros LDAP padrão do Sun Java System Directory Server

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)(objectclass=ldapsubentry))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	nsRole:nsRole
userFilter	sequência	(&(uid=%v)(objectclass=inetOrgPerson))
userIdMap	sequência	inetOrgPerson:uid

Tabela 7. Lista de filtros LDAP padrão do Netscape Directory Server

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	groupOfNames:member;groupOfUniqueNames:uniqueMember
userFilter	sequência	(&(uid=%v)(objectclass=inetOrgPerson))
userIdMap	sequência	inetOrgPerson:uid

Tabela 8. Lista de filtros LDAP padrão do Novell eDirectory

Nome do Atributo	Tipo de Dados	Valor Padrão
groupFilter	sequência	(&(cn=%v)(objectclass=groupOfNames))
groupIdMap	sequência	*:cn
groupMemberIdMap	sequência	groupOfNames:member
userFilter	sequência	(&(cn=%v)(objectclass=Person))
userIdMap	sequência	person:cn

Mudando configurações de cache LDAP

Mudando as configurações de cache do Lightweight Directory Access Protocol (LDAP) que são usadas para autenticação no IBM® Cloud Private.

Os parâmetros de configuração de cache LDAP são listados nas tabelas a seguir. Para obter mais informações, consulte Registro do usuário LDAP (ldapRegistry) .

Nota: especifique um número inteiro positivo seguido por uma unidade de tempo, que pode ser horas (h), minutos (m), segundos (s) ou milissegundos (ms). Por exemplo, especifique 500 milissegundos como 500 ms. É possível incluir diversos valores em uma única entrada. Por exemplo, 1s500ms é equivalente a 1,5 segundos. Esta nota é aplicável apenas aos parâmetros `LDAP_ATTR_CACHE_TIMEOUT` e `LDAP_SEARCH_CACHE_TIMEOUT`.

Parâmetro	Descrição	Valor Padrão
LDAP_ATTR_CACHE_ENABLED	Ative ou desative o cache de atributo LDAP.	true
LDAP_ATTR_CACHE_SIZE	Número de entidades que podem ser armazenadas no cache.	2000
LDAP_ATTR_CACHE_SIZELIMIT	Número máximo de atributos por entidade LDAP que são	

armazenados em cache. |2000| | LDAP_ATTR_CACHE_TIMEOUT|O tempo máximo em que o conteúdo do cache de atributo LDAP está disponível. Quando o tempo especificado decorre, o cache do atributo LDAP é limpo. |1200s|

Parâmetro	Descrição	Valor Padrão
LDAP_SEARCH_CACHE_ENABLED	Ative ou desative o cache de resultados da procura LDAP.	true
LDAP_SEARCH_CACHE_SIZE	Número de resultados da procura que são armazenados no cache.	2000
LDAP_SEARCH_CACHE_SIZELIMIT	O número máximo de resultados que podem ser armazenados em	

cache para uma única procura LDAP. | 2000 | | LDAP_SEARCH_CACHE_TIMEOUT|O tempo máximo em que o conteúdo do cache de resultados da procura está disponível. Quando o tempo especificado decorre, o cache de resultados da procura é limpo. | 1200s|

Mudando os valores de parâmetro usando kubectl

Para mudar os valores de parâmetro, conclua as etapas a seguir:

1. Configure a CLI do `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o ConfigMap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

3. Mude os valores de atributo a seguir, conforme necessário:
   - LDAP_ATTR_CACHE_ENABLED
   - LDAP_ATTR_CACHE_SIZE
   - LDAP_ATTR_CACHE_SIZELIMIT
   - LDAP_ATTR_CACHE_TIMEOUT
   - LDAP_SEARCH_CACHE_ENABLED
   - LDAP_SEARCH_CACHE_SIZE
   - LDAP_SEARCH_CACHE_SIZELIMIT
   - LDAP_SEARCH_CACHE_TIMEOUT
4. Salve e feche o ConfigMap.
5. Reinicie os pods `auth-idp`

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

6. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp`. O status deve ser mostrado como `4/4 Running` para todos os pods.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Mudando os valores de parâmetro usando a console de gerenciamento

1. Efetue login no console como um usuário com acesso de administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps.**
3. Procure por `platform-auth-idp`.
4. Clique em **...** > `Editar`.
5. Mude os valores de atributo a seguir, conforme necessário:
   - LDAP_ATTR_CACHE_ENABLED
   - LDAP_ATTR_CACHE_SIZE
   - LDAP_ATTR_CACHE_SIZELIMIT
   - LDAP_ATTR_CACHE_TIMEOUT
   - LDAP_SEARCH_CACHE_ENABLED
   - LDAP_SEARCH_CACHE_SIZE
   - LDAP_SEARCH_CACHE_SIZELIMIT
   - LDAP_SEARCH_CACHE_TIMEOUT
6. Clique em **Enviar**.

7. No menu de navegação, clique em **Cargas de trabalho > DaemonSets**.
8. Localize `auth-idp`.
9. Clique em **...** > `Editar`. Uma janela `Editar DaemonSet` é exibida.
10. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é para recarregar os pods `auth-idp` com os valores de ConfigMap mais recentes.
11. Clique em `auth-idp`.
12. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp` na área de janela **Pods**. O status de todos os pods deve ser mostrado como `4/4` sob o nome do campo **Pronto**.

Mudando configurações de procura LDAP

Mudando as configurações de procura do Lightweight Directory Access Protocol (LDAP) no IBM® Cloud Private.

Configure a consideração de atributos para procura LDAP

Por padrão, o IBM Cloud Private procura por usuários e grupos nos atributos de nome comum (CN) e de identificador (ID), conforme definido nas configurações de conexão LDAP. Exemplos de atributos de ID são `uid`, `sAMAccountName` e `emailAddress`.

Esse comportamento pode ser mudado definindo os parâmetros de configuração a seguir:

Os dois atributos a seguir são usados para a procura:

- `LDAP_SEARCH_CN_ATTR_ONLY`: Procurar por CN apenas. O valor padrão é `false`.
- `LDAP_SEARCH_ID_ATTR_ONLY`: Procurar por ID apenas. O valor padrão é `false`.

Para usar apenas o atributo CN para procurar por usuários ou grupos, configure o valor `LDAP_SEARCH_CN_ATTR_ONLY` como `true`. Para usar apenas o atributo de ID para procurar por usuários ou grupos, configure o valor `LDAP_SEARCH_ID_ATTR_ONLY` como `true`.

A seguir estão as etapas para mudar os valores dos parâmetros de configuração de procura LDAP acima:

Mudando os valores de atributos usando kubectl

Para mudar os valores, conclua as etapas a seguir:

1. Configure a CLI do `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o ConfigMap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

   Configure os valores de atributo a seguir, conforme necessário, como `true` ou `false`.

   - `LDAP_SEARCH_CN_ATTR_ONLY`
   - `LDAP_SEARCH_ID_ATTR_ONLY`
     Nota: configure apenas um valor como `true`.
3. Salve e feche o ConfigMap.
4. Reinicie os pods `auth-idp`

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```
5. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp`. O status deve ser mostrado como `4/4 Running` para todos os pods.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Mudando os valores de atributo usando a console de gerenciamento

1. Efetue login no console como um usuário com acesso de administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps.**
3. Procure por `platform-auth-idp`.
4. Clique em **...** > `Editar`.
5. Mude um dos valores de atributo a seguir para `true`, conforme necessário:
   - `LDAP_SEARCH_CN_ATTR_ONLY`

- `LDAP_SEARCH_ID_ATTR_ONLY` **Nota:** configure apenas um valor como `true`.

6. Clique em **Enviar**.
7. No menu de navegação, clique em **Cargas de trabalho > DaemonSets**.
8. Localize `auth-idp`.
9. Clique em **...** > `Editar`. Uma janela `Editar DaemonSet` é exibida.
10. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é para recarregar os pods `auth-idp` com os valores de ConfigMap mais recentes.
11. Clique em `auth-idp`.
12. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp` na área de janela **Pods**. O status de todos os pods deve ser mostrado como `4/4` sob o nome do campo **Pronto**.

Mudando o limite de tamanho da procura e os valores de limite de tempo LDAP

A seguir estão os parâmetros de configuração do limite de procura LDAP:

- `LDAP_SEARCH_SIZE_LIMIT`: o valor padrão é "50". O intervalo de valores é de 50 a 100.
- `LDAP_SEARCH_TIME_LIMIT`: o valor padrão é "5". O intervalo de valores é de 5 a 50.

Mudando os valores usando kubectl

Para mudar os valores, conclua as etapas a seguir:

1. Configure a CLI do `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o ConfigMap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

   Mude os valores a seguir, conforme necessário:

   - `LDAP_SEARCH_SIZE_LIMIT`
   - `LDAP_SEARCH_TIME_LIMIT`

3. Salve e feche o ConfigMap.
4. Reinicie os pods `auth-idp`

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

5. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp`. O status deve ser mostrado como `4/4 Running` para todos os pods.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Mudando os valores usando a console de gerenciamento

1. Efetue login no console como um usuário com acesso de administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps.**
3. Procure por `platform-auth-idp`.
4. Clique em **...** > `Editar`.
5. Mude os valores de atributo a seguir, conforme necessário:
   - `LDAP_SEARCH_SIZE_LIMIT`
   - `LDAP_SEARCH_TIME_LIMIT`
6. Clique em **Enviar**.
7. No menu de navegação, clique em **Cargas de trabalho > DaemonSets**.
8. Localize `auth-idp`.
9. Clique em **...** > `Editar`. Uma janela `Editar DaemonSet` é exibida.
10. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é para recarregar os pods `auth-idp` com os valores de ConfigMap mais recentes.
11. Clique em `auth-idp`.
12. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp` na área de janela **Pods**. O status de todos os pods deve ser mostrado como `4/4` sob o nome do campo **Pronto**.

Mude o uso de caracteres curinga de procura LDAP

A seguir está o parâmetro de configuração de caracteres curinga de procura LDAP:

- `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`: o valor padrão é "false".

No IBM Cloud Private, caracteres curinga, como asterisco, são usados na sequência de procura para corresponder ao valor de subsequência do atributo do servidor LDAP.

Configure o valor `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS` como `true` para excluir caracteres curinga (*) na sequência de procura e para corresponder ao valor exato do atributo do servidor LDAP.

A seguir está a sequência de procura padrão:

```
(|(&(cn=*<searchstring>*)(objectclass=person))(&(uid=*<searchstring>*)(objectclass=person)))
```

A seguir está a sequência de procura quando o `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS` é configurado como `true`:

```
(|(&(cn=<searchstring>)(objectclass=person))(&(uid=<searchstring>)(objectclass=person)))
```

Para mudar o uso de caracteres curinga na sequência de procura, conclua as etapas a seguir:

Mudando o parâmetro de configuração de caracteres curinga usando kubectl

Para mudar os valores, conclua as etapas a seguir:

1. Configure a CLI do `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o ConfigMap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

   Mude o valor a seguir, conforme necessário:

   - `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`
3. Salve e feche o ConfigMap.
4. Reinicie os pods `auth-idp`

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```
5. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp`. O status deve ser mostrado como `4/4 Running` para todos os pods.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Mudando o parâmetro de configuração de caracteres curinga usando a

console de gerenciamento

1. Efetue login no console como um usuário com acesso de administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps.**
3. Procure por `platform-auth-idp`.
4. Clique em **...** > `Editar`.
5. Mude o valor a seguir, conforme necessário:
   - `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`
6. Clique em **Enviar**.
7. No menu de navegação, clique em **Cargas de trabalho > DaemonSets**.
8. Localize `auth-idp`.
9. Clique em **...** > `Editar`. Uma janela `Editar DaemonSet` é exibida.
10. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é para recarregar os pods `auth-idp` com os valores de ConfigMap mais recentes.
11. Clique em `auth-idp`.
12. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp` na área de janela **Pods**. O status de todos os pods deve ser mostrado como `4/4` sob o nome do campo **Pronto**.

Mudando a propriedade de configuração do Logjam

O parâmetro de configuração do Logjam é `LOGJAM_DHKEYSIZE_2048_BITS_ENABLED`. O valor padrão é `LOGJAM_DHKEYSIZE_2048_BITS_ENABLED: true`.

É possível configurar o valor de parâmetro como `true` ou `false`.

Configure `LOGJAM_DHKEYSIZE_2048_BITS_ENABLED` como `true` para evitar o ataque de vulnerabilidade de segurança Logjam para desaprovar a troca de chave SHA-1 e Diffie-Hellman (DH) que seja menor que 2.048 bits.

A seguir estão as etapas para mudar o valor:

Mudando o valor de parâmetro usando kubectl

1. Configure a CLI do `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o ConfigMap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

3. Mude os valores de atributo a seguir, conforme necessário:
   - Configure `LOGJAM_DHKEYSIZE_2048_BITS_ENABLED` como `true` ou `false`.
4. Salve e feche o ConfigMap.
5. Reinicie os pods `auth-idp`

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

6. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp`. O status deve ser mostrado como `4/4 Running` para todos os pods.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Mudando os valores de parâmetro usando a console de gerenciamento

1. Efetue login no console como um usuário com acesso de administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps.**
3. Procure por `platform-auth-idp`.
4. Clique em **...** > `Editar`.
5. Mude o valor de parâmetro `LOGJAM_DHKEYSIZE_2048_BITS_ENABLED` para `true` ou `false`.
6. Clique em **Enviar**.
7. No menu de navegação, clique em **Cargas de trabalho > DaemonSets**.
8. Localize `auth-idp`.
9. Clique em **...** > `Editar`. Uma janela `Editar DaemonSet` é exibida.
10. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é para recarregar os pods `auth-idp` com os valores de ConfigMap mais recentes.
11. Clique em `auth-idp`.
12. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp` na área de janela **Pods**. O status de todos os pods deve ser mostrado como `4/4` sob o nome do campo **Pronto**.

IAM para cargas de trabalho do IBM Cloud Private

Várias cargas de trabalho de conteúdo são integradas na plataforma do IBM Cloud Private por meio de implementações do Helm. Essas cargas de trabalho podem alavancar serviços do IBM Cloud Private para o Identity & Access Management (IAM) para configurar a Conexão Única (SSO) para seus serviços. Este tópico explica como as cargas de trabalho de conteúdo podem ativar a autenticação e a autorização para seus serviços usando o IAM e os recursos que eles podem usar no IBM Cloud Private para alguns de seus casos de uso.

Para uma carga de trabalho típica que é integrada à plataforma IBM Cloud Private, deve-se executar as seguintes etapas para configurar a SSO:

1. Integração de autenticação por meio do registro do cliente
   - Obtenção do segredo OAUTH para registro
   - Construção da carga útil do registro do cliente
   - Chamada da API de registro do cliente
2. Aplicação de autenticação por cargas de trabalho

    - Autenticação da IU
    - Autenticação da CLI

3. Integração da autorização

    - Serviço de integração
    - Gateway de Controle de Acesso da API
    - Tipo de serviço de integração
    - Administração de controle de acesso precisa por meio de equipes
    - Aplicação da autorização

4. Configure casos de uso não orientados pelo usuário

Para obter detalhes, consulte Integração de autenticação e conexão única e Integração de autorização, administração e cumprimento.

IAM para comunicação de serviço para serviço

O IBM Cloud Identity and Access Management (IAM) permite a capacidade de criar IDs de serviço e chaves da interface de programação de aplicativos (API) para IDs de serviço. Um ID de serviço é semelhante a um ID funcional ou a um ID do aplicativo e é usado para autenticar serviços, e não para representar um usuário.

É possível criar IDs de serviço e ligá-los ao namespace do escopo, como uma conta do IBM Cloud, uma organização do IBM Cloud Private Cloud Foundry ou um espaço do IBM Cloud Private Cloud Foundry. No entanto, para adotar o IBM Cloud IAM, é melhor ligar os IDs de serviço a uma conta do IBM Cloud. Esta ligação é feita para fornecer ao ID de serviço um contêiner no qual residir. Esse contêiner também define quem pode atualizar e excluir o ID de serviço e quem pode criar, atualizar, ler e excluir chaves de API que estão associadas a esse ID de serviço. É importante observar que um ID de Serviço não está relacionado a um usuário.

Considere um exemplo em que um serviço de conteúdo, como o WebSphere Application Server (WAS), se comunica com o serviço de medição usando um ID de serviço.

1. O serviço de medição cria um ID de serviço e uma chave de API de um usuário com privilégios necessários.
2. O serviço de medição cria políticas para o ID do serviço.
3. O usuário compartilha ou publica a chave de API no WAS.
4. O WAS chama as APIs do serviço de medição usando a chave de API.
5. O serviço de medição obtém um token de acesso para a chave de API.
6. O serviço de medição faz introspecção do token para obter o ID de serviço do IAM.
7. O serviço de medição valida o ID de serviço.
8. O serviço de medição conclui a operação da API e retorna uma resposta ao WAS.

Conclua as seguintes etapas para ativar a comunicação de serviço para serviço:

1. Criar ID de serviço e chave de API
2. Criar políticas para o ID de serviço
3. Obter token de acesso para a chave API
4. Fazer introspecção do token de acesso para obter o ID de serviço do IAM

Criar ID de serviço e chave de API

Existem quatro maneiras de criar um ID de serviço e uma chave de API para qualquer serviço.

- Criação automatizada de ID de serviço e de chave de API usando anotações secretas do Kubernetes.
- Criar ID de serviço e chave de API usando a interface da linha de comandos (CLI) do IBM Cloud Private. Para obter mais informações, consulte Gerenciando seu cluster com a CLI do IBM Cloud Private (cloudctl).
- Criar ID de serviço e chave de API usando a console de gerenciamento.
- Criar ID de serviço e chave de API usando as APIs.

Criação automatizada de ID de serviço e de chave de API usando anotações secretas do Kubernetes

Para gerar um ID de serviço e uma chave de API para qualquer serviço, é necessário criar um segredo do Kubernetes com as três anotações a seguir:

- `ibm.com/iam-service.name`, que é um nome exclusivo do serviço.
- `ibm.com/iam-service.id`, que é um nome da chave que é injetada nesse segredo junto com o ID do serviço.

- `ibm.com/iam-service.api-key`, que é o nome da chave que é injetada neste segredo junto com a chave de API.

A seguir está um modelo de exemplo de um segredo:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: myservice-secret
  namespace: mynamespace
  annotations:
    ibm.com/iam-service.name: "myservice-service"
    ibm.com/iam-service.id: "myservice-service-id"
    ibm.com/iam-service.api-key: "myservice-api-key"
data:
  ...
```

O `myservice-secret` é criado com as anotações `ibm.com/myservice-service.id`, `ibm.com/myservice-service.api-key` e `ibm.com/myservice-service.name`. O serviço `secret-watcher` atualiza o recurso secreto que contém essas anotações e injeta os detalhes do ID de serviço e da chave de API na seção de dados do segredo.

Saída de exemplo do segredo:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: myservice-secret
  namespace: mynamespace
  annotations:
    ibm.com/iam-service.id: "myservice-service-id"
    ibm.com/iam-service.api-key: "myservice-api-key"
    ibm.com/iam-service.name: "myservice-service"
data:
  myservice-service-id: cb0719e2-3480-11e9-b210-d663bd873d93
  myservice-api-key: 20346eed-8e01-47e8-b4f8-1efe6fec2408
  ...
```

Qualquer serviço agora pode ler o ID de serviço e a chave de API a partir do segredo.

Criar ID de serviço e chave de API usando a CLI do IBM Cloud Private

Siga estas etapas para criar o ID de serviço e a chave de API:

1. Crie um ID de serviço.

   Consulte cloudctl iam service-id-create.

2. Criar uma chave API.

   Consulte cloudctl iam service-api-key-create.

Criar ID de serviço e chave de API usando a console de gerenciamento

Consulte Criando um ID de serviço usando a IBM Cloud Privateconsole de gerenciamento.

Criar ID de serviço e chave de API usando as APIs

Siga estas etapas para criar o ID de serviço e a chave de API:

1. Crie o ID do serviço.

   Consulte Criar um ID de serviço.

2. Criar uma chave API.

   Consulte Criar uma chave de API.

Criar políticas para o ID de serviço

Existem três maneiras de criar políticas para o ID de serviço.

- Criar políticas para o ID de serviço usando a CLI do IBM Cloud Private
- Criar políticas para o ID de serviço usando a console de gerenciamento
- Criar políticas para o ID de serviço usando as APIs

Criar políticas para o ID de serviço usando a CLI do IBM Cloud Private

Consulte cloudctl iam service-policy-create.

Criar políticas para o ID de serviço usando a console de gerenciamento

Consulte Criando um ID de serviço usando a IBM Cloud Privateconsole de gerenciamento.

Criar políticas para o ID de serviço usando as APIs

Consulte Criar uma política de acesso para um ID de serviço.

Obter token de acesso para a chave de API

Use a seguinte API para obter um token para um ID de serviço e chave de API:

Gerar um token do OpenID Connect (OIDC).

Fazer introspecção do token de acesso para obter o ID de serviço do IAM

Use a seguinte API para fazer introspecção do token de acesso:

Fazer introspecção de um token OIDC

ID de serviço e equipes

Um aplicativo ou um serviço usa um ID de serviço para chamar as APIs de vários microsserviços. O ID de serviço pode receber acesso somente ao conjunto de serviços que são requeridos pelo aplicativo ou serviço. Cada aplicativo pode ter sua própria combinação de ID de serviço e chave de API, o que permite uma fácil rotação de uma chave, sem afetar outros aplicativos ou usuários.

Os administradores de cluster podem designar um ID de serviço a uma equipe e gerenciar o ID de serviço. Os usuários com função de administrador podem criar e excluir chaves de API e podem acessar políticas que são designadas a um ID de serviço. O ID de serviço deve estar ligado a pelo menos um namespace. Se você não designar um namespace ao criar um ID de serviço, a criação do ID de serviço falhará.

Para obter informações sobre como designar um ID de serviço a uma equipe, consulte Criando um ID de serviço usando a IBM Cloud Privateconsole de gerenciamento.

Integração do IAM com o OpenShift

Este tópico contém os detalhes da implementação da integração de autenticação e autorização do IBM Cloud Private com o Red Hat OpenShift. O OpenShift é integrado à autenticação de estruturas do IBM Cloud Private e de autorização.

Considerações de design

1. O OpenShift usa projetos, enquanto que o IBM Cloud Private usa namespaces. Os projetos e namespaces são tratados no estado em que se encontram no OpenShift e no IBM Cloud Private. Nenhum mapeamento cruzado é feito automaticamente.
2. O servidor de API do Kube do OpenShift não é executado na porta 8001. A console de gerenciamento do IBM Cloud Private usa atualmente a porta 8001 na página `Configurar Cliente`.
3. A administração de autenticação e administração acontece por meio do IBM Cloud Private. O fluxo é apenas do IBM Cloud Private para o OpenShift. Se um usuário for criado no OpenShift, ele não fluirá de volta para o IBM Cloud Private.

Autenticação

Integração com o provedor OIDC do IBM Cloud Private

A autenticação do IBM Cloud Private é suportada pelo seu provedor OIDC que é executado no WebSphere Liberty. A autenticação do OpenShift é integrada com o serviço de autenticação do IBM Cloud Private, conforme descrito no tópico OpenID Connect. Essa configuração de serviço de autenticação na qual o OpenShift usa a autenticação do OIDC do IBM Cloud Private ocorre automaticamente no término da instalação dos serviços de segurança no IBM Cloud Private.

Integração do LDAP

A autenticação do usuário do IBM Cloud Private depende grandiosamente de sua integração com o servidor Enterprise Lightweight Directory Access Protocol (ELDAP). O provedor OIDC IBM Cloud Private é suportado pelo servidor Liberty. Portanto, quando o servidor Liberty é configurado com um LDAP por meio do IBM Cloud Private, a autenticação também deve funcionar sem problemas do console do OpenShift. Isso significa que os usuários que são definidos no LDAP podem efetuar login no OpenShift e são autenticados pelos servidores OIDC e Liberty do IBM Cloud Private depois que a integração do OIDC é obtida.

Grupos de usuários

Com o IBM Cloud Private, é possível importar grupos de usuários do LDAP. O OpenShift também suporta o grupo de usuários determinando um registro do grupo de usuários que é criado no OpenShift. A integração do IBM Cloud Private with OpenShift manipula a criação do respectivo registro de grupo de usuários no OpenShift internamente quando administrado por meio da página IBM Cloud Private`Administração da equipe`. O código a seguir é um exemplo do registro do grupo de usuários do OpenShift:

```
apiVersion: v1
kind: Group
metadata:
  annotations:
    openshift.io/ldap.sync-time: 2015-10-13T10:08:38-0400
    openshift.io/ldap.uid: cn=admins,ou=groups,dc=example,dc=com
    openshift.io/ldap.url: LDAP_SERVER_IP:389
  creationTimestamp:
  name: Administrators
users:
- jane.smith@example.com
- jim.adams@example.com
```

Tokens

O IBM Cloud Private funciona com dois tokens:

- Token de acesso: isso é fornecido pelo servidor Liberty e é usado para autenticação no IBM Cloud Private.
- Token de identidade: este token é gerado pelo `platform-identity-provider` do IBM Cloud Private e é usado para autenticação e autorização para o servidor da API do Kube e para a CLI kubectl.

Após a integração do OpenShift para usar nosso serviço OIDC, o token de identidade é substituído pelo token do OpenShift no `platform-identity-provider`. Chamadas para o servidor da API Kube e para a CLI kubectl continuam funcionando sem requerer mudanças na IU para o IBM Cloud Private e o OpenShift.

Impactos de autenticação na IU

- CLI do IBM Cloud Private: o login funcionará no estado em que se encontra sem mudanças.
- IBM Cloud Private console de gerenciamento: Nenhum impacto. O token de acesso está disponível no cookie e o token de identidade está disponível por meio da API `identitytoken`. As chamadas do IAM com o token de acesso e as chamadas do Kube com o token de identidade continuam funcionando no estado em que se encontram.
- CLI do OpenShift: a CLI do OpenShiftatualmente não suporta a integração de login com o provedor OIDC. Isso é devido à autenticação no OpenShift por meio do provedor OIDC que acontece somente por meio do OAUTH Dance no navegador. Para efetuar login na CLI do OpenShift, é necessário obter o token por meio da GUI. Por exemplo, o menu `Copy Login Command` da GUI do OpenShift e o menu `Configure Client` na console de gerenciamento do IBM Cloud Private.
- GUI do OpenShift: Nenhum impacto. O trabalho do token funciona no estado em que se encontra.

Autorização

O IBM Cloud Private integra dois modelos de autorização denominados IAM do Cloud e RBAC do Kubernetes. A integração do IBM Cloud Private with OpenShift continua a suportar ambos os modelos de autorização.

A autorização do Kubernetes no IBM Cloud Private é acionada pela informação de ligação de função baseada em equipe que está presente no token de identidade. Como não há nenhum conceito de equipe no OpenShift, a autorização do Kubernetes é orientada

pela ligação de função de Cluster e pela ligação da função Projeto e Namespace para o usuário e grupos de usuários. Esse modelo funciona bem com o IBM Cloud Private e o OpenShift.

Mapeamento de função

Funções do OpenShift

- cluster-admin
- admin
- edit
- view
- deployer
- image-builder
- image-puller

Mapeando funções do IBM Cloud Private

- Administrador do cluster (cluster-admin)
- Administrador (admin)
- Operador (edit)
- Editor (edit)
- Visualizador (view)

As ligações de função do cluster e as ligações de função são criadas para usuários e grupos listados. O mapeamento de função do IBM Cloud Private para o OpenShift é transparente para o usuário.

Impactos de autorização na IU

- CLI do IBM Cloud Private: a Autorização continua funcionando no estado em que se encontra com o token de identidade. Nenhum impacto.
- IBM Cloud Private console de gerenciamento: Nenhum impacto. O token de identidade está disponível por meio da API `identitytoken`. As chamadas do IAM com o token de acesso e as chamadas do Kube com o token de identidade continuam funcionando no estado em que se encontram.
- CLI do OpenShift: Nenhum impacto. O token de identidade pode ser buscado a partir da GUI. Por exemplo, o menu `Copy Login Command` da GUI do OpenShift e o menu `Configure Client` na console de gerenciamento do IBM Cloud Private. O restante funciona no estado em que se encontra.
- GUI do OpenShift: Nenhum impacto. O token funciona no estado em que se encontra sem mudanças e a autorização necessária é cumprida.

Resolução de problemas do IAM

Resolva problemas do Identity and Access Management (IAM).

- Efetuar login
- LDAP
- Problemas do pod

Isolamento no IBM Cloud Private

O IBM Cloud Private oferece suporte de ocupação variada por meio de isolamento de usuário, de cálculo e de rede dentro de um cluster. Os recursos físicos e lógicos dedicados são necessários para que o cluster atinja o isolamento da carga de trabalho. A ocupação variada requer a aplicação de várias técnicas de isolamento que são descritas neste tópico.

O isolamento de usuário, de cálculo e de rede é aplicado restringindo as implementações de carga de trabalho para recursos físicos e virtuais. O isolamento cumprido também permite que o administrador do cluster controle a área de cobertura que é alocada para várias equipes com base em seus requisitos. A seguir estão alguns pré-requisitos chave para atingir o isolamento de implementações em nós do cluster.

Planejar modelo de implementação para isolamento

Há vários níveis de ocupação variada. O administrador de cluster deve analisar os requisitos de carga de trabalho para determinar quais níveis são necessários. Os recursos de isolamento a seguir podem ser usados para satisfazer esses requisitos:

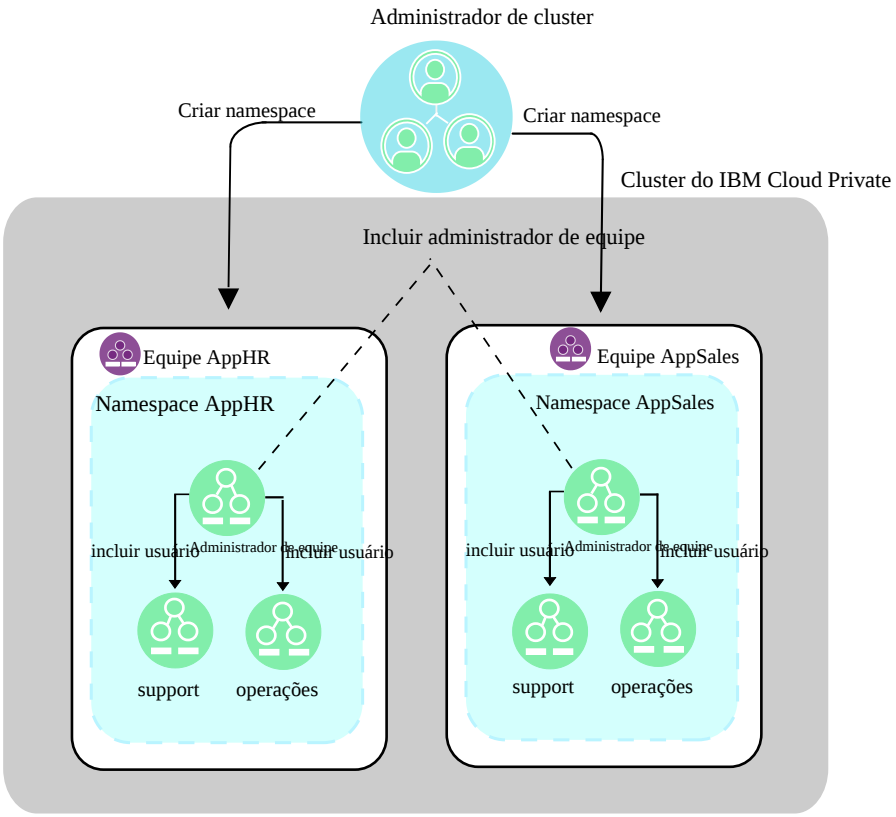
- **Grupos de hosts**: como parte da configuração de pré-instalação, o administrador de cluster pode configurar grupos de nós para grupos de hosts do trabalhador e grupos de hosts do proxy. Esta operação também envolve o pré-planejamento dos namespaces, já que cada grupo de hosts é mapeado para um namespace.
- **Sub-rede de VLAN**: o administrador de infraestrutura de rede pode planejar vários intervalos de sub-rede para cada nó ou grupos de hosts antes da instalação do IBM Cloud Private.
- **Suporte a múltiplos LDAPs**: múltiplos servidores LDAP podem ser configurados e o administrador de cluster pode formar equipes de usuários e de grupos de usuários a partir de vários domínios LDAP.
- **Namespaces**: o administrador de cluster pode criar namespaces para agrupamento lógico de recursos. Os namespaces podem ser criados após a instalação do IBM Cloud Private. Se o administrador de cluster optar por ter grupos de hosts, o planejamento de namespace será feito antes da instalação.
- **Controladores de ingresso de rede**: o administrador de cluster deve planejar os controladores de ingresso antes da instalação para permitir que o instalador crie controladores de ingresso para cada controlador que esteja mapeado para um grupo de hosts e para um namespace.
- **Usuários, grupos de usuários, equipes**: os usuários e grupos de usuários podem ser integrados em uma plataforma do IBM Cloud e também agrupados em equipes que estejam mapeadas para namespaces e outros recursos.
- **Políticas de rede**: os administradores e operadores da Equipe podem criar políticas de rede para criar regras de firewall no escopo do namespace.
- **Políticas de segurança de pod**: o administrador de cluster pode criar políticas que permitem ou negam que as imagens de contêiner sejam executadas em namespaces ou nós de seleção.

Cenários

A seção a seguir abrange vários cenários de caso de uso e possui mais detalhes sobre a configuração para atingir o isolamento em cada cenário.

Cenário 1: equipes e namespaces isolados entre usuários e grupos de usuários

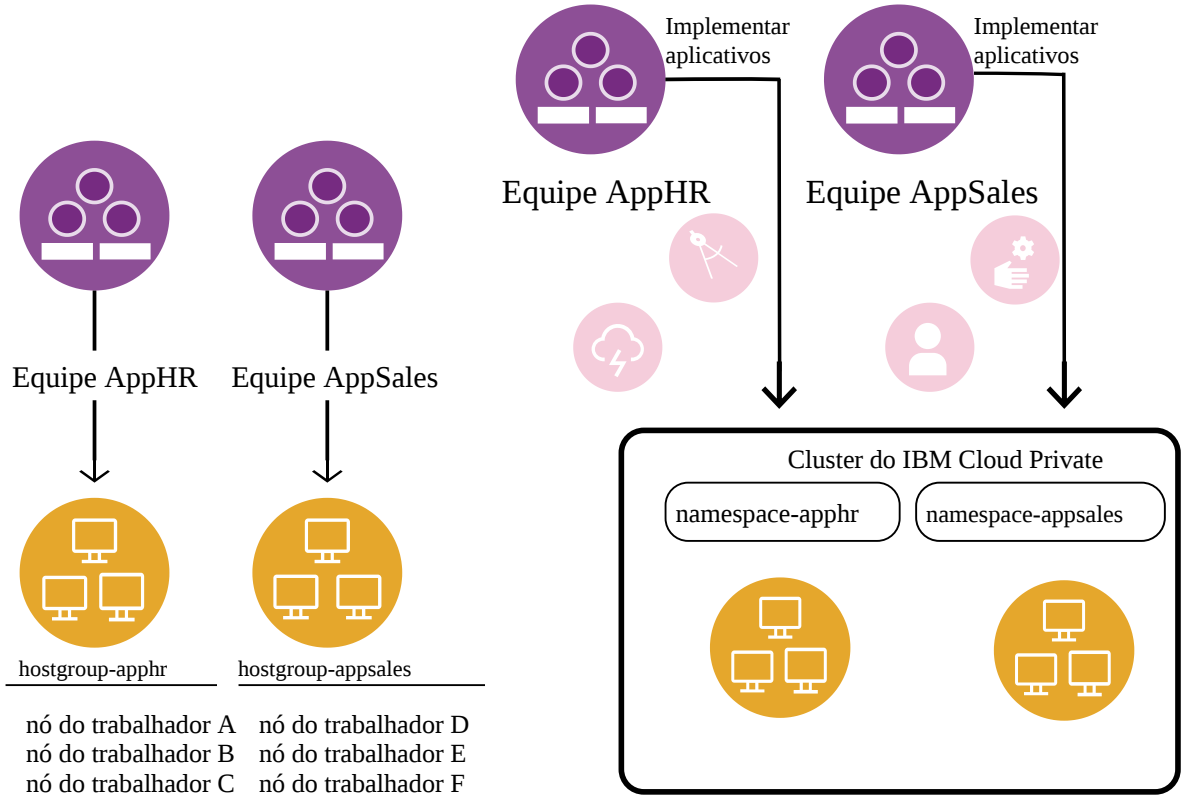
Você deseja usar um único cluster do IBM Cloud Private para múltiplas equipes do projeto do aplicativo com isolamento completo. Pode haver uma equipe de Operações que gerencia um único cluster com a alta disponibilidade configurada. Os membros da equipe de Operações definem namespaces individuais para equipes de projeto separadas. Eles designam a função de administrador para uma ou mais pessoas por equipe do projeto, que, por sua vez, definem os membros e as funções da equipe para esse namespace. Os membros da equipe de Operações que possuem a função do administrador do cluster são responsáveis pela configuração do armazenamento compartilhado e pelo monitoramento de todos os namespaces.



- **Planejamento**
  - Planeje os grupos de usuários e grupos de usuários em várias equipes
- **Pôster de**
  - Instale o IBM Cloud Private
- **Pós-instalação**
  - Mapeie as equipes para um ou mais Namespaces com base no requisito de agrupamento de recursos
  - Crie equipes e designe usuários e recursos; consulte Equipes para obter mais informações

Cenário 2: equipes e namespaces isolados e nós dedicados para namespaces

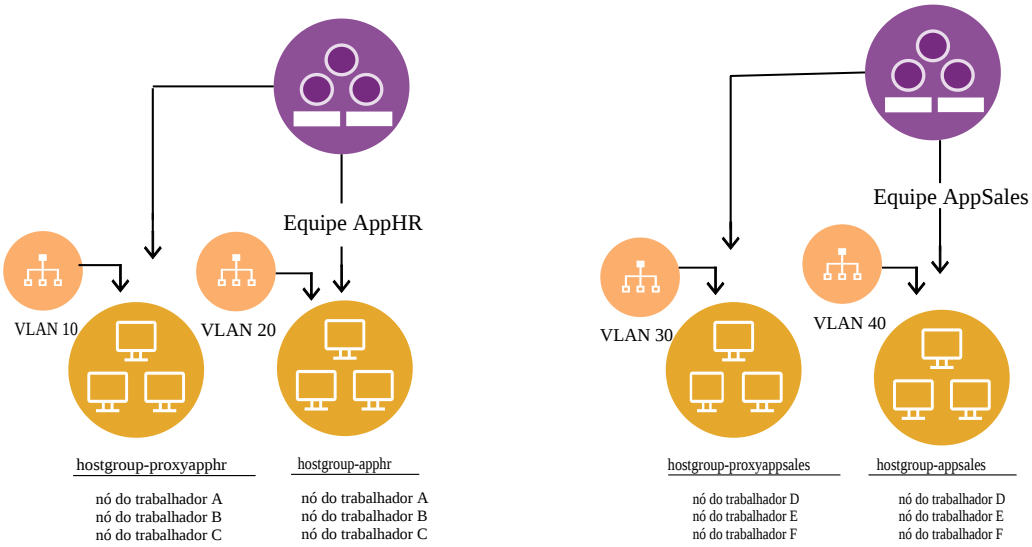
Há múltiplas equipes em sua organização com namespaces dedicados para cada equipe. As implementações de aplicativos e cargas de trabalho de qualquer equipe devem acontecer somente dentro do namespace designado à equipe. Cada equipe possui um grupo planejado de nós, que são servidores físicos ou máquinas virtuais, que são incluídos no cluster. Esse cluster pode ter muitos outros nós de várias equipes que são configurados e gerenciados como parte do mesmo cluster do IBM Cloud Private. Todas as implementações que são feitas por uma equipe específica devem ser hospedadas somente no grupo de nós ao qual a equipe foi designada.

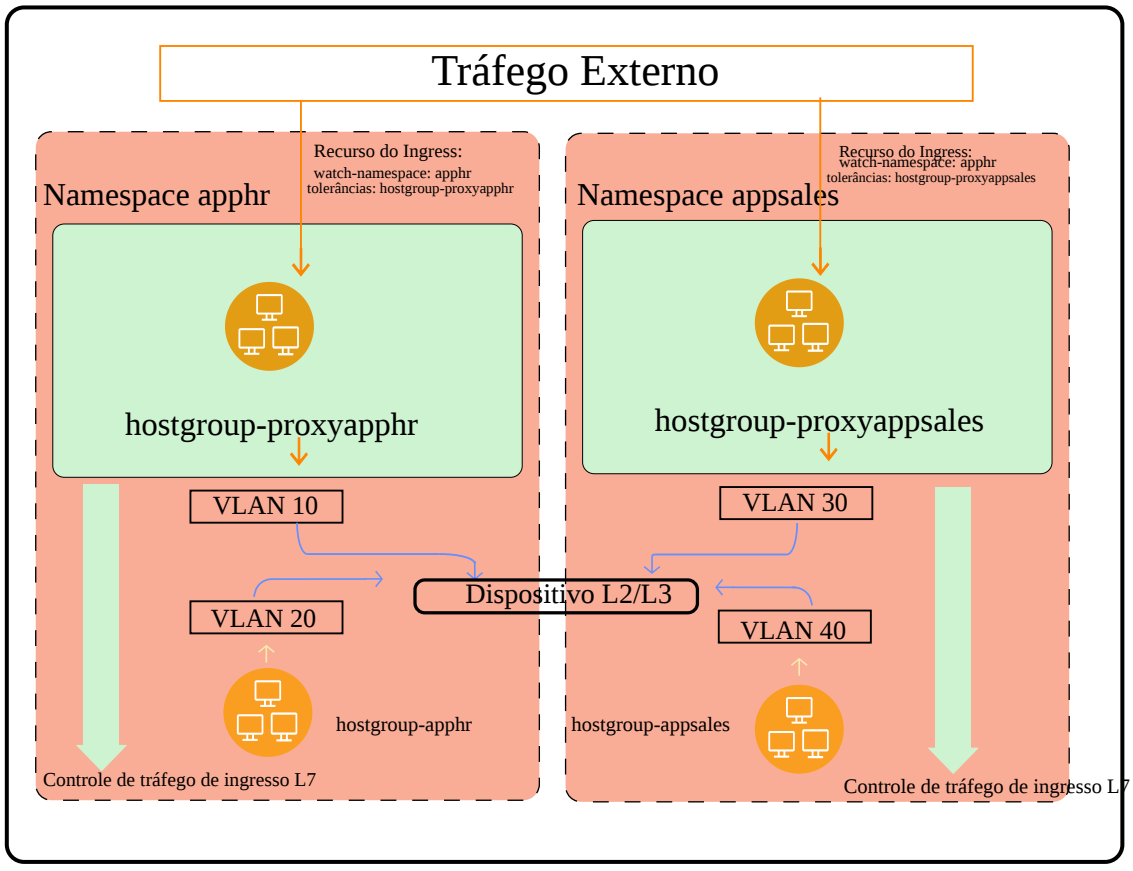


- **Planejamento**
  - Planeje os grupos de usuários e grupos de usuários em várias equipes
  - Planeje vários grupos de hosts customizados de nós do trabalhador
  - Planeje namespaces antecipadamente, que serão mapeados para cada grupo de hosts do trabalhador
- **Pôster de**
  - Configure os grupos de hosts customizados para os trabalhadores no arquivo host
  - Configure namespaces, que serão mapeados para cada grupo de hosts que está configurado em `<installation_directory>/cluster/config.yaml` como um parâmetro `isolated_namespaces`
  - Instale o IBM Cloud Private
- **Pós-instalação**
  - Mapeie equipes para um ou mais namespaces com base no requisito de agrupamento de recursos
  - Crie equipes e designe usuários e recursos; consulte Equipes para obter mais informações

Cenário 3: equipes e namespaces isolados, nós dedicados para namespaces, redes VLAN dedicadas e namespaces controlados por ingresso

As cargas de trabalho são executadas em uma infraestrutura de rede compartilhada com um cluster no qual várias equipes da organização têm cargas de trabalho em seus namespaces isolados dedicados. O tráfego de rede para os aplicativos no namespace de um cluster não deve ter nenhuma interferência no tráfego de rede em outros namespaces e todo o tráfego deve ser confinado no namespace da implementação. Além disso, os nós podem ser agrupados e designados com um intervalo de sub-rede VLAN dedicado.

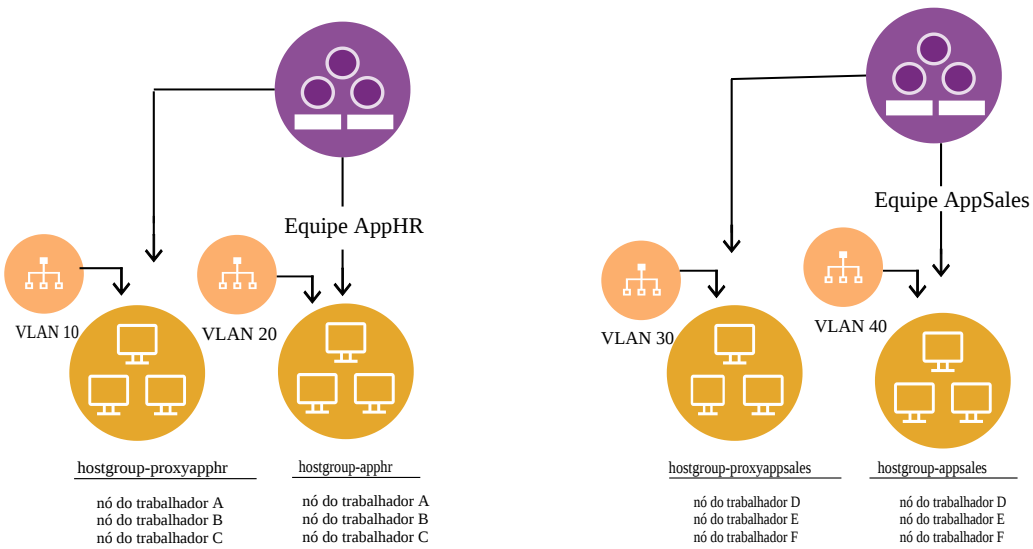

Equipe AppHR
VLAN 10
VLAN 20
hostgroup-proxyapphr
hostgroup-apphr
nó do trabalhador A
nó do trabalhador B
nó do trabalhador C
nó do trabalhador A
nó do trabalhador B
nó do trabalhador C
Equipe AppSales
VLAN 30
VLAN 40
hostgroup-proxyappsales
hostgroup-appsales
nó do trabalhador D
nó do trabalhador E
nó do trabalhador F
nó do trabalhador D
nó do trabalhador E
nó do trabalhador F

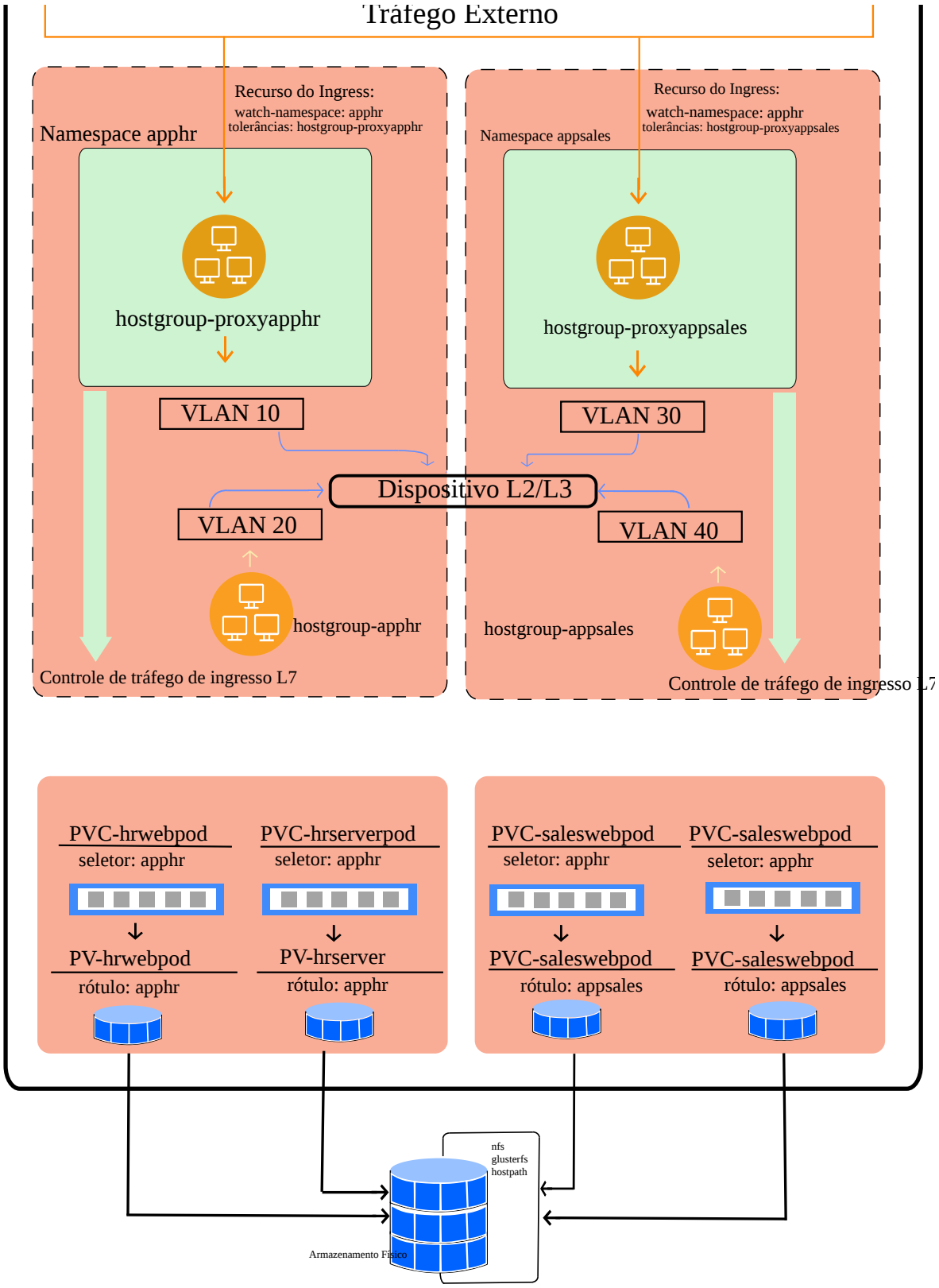

Tráfego Externo
Recurso do Ingress:
watch-namespace: apphr
tolerâncias: hostgroup-proxyapphr
Namespace apphr
hostgroup-proxyapphr
VLAN 10
VLAN 20
hostgroup-apphr
Controle de tráfego de ingresso L7
Recurso do Ingress:
watch-namespace: apphr
tolerâncias: hostgroup-proxyappsales
Namespace appsales
hostgroup-proxyappsales
VLAN 30
VLAN 40
hostgroup-appsales
Controle de tráfego de ingresso L7
Dispositivo L2/L3

- **Planejamento**
  - Planeje os grupos de usuários e grupos de usuários em várias equipes
  - Planeje vários grupos de hosts customizados para nós de proxy e de trabalhador
  - Planeje os mapeamentos de namespace para grupos de hosts de proxy e de trabalhador
  - Planeje sub-redes de VLAN para cada grupo de hosts
- **Pôster de**
  - Configure os grupos de hosts customizados para trabalhadores e nós de proxy nos arquivos host
  - Configure namespaces, que serão mapeados para cada grupo de hosts do trabalhador configurado em `<installation_directory>/cluster/config.yaml`
  - Configure namespaces, que serão mapeados para cada grupo de hosts de proxy configurado em `<installation_directory>/cluster/config.yaml` como um parâmetro `isolated_proxies`
  - Configure a rede para grupos de hosts usando VLANs únicas ou múltiplas
  - Instale o IBM Cloud Private
- **Pós-instalação**
  - Mapeie equipes para um ou mais namespaces com base no requisito de agrupamento de recursos
  - Crie equipes e designe usuários e recursos; consulte Equipes para obter mais informações

Cenário 4: equipes e namespaces isolados, nós dedicados a namespaces, redes VLAN dedicadas, namespaces controlados por ingresso, solicitações de volume de persistência e volumes de persistência

Você deseja compartilhar o servidor da classe de armazenamento, como o NFS e o GlusterFS, nos namespaces com múltiplos locatários. Você cria volumes de persistência com base na classe de armazenamento. Para isolar o ambiente de armazenamento para namespaces dedicados, o volume de persistência com os rótulos deve ser criado. Quando os pods são implementados em um namespace isolado, as solicitações de volume de persistência podem ser configuradas para que se liguem ao volume de persistência específico usando seletores.




Tráfego Externo
Recurso do Ingress:
watch-namespace: apphr
tolerâncias: hostgroup-proxyapphr
Namespace apphr
hostgroup-proxyapphr
VLAN 10
VLAN 20
hostgroup-apphr
Controle de tráfego de ingresso L7
Recurso do Ingress:
watch-namespace: apphr
tolerâncias: hostgroup-proxyappsales
Namespace appsales
hostgroup-proxyappsales
VLAN 30
VLAN 40
hostgroup-appsales
Controle de tráfego de ingresso L7
Dispositivo L2/L3
PVC-hrwebpod
seletor: apphr
PV-hrwebpod
rótulo: apphr
PVC-hrserverpod
seletor: apphr
PV-hrserver
rótulo: apphr
PVC-saleswebpod
seletor: apphr
PVC-saleswebpod
rótulo: appsales
PVC-saleswebpod
seletor: apphr
PVC-saleswebpod
rótulo: appsales
nfs
glusterfs
hostpath
Armazenamento Físico

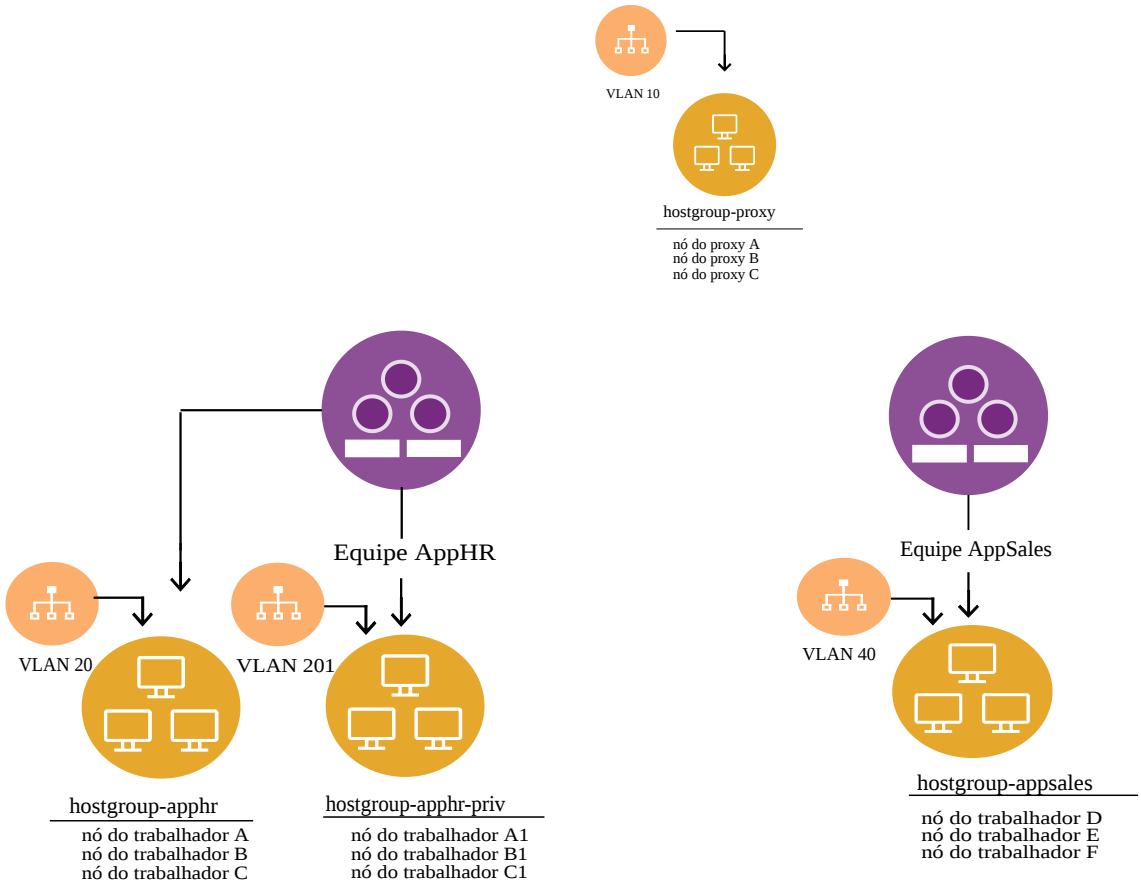
- **Planejamento**
  - Planeje os grupos de usuários e grupos de usuários em várias equipes
  - Planeje vários grupos de hosts customizados para nós de proxy e de trabalhador
  - Planeje os mapeamentos de namespace para grupos de hosts de proxy e de trabalhador
  - Planeje sub-redes de VLAN para cada grupo de hosts
  - Planeje volumes de persistência para cada namespace que for necessário.
- **Instalação**
  - Configure os grupos de hosts customizados para trabalhadores e nós de proxy nos arquivos host
  - Configure namespaces, que serão mapeados para cada grupo de hosts do trabalhador configurado em `<installation_directory>/cluster/config.yaml`
  - Configure namespaces, que serão mapeados para cada grupo de hosts de proxy configurado em `<installation_directory>/cluster/config.yaml` como um parâmetro `isolated_proxies`
  - Configure a rede para grupos de hosts usando VLANs únicas ou múltiplas
  - Instale o IBM Cloud Private
- **Pós-instalação**
  - Mapeie equipes para um ou mais namespaces com base no requisito de agrupamento de recursos
  - Crie equipes e designe usuários e recursos; consulte Equipes para obter mais informações.
  - Implemente os pods que possuem solicitações de volume de persistência. As solicitações de volume persistente têm seletores de rótulo que são mapeados para o volume de persistência apropriado criado para seu namespace. Para obter mais detalhes sobre como criar PVs e PVC, consulte Isolamento de armazenamento

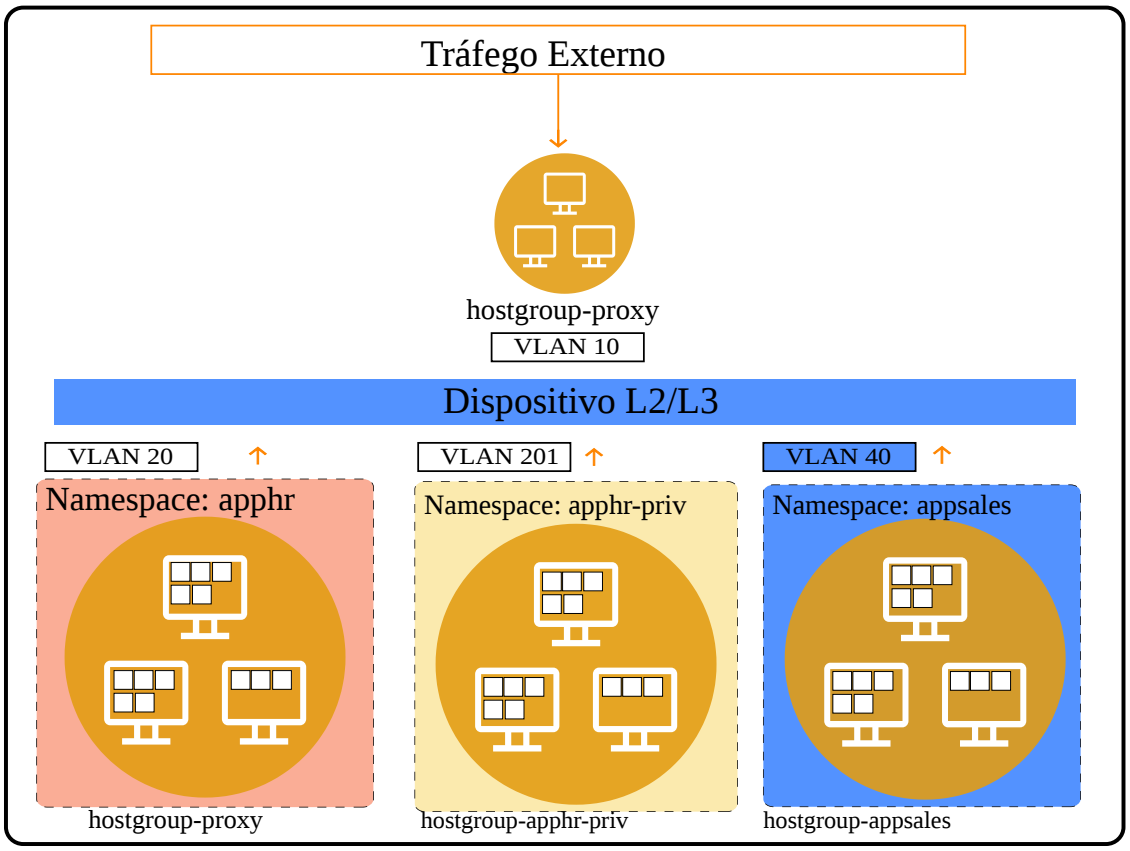
Cenário 5: equipes e namespaces isolados, nós dedicados para namespaces, redes isoladas e isolamento de pod

Este cenário é construído no Cenário 2, fornecendo o isolamento de usuário, da rede e de cálculo que inclui aumentos de estabilidade, de desempenho e de segurança para as imagens do contêiner. As imagens de contêiner, em execução em pods, são isoladas para selecionar grupos de nós usando namespaces. Portanto, os namespaces do aplicativo devem ser subdivididos em namespaces extras. O isolamento de ingresso do proxy não pode ser usado para a segregação de rede ao usar o isolamento de pod, já que vários namespaces são usados. Use VLANs ou políticas de rede para segregar as redes do nó do trabalhador umas das outras.

Isolamento de pod e de rede usando VLANs

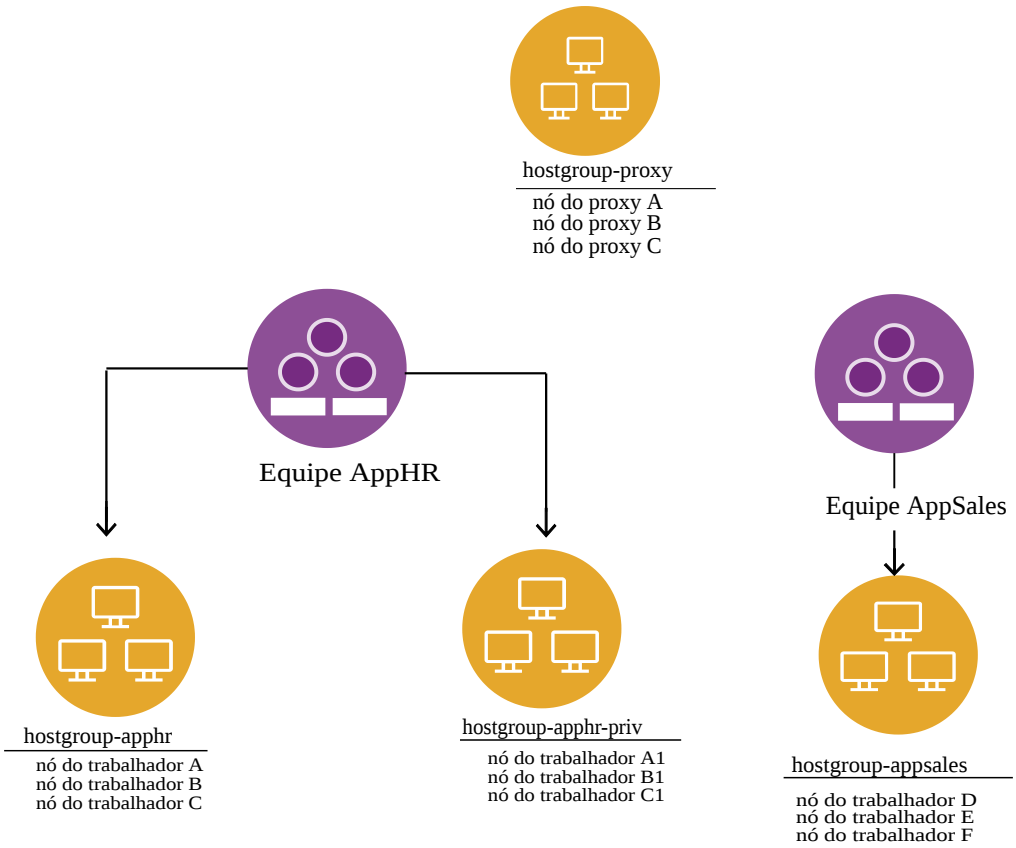
As VLANs de rede no nível de infraestrutura podem ser usadas para isolar o tráfego em um aplicativo do outro e como grupos de hosts. Para implementar o isolamento de pod e de rede usando VLANs, separe namespaces em duas ou mais categorias de privilégio de pod, que variam da menos privilegiada para a mais privilegiada.

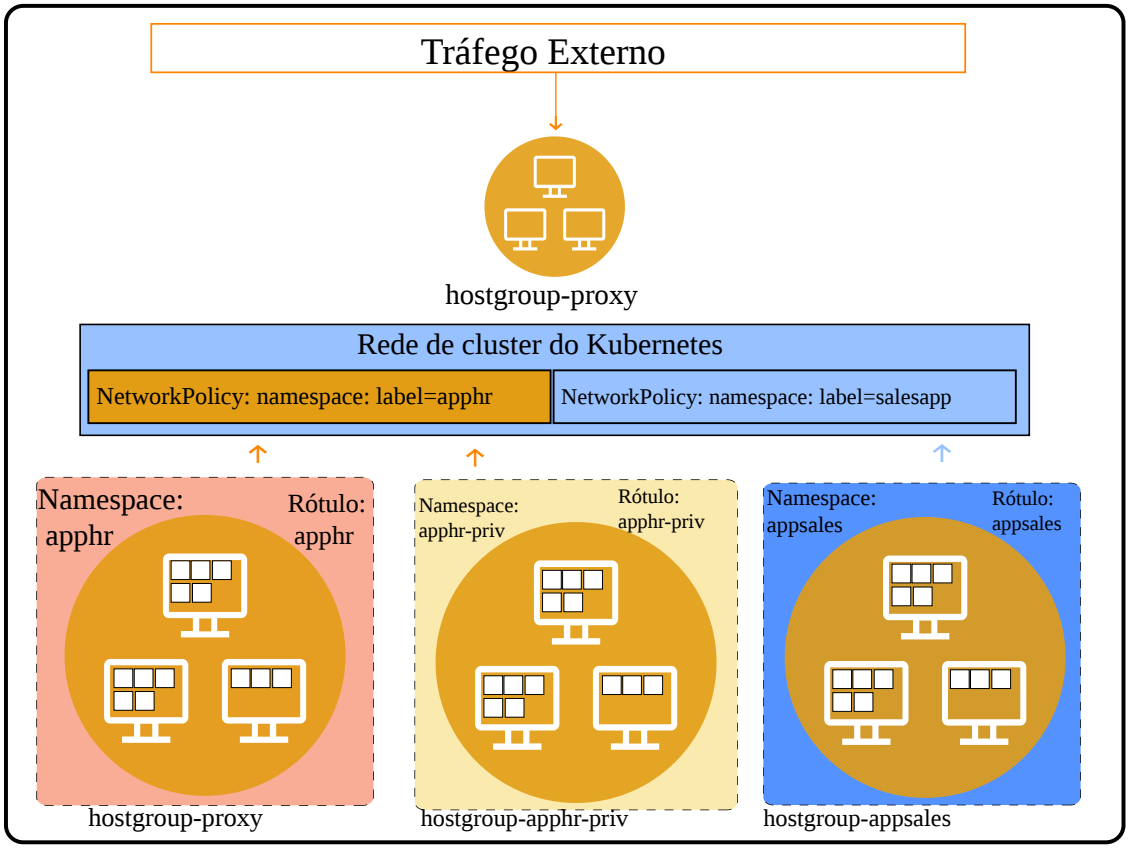




Isolamento de pod e de rede usando políticas de rede

As políticas de rede do Kubernetes podem ser usadas para o tráfego entre namespaces ou pods usando rótulos. Para implementar o isolamento de pod e de rede usando políticas de rede, separe namespaces em duas ou mais categorias de privilégio de pod que variam da menos privilegiada para a mais privilegiada e use um rótulo de namespace comum para permitir o tráfego de ingresso e de egresso entre namespaces no mesmo aplicativo.





- **Planejamento**
  - Os grupos de contexto de segurança da política de segurança de pod devem ser definidos

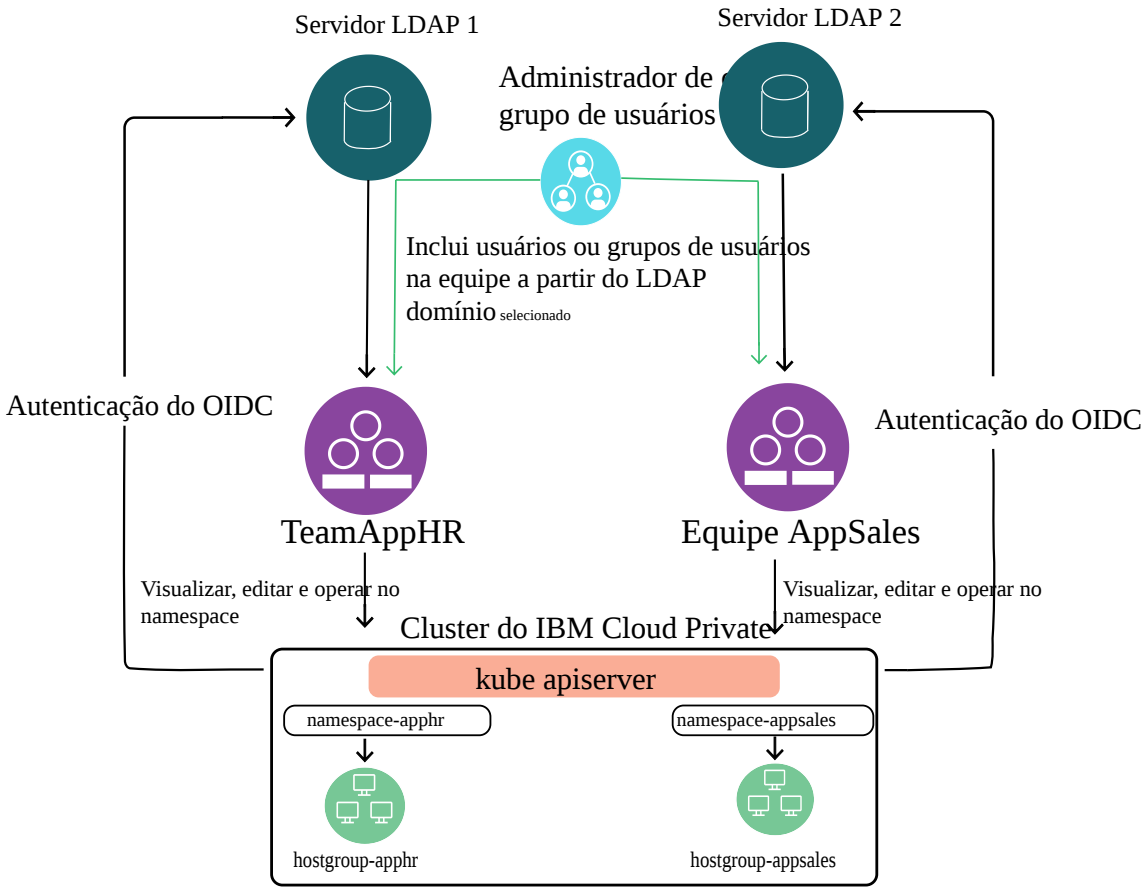
- Os namespaces devem ser identificados para aplicativos e para políticas de segurança de pod
- Os grupos de hosts devem ser mapeados para namespaces para isolar o cálculo de aplicativo
- As VLANs podem, opcionalmente, ser criadas para os grupos de hosts de firewall entre si

- **Instalação**
  - Configure nós para o IBM Cloud Private.
  - Configure a rede para nós usando VLANs únicas ou múltiplas
  - Configure e instale o IBM Cloud Private, especificando os grupos de hosts isolados por namespace

- **Pós-instalação**
  - Configure equipes, designando usuários aos namespaces
  - Configure políticas de segurança de pod para namespaces
  - Configure políticas de rede para namespaces

Cenário 6: múltiplos LDAPs configurados para um único cluster

Há múltiplos servidores LDAP, como o Open LDAP e o Active Directory, que são configurados para vários departamentos em uma organização. Os membros de cada departamento são isolados para seu próprio grupo de usuários, formando uma equipe. Cada equipe tem namespaces dedicados, alcançando ambientes de namespace isolados para os vários departamentos que acessam o mesmo cluster do IBM Cloud Private.



- **Planejamento**
  - Planeje os grupos de usuários entre vários servidores LDAP para equipes
- **Pôster de**
  - Configure múltiplos servidores de domínio LDAP para o IBM Cloud Private
  - Instale o IBM Cloud Private
- **Pós-instalação**
  - Configurar equipes de usuários a partir de múltiplos LDAPs
  - Mapeie equipes para um ou mais Namespaces com base no requisito de agrupamento de recursos.
  - Crie equipes e designe usuários e recursos; consulte Equipes para obter mais informações

As técnicas de isolamento em um ambiente em cluster do IBM Cloud Private são mais detalhadas nas seções a seguir:

- Isolamento baseado em RBAC
- Isolamento de LDAP
- Isolamento de Pod

Isolamento de armazenamento

O armazenamento envolve o pré-planejamento de recursos de armazenamento com base nos requisitos do aplicativo.

Quando os tipos de armazenamentos suportados são escolhidos, recursos do Kubernetes, como classes de armazenamento, volumes de persistência e solicitações de volume de persistência, podem ser criados, que podem ser usados para ambientes isolados.

Criando volumes persistentes

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Para visualizar o volume persistente na console de gerenciamento, no menu de navegação, selecione **Plataforma > Armazenamento**.

É possível criar volumes persistentes inserindo os valores de parâmetro na caixa de diálogo **Criar PersistentVolume** ou colando um arquivo YAML na página **Criar recurso**.

Criando volumes persistentes usando a caixa de diálogo Criar PersistentVolume

1. No menu de navegação, clique em **Plataforma > Armazenamento**.
2. Clique em **Criar PersistentVolume**.
3. Insira os detalhes de armazenamento. Para criar o armazenamento, os parâmetros a seguir são necessários:
    - Nome
    - Capacidade
    - Modo de acesso
    - Política de recuperação
    - Tipo de armazenamento: apenas NFS, GlusterFS, hostPath ou vSphere pode ser selecionado. Para usar outros tipos de armazenamento, use a página **Criar recurso**.
    - Parâmetros para armazenamento: esse parâmetro depende do tipo de armazenamento selecionado. Por exemplo, se você escolher o armazenamento NFS, será necessário especificar o servidor e o caminho para o armazenamento
4. Clique em **Criar**.

Criando PersistentVolumes usando a página Criar recurso

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML na caixa de diálogo.
3. Clique em **Criar**.

Após a conclusão da implementação, um novo PersistentVolume será exibido na lista. Revise o status do PersistentVolume. O volume deve estar no estado Disponível.

Criando PersistentVolumesClaim

Tipo de usuário ou nível de acesso necessários: Administrador de cluster ou administrador da equipe

É possível criar PersistentVolumeClaims para alocar armazenamento para seu aplicativo. Use essa tarefa para criar PersistentVolumeClaims para seu aplicativo. Antes de poder criar um PersistentVolumeClaims, um PersistentVolume deve estar disponível em seu cluster.

Um PersistentVolume disponível é ligado a um PersistentVolumeClaim e pode ser usado por um aplicativo. Cada PersistentVolume pode ser ligado apenas a um PersistentVolumeClaim. É possível criar PersistentVolumeClaims inserindo os valores de parâmetro na caixa de diálogo Criar PersistentVolumeClaim ou colando um arquivo YAML na janela "Criar recurso".

Para visualizar uma lista de PersistentVolumeClaim, clique em **Plataforma> Armazenamento > PersistentVolumeClaim** no menu de navegação.

Criando PersistentVolumeClaims usando a caixa de diálogo Criar PersistentVolumeClaim

1. No menu de navegação, clique em **Plataforma > Armazenamento**.
2. Selecione **PersistentVolumeClaim**.
3. Clique em Criar **PersistentVolumeClaim**.
4. Insira os detalhes de PersistentVolumeClaim na caixa de diálogo Criar PersistentVolumeClaim.

   Para criar um PersistentVolumeClaim, os parâmetros a seguir são necessários:

   - Nome - fornece um nome para o PersistentVolumeClaim.
   - Solicitações de armazenamento - quantidade de armazenamento necessária.
   - Modo de acesso - Para volumes que suportam vários modos de acesso, deve-se especificar o modo requerido.
5. Clique em **Criar**.

Criando PersistentVolumeClaims usando a página Criar recurso

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na página Criar recurso.
3. Clique em **Criar**.

Se um PersistentVolumeClaim for criado com êxito, um novo PersistentVolumeClaim será exibido na lista PersistentVolumeClaims. Revise o status. O PersistentVolumeClaim deve ter um status igual a `Bound`.

Controle de acesso baseado na função

O IBM® Cloud Private suporta várias funções. Sua função determina as ações que você pode fazer.

O Kubernetes oferece mecanismos de autorização de controle de acesso baseado em função (RBAC), que foram estendidos no IBM Cloud Private, em que os usuários da plataforma de cluster podem ser agrupados em equipes e têm namespaces dedicados às equipes. Com o IBM Cloud Private, é possível criar uma equipe e incluir usuários, grupos de usuários e recursos na equipe. Todos os usuários em uma equipe têm acesso aos recursos da equipe. Um usuário, grupo de usuários ou recurso pode ser designado a várias equipes.

O IBM Cloud Private tem um administrador de cluster que tem acesso a todo o cluster, ao passo que outros usuários podem ser classificados em várias funções, como Administrador, Editor, Operador, Auditor e Visualizador para vários namespaces aos quais eles têm acesso. Com base na função designada ao usuário/grupo de usuários, o nível de acesso para cada recurso lógico no cluster é definido.

Função e ações de administrador do cluster

O IBM Cloud Private suporta a função de administrador do cluster. O administrador de cluster tem acesso completo à plataforma IBM Cloud Private.

As ações a seguir podem ser concluídas pelo administrador de cluster:

- Conectar-se a um diretório LDAP
- Criar equipes, incluir usuários e designar a eles as funções do IAM
- Gerenciar cargas de trabalho, infraestrutura e aplicativos em todos os namespaces
- Criar namespaces
- Designar cotas
- Incluir políticas de segurança de pod
- Incluir um repositório Helm interno
- Excluir um repositório Helm interno
- Incluir gráficos Helm no repositório Helm interno
- Remover gráficos Helm do repositório Helm interno
- Sincronizar repositórios Helm internos e externos
- Gerenciar classes de armazenamento e volumes persistentes em todos os namespaces
- Incluir, remover e atualizar políticas de cumprimento de segurança de imagem

- Incluir, remover e atualizar IDs de serviço no cluster
- Registrar e incluir o Broker de Serviço para buscar ClusterServiceClasses e ClusterServicePlans
- Implementar o gráfico do Broker de Serviço para buscar ClusterServiceClasses e ClusterServicePlans

Para obter mais informações sobre como incluir políticas de segurança de pod, consulte Criando políticas de segurança de pod.

Funções e ações do IAM

É possível designar uma função do IAM a usuários ou grupos de usuários quando você os inclui em uma equipe. Dentro de uma equipe, cada usuário ou grupo de usuários pode ter apenas uma função. No entanto, um usuário pode ter várias funções em uma equipe quando você inclui um usuário individualmente e também como um membro do grupo de uma equipe. Nesse caso, o usuário pode agir com base na maior função designada ao usuário. Por exemplo, se você incluir o usuário como um administrador e designar uma função `Visualizador` ao grupo do usuário, o usuário poderá agir como um administrador para a equipe.

Um grupo de usuários ou usuário pode ser um membro de várias equipes e ter diferentes funções em cada equipe.

Uma função do IAM define as ações que um usuário pode executar nos recursos da equipe.

O IBM Cloud Private suporta essas funções do IAM:

Nota: somente o Administrador de cluster e o Administrador podem gerenciar equipes, usuários e funções. O Administrador não pode designar a função de Administrador de cluster para qualquer usuário ou grupo.

Função	Descrição	Ações
Visualizador	Tem acesso somente leitura.	As ações a seguir podem ser concluídas por um Visualizador: • Visualizar informações sobre os recursos

da equipe

- Visualize os dados de medição

O Visualizador não pode visualizar as páginas do console de gerenciamento a seguir:

- Painel
- Segredos
- Nós
- Identidade e Acesso
- Segurança de Recurso

| |Editor|Tem acesso de leitura e edição.|As ações a seguir podem ser concluídas por um Editor:

- Visualize os dados de medição

O Editor não pode visualizar as páginas do console de gerenciamento a seguir:

- Painel
- Nós
- Identidade e Acesso
- Segurança de Recurso

| |Auditor|Tem acesso de leitura.|As ações a seguir podem ser concluídas por um Auditor:

- Visualizar logs
- Visualize os dados de medição

O Auditor não pode ver o seguinte console de gerenciamento a seguir:

- Painel
- Nós
- Identidade e Acesso
- Segurança de Recurso

| |Operador|Tem acesso de leitura, edição e criação.|As ações a seguir podem ser concluídas por um Operador:

- Acessar painéis e dados de monitoramento
- Visualize os dados de medição
- APIs do serviço de monitoramento de acesso

O Operador não pode visualizar as páginas do console de gerenciamento a seguir:

- Painel
- Nós
- Identidade e Acesso
- Segurança de Pod

Nota: Os operadores não podem criar ou excluir uma política de imagem.| |Administrador|Tem acesso de inclusão, atualização, visualização e exclusão.|Você deve ser designado a uma equipe de diretório LDAP pelo administrador de cluster para concluir as ações a seguir:

- Criar equipes
- Designar recursos a outras equipes
  Nota: os Administradores podem designar recursos a equipes que o Administrador de Cluster designou a eles.
- Gerenciar usuários, grupos e funções para suas equipes
  Nota: Os administradores não podem designar a função de administrador do cluster para qualquer usuário ou grupo.
- Ler, atualizar e excluir recursos de uma equipe
- Acessar painéis e dados de monitoramento
- Acessar a página Medição para visualizar dados de medição e a página de chaves de API
- APIs do serviço de monitoramento de acesso

O Administrador não pode ver o seguinte console de gerenciamento a seguir:

- Painel
- Nós
- Segurança de Pod

Nota: Os administradores podem visualizar, criar ou excluir uma política de imagem.| |Administrador de Cluster|Tem acesso completo à plataforma IBM Cloud Private.|Consulte Função e ações do administrador de cluster|

Nota: os visualizadores e editores não podem visualizar logs em nenhuma das páginas da console de gerenciamento do IBM Cloud Private.

RBAC para recursos do Catalog e do Helm

Tabela 3. Ações permitidas do repositório Helm com base na função do IAM

Ação	Administrador	Operador	Editor	Auditor	Visualizador
Incluir um repositório Helm interno					
Sincronizar repositórios Helm internos e externos					
Excluir repositório Helm interno					
Incluir gráficos Helm no repositório Helm interno	X				
Remover gráficos Helm do repositório Helm interno	X				
Implementar gráficos Helm	X	*			
Retroceder liberações do Helm	X	X	X		
Atualizar liberações do Helm	X	X	X		
Excluir liberações do Helm	X				

X-Operação é suportada

* - A implementação e o upgrade de liberações do Helm não são suportados para gráficos que removem recursos usando ganchos ou tarefas. Para obter mais informações, consulte o arquivo leia-me ou a documentação do gráfico.

RBAC para recursos do Key Management Service (KMS)

Tabela 4. RBAC para KMS

Ação	Descrição	Administrador de cluster	Administrador	Editor	Visualizador
Crie	Gerar e importar chaves	X	X	X	

Ação	Descrição	Administrador de cluster	Administrador	Editor	Visualizador
Excluir	Excluir chaves	X	X		
Rotação de chave	Girar chaves	X	X		
Lista	Listar todas as chaves	X	X	X	
Ler	Ler material e metadados de chaves	X	X	X	
Quebrar	Usar CRK para criptografar DEK	X	X	X	X
Desagrupar	Usar CRK para decriptografar DEK	X	X	X	X

X-Operação é suportada

Para obter uma descrição detalhada de cada ação na tabela 4, consulte APIs do Key Management Service.

RBAC para recursos Kubernetes

A função do IAM que você designa a um usuário também define as ações que o usuário pode executar nos recursos do Kubernetes que são designados à equipe. Por exemplo, se user1 for um operador na team1 e team1 tiver o recurso namespace1, o user1 poderá visualizar e atualizar informações de namespace1. User1 também poderá criar recursos, por exemplo pods, em namespace1. Se você remover user1 de team1, removerá a ligação de função de user1 para os recursos em team1. Se user1 fizer parte de outra equipe, digamos team2, que possui o mesmo namespace, a ligação de função de user1 para o namespace em team2 não será afetada quando você remover o usuário de team1.

Tabela 5. Ações permitidas com base na função do IAM

Ação	Administrador	Operador	Editor	Visualizador
obter	X	X	X	X
listar	X	X	X	X
observar	X	X	X	X
atualizar	X	X	X	
correção	X	X	X	
criar	X			
excluir	X			
deletecollection	X			

Tabela 6. Permissões de recursos aceitas por função

Recurso	Administrador	Operador	Editor	Visualizador
clusterrolebindings.rbac.authorization.k8s.io	X			
clusterservicebrokers.servicecatalog.k8s.io (only view access)	X	X	X	X
clusterserviceclasses.servicecatalog.k8s.io (only view access)	X	X	X	X
clusterserviceplans.servicecatalog.k8s.io (only view access)	X	X	X	X
configmaps	X	X	X	X
cronjobs.batch	X	X	X	X
daemonsets.apps	X	X	X	X
daemonsets.extensions	X	X	X	X
deployments.apps	X	X	X	X
deployments.extensions	X	X	X	X
deployments.apps/rollback	X	X		
deployments.extensions/rollback	X	X		
deployments.apps/scale	X	X	X	
deployments.extensions/scale	X	X	X	X
terminais	X	X	X	X
eventos	X	X	X	X
horizontalpodautoscalers.autoscaling	X	X	X	X
images.icp.ibm.com	X	X	X	X
imagepolicies	X	X		
ingresses.extensions	X	X	X	X
jobs.batch	X	X	X	X
limitranges	X	X	X	X

Recurso	Administrador	Operador	Editor	Visualizador
localsubjectaccessreviews.authorization.k8s.io	X			
namespaces	X	X	X	X
namespaces/status	X	X	X	X
networkpolicies.extensions	X	X	X	X
networkpolicies.networking.k8s.io	X	X	X	X
persistentvolumeclaims	X	X	X	X
poddisruptionbudgets.policy	X			
pods	X	X	X	X
pods/attach	X	X	X	X
pods/exec	X	X	X	X
pods/log	X	X	X	X
pods/portforward	X	X	X	X
pods/proxy	X	X	X	
pods/status	X	X	X	
replicasets.apps	X	X	X	X
replicasets.extensions	X	X	X	X
replicasets.apps/scale	X	X	X	X
replicasets.extensions/scale	X	X	X	X
replicationcontrollers	X	X	X	X
replicationcontrollers/scale	X	X	X	X
replicationcontrollers.extensions/scale	X	X	X	X
replicationcontrollers/status	X	X	X	X
resourcequotas	X	X	X	X
resourcequotas/status	X	X	X	X
rolebindings.rbac.authorization.k8s.io	X			
roles.rbac.authorization.k8s.io	X			
scheduledjobs.batch	X			
segredos	X	X	X	
serviceaccounts	X	X	X	X
servicebindings.servicecatalog.k8s.io	X	X	X	X
servicebindings.servicecatalog.k8s.io/status	X	X	X	X
serviceinstances.servicecatalog.k8s.io	X	X	X	X
serviceinstances.servicecatalog.k8s.io/status	X	X	X	X
serviços	X	X	X	
services/proxy	X	X	X	X
statefulsets.apps	X	X	X	X

RBAC para recursos do IAM

Tabela 7. RBAC para recursos do IAM

Recurso do IAM	Ação	Administrador	Operador	Editor	Auditor	Visualizador
	Explorador de API do Identity Management	X	X	X	X	X
Certificado: /idmgmt/identity/api/v1/certificates						
	Criar certificado de usuário	X	X	X	X	X
	Certificado do usuário de	X	X	X	X	X
	Excluir certificado de usuário	X	X	X	X	X
Conta: /idmgmt/identity/api/v1/account						
	Ler IBM Cloud Private conta padrão	X	X	X	X	X
	Criar IBM Cloud Private conta padrão	X				

Recurso do IAM	Ação	Administrador	Operador	Editor	Auditor	Visualizador
	Atualizar a conta padrão do IBM Cloud Private	X				
	Excluir IBM Cloud Private conta padrão	X				
Diretório: /idmgmt/identity/api/v1/directory/ldap						
	Ler detalhes do diretório LDAP	X				
Usuário: /idmgmt/identity/api/v1/users						
	Criar detalhes do usuário	X				
	Ler detalhes do usuário	X	X	X	X	X
	Atualizar detalhes do usuário	X				
	Excluir detalhes do usuário	X				
Grupo de usuários: /idmgmt/identity/api/v1/usergroup						
	Criar detalhes do grupo de usuários	X				
	Ler detalhes do grupo de usuários	X	X	X	X	X
	Atualizar detalhes do grupo de usuários	X				
	Excluir detalhes do grupo de usuários	X				
Equipe: /idmgmt/identity/api/v1/teams						
	Criar detalhes da equipe	X				
	Detalhes da equipe de leitura	X	X	X	X	X
	Atualizar detalhes da equipe	X				
	Excluir detalhes da equipe	X				
Recurso: /idmgmt/identity/api/v1/resources						
	Criar detalhes do recurso	X				
	Ler detalhes do recurso	X				
	Atualizar detalhes do recurso	X				
	Excluir detalhes do recurso	X				
Preferências do usuário: /idmgmt/identity/api/v1/userpreferences						
	Criar preferências do usuário	X	X	X	X	X
	Ler preferências do usuário	X	X	X	X	X
	Atualizar preferências do usuário	X	X	X	X	X
Autenticação de Security Assertion Markup Language (SAML): /idmgmt/v1/saml						
	Obter status SAML	X				
	Atualizar ou reconfigurar a autenticação SAML	X				
	Criar ou configurar a autenticação SAML	X				
ID de serviço: /iam-token/serviceids						
	Crie um ID de serviço	X	X	X	X	X
	Listar detalhes do ID de Serviço	X	X	X	X	X
	Atualizar um ID de serviço	X	X	X	X	X
	Excluir um ID de serviço	X	X	X	X	X
Chave de API: /iam-token/apikeys						
	Criar uma chave de API	X	X	X	X	X
	Listar todas as chaves API	X	X	X	X	X
	Atualizar uma chave de API	X	X	X	X	X
	Excluir uma chave de API	X	X	X	X	X

Recurso do IAM	Ação	Administrador	Operador	Editor	Auditor	Visualizador
Política de serviço: /v1/scopes/{scope}/service_ids/{serviceId}/policies						
	Criar detalhes da política de serviço	X	X	X	X	X
	Detalhes da política de serviço de leitura	X	X	X	X	X
	Atualizar detalhes da política de serviço	X	X	X	X	X
	Excluir detalhes da política de serviço	X	X	X	X	X

Nota: um usuário pode criar políticas de ID de serviço com o mesmo nível de acesso que o usuário tem. O usuário não pode criar ou designar políticas com uma função superior a um ID de serviço.

Controle de acesso baseado em função para o IBM Multicloud Manager

Sua função determina a página de tópico que pode ser visualizada no IBM Multicloud Manager. Para configurar a função para um usuário para um documento de política, visualize o exemplo de política do IBM Multicloud Manager.

Visualize as funções e as páginas de tópico às quais cada função tem acesso na tabela a seguir:

Tabela 1. Ações de controle de acesso baseado na função para IBM Multicloud Manager

Página Tópico	Administrador de Cluster	Administrador	Operador	Editor	Visualizador
Visão Geral	x	x	x		
Clusters	x	x	x		
Políticas	x				
Aplicativos	x	x	x		
Liberações do Helm	x	x	x		
Pods	x	x	x		
Nós	x	x			
Armazenamento	x	x			
Topologia	x	x	x		
Gerenciamento de eventos	x	x	x		
Identidade e Acesso	x				
Ativar para o IBM Cloud Private	x				
Introdução	x	x	x		

Equipes

Uma equipe é necessária para designar funções aos usuários e gerenciar o acesso aos recursos.

- Criar equipes
- Incluir usuários em uma equipe
- Incluir grupos em uma equipe
- Incluindo RBAC em equipes para IDs de serviço
- Incluir recursos em uma equipe
- Remover usuários de uma equipe
- Remover grupos de uma equipe
- Remover recursos de uma equipe
- Remover equipes

Criar equipes

Criar uma equipe.

É possível criar uma equipe e incluir usuários, grupos de usuários e recursos na equipe. Todos os usuários em uma equipe têm acesso aos recursos da equipe. Um usuário, grupo de usuários ou recurso pode ser designado a várias equipes.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua essas etapas para criar uma equipe.

1. Efetue logon como um administrador de cluster.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**.
3. Clique em **Criar equipe**.
4. Na página Criar equipe, insira um nome para a equipe no campo **Nome da equipe**. Formato: de 1 a 50 caracteres alfanuméricos; espaço em branco é permitido; caracteres especiais que são permitidos: – _
5. (Opcional) É possível incluir usuários ou grupos de usuários na equipe.
    1. Selecione o domínio LDAP no qual a autenticação é armazenada.
    2. Procure usuários individuais ou grupos de usuários pelo nome. **Nota**: deve-se pressionar a tecla Enter para obter resultados do servidor LDAP.
    3. Selecione os usuários ou grupos de usuários que você deseja incluir.
    4. Selecione uma função para um usuário ou grupo de usuários. Todos os membros em um grupo recebem a mesma função que você selecionar. Para obter mais informações sobre as funções no IBM Cloud Private, consulte Controle de acesso baseado na função.
    5. Clique em **Criar**.
6. Designe usuários, grupos de usuários e recursos para a equipe. Consulte Incluir usuários em uma equipe, Incluir grupos em uma equipe e Incluir recursos em uma equipe para obter mais informações.

Incluir usuários em uma equipe

Inclua um usuário em uma equipe.

Sua conexão LDAP deve ser configurada antes de incluir usuários em uma equipe. Para obter mais informações sobre como configurar uma conexão LDAP, consulte Configurando a autenticação LDAP.

Deve-se criar uma equipe antes de poder incluir usuários em uma equipe. Para obter mais informações sobre como criar uma equipe, consulte Criar equipes.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua essas etapas para incluir usuários em uma equipe.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**.
3. Selecione o nome da equipe na lista de equipes.
4. Selecione a guia **Usuários**.
5. Selecione **Incluir usuário**. A caixa de diálogo "Incluir usuário" é exibida.
    1. Selecione o domínio LDAP no qual a autenticação é armazenada.
    2. Procure os usuários pelo nome. Ao inserir ou modificar o texto da procura, 50 resultados são exibidos. Insira o texto até que os usuários que deseja incluir sejam exibidos.
    3. Selecione os usuários que deseja incluir.
    4. Selecione uma função para cada usuário. Para obter mais informações sobre as funções no IBM Cloud Private, consulte Controle de acesso baseado na função.
    5. Clique em **Incluir**.

Os usuários são incluídos na equipe.

Incluir grupos em uma equipe

Inclua um grupo em uma equipe.

A conexão LDAP deve ser configurada antes de incluir grupos em uma equipe. Para obter mais informações sobre como configurar uma conexão LDAP, consulte Configurando a autenticação LDAP.

Deve-se criar uma equipe antes de poder incluir um grupo em uma equipe. Para obter mais informações sobre como criar uma equipe, consulte Criar equipes.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua essas etapas para incluir grupos de usuários em uma equipe.

1. Efetue logon como administrador.
2. No menu de navegação, selecione **Gerenciar** > **Equipes**.
3. Selecione o nome da equipe na lista de equipes.
4. Selecione a guia **Grupos**.
5. Selecione **Incluir grupo**. A caixa de diálogo **Incluir grupo** é exibida.
6. Selecione o Domínio LDAP no qual a autenticação de grupo é armazenada.
7. Procure os grupos pelo nome.
8. Selecione os grupos que você deseja designar à equipe.
9. Selecione a função para cada grupo. Para obter mais informações sobre as funções no IBM® Cloud Private, consulte Controle de acesso baseado na função.

   Nota: quando você designa uma função a um grupo, todos os usuários no grupo recebem a mesma função.
10. Selecione **Salvar**.

Os grupos são incluídos na equipe.

Incluindo o RBAC em suas equipes para IDs de serviço

É possível implementar permissões de função específicas nas suas equipes para IDs de serviço.

Antes de implementar o RBAC, conclua as etapas a seguir:

- Configure sua conexão LDAP. Para obter mais informações, consulte Configurando a conexão LDAP.
- Crie uma equipe e inclua usuários. Para obter mais detalhes para incluir usuários em sua equipe, consulte Incluir usuários em uma equipe.
- Ligue sua equipe a um ID de serviço. Para obter mais detalhes, consulte Ligando um ID de serviço a uma equipe

Funções específicas são incluídas no ID do serviço a partir da política de acesso. Para obter mais detalhes sobre como criar uma política de acesso, consulte Criar uma política de acesso para um ID de serviço.

Inclua um ID de serviço em sua equipe para que administradores e operadores possam gerenciar o ID de serviço.

1. No menu de navegação, clique em **Gerenciar** > **Identidade & Acesso** > **Equipes**.

   Nota: sua equipe deve ter acesso ao mesmo namespace que é ligado pelo seu ID de serviço.
2. Clique na guia **IDs de serviço**.
3. Clique em **Incluir IDs de serviço** e selecione o ID de serviço na lista.

   Nota: a lista é limpa se não houver IDs de serviço ligados a um namespace que esteja designado à equipe

Incluir recursos em uma equipe

Inclua um recurso em uma equipe.

É possível incluir um recurso por meio de seu cluster do IBM® Cloud Private. Os membros da equipe têm acesso apenas aos recursos que são incluídos na equipe.

Deve-se criar uma equipe antes de poder incluir um recurso nela. Para obter mais informações sobre como criar uma equipe, consulte Criar equipes.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Importante: certifique-se de que as conexões LDAP que estão associadas à equipe sejam incluídas como um recurso da equipe.

Conclua estas etapas para incluir recursos em uma equipe.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**.
3. Selecione o nome da equipe na lista de equipes.
4. Clique em **Recursos**.
5. Clique em **Incluir recurso**. Uma lista de recursos que estão disponíveis é exibida.
6. Selecione os recursos que você deseja incluir.
7. Clique em **Incluir recurso**.

Os recursos são incluídos na equipe.

Incluindo o repositório do Helm e o namespace em uma equipe

O repositório do Helm e o namespace devem ser configurados para que a equipe visualize as entradas do Catalog.

Sua conexão LDAP deve ser configurada antes de incluir o repositório do Helm e o namespace em uma equipe. Para obter mais informações sobre como configurar uma conexão LDAP, consulte Configurando a autenticação LDAP.

Deve-se criar uma equipe antes de poder incluir o repositório do Helm e o namespace em uma equipe. Para obter mais informações sobre como criar uma equipe, consulte Criar equipes.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

É possível incluir o repositório do Helm e o namespace concluindo as etapas a seguir:

1. Na navegação, selecione *Gerenciar* > *Identidade e acesso* > *Equipes*.
2. Selecione a equipe que você deseja atualizar.
3. Selecione **Recursos** e verifique se há um repositório do Helm e um namespace listado.
4. Se for necessário incluir os recursos, selecione **Gerenciar recursos**.
5. Marque a caixa de seleção para o repositório do Helm e o namespace que você deseja incluir na equipe.
6. Selecione **Salvar** para salvar suas mudanças.

As configurações são aplicadas à equipe.

Remover usuários de uma equipe

Remover um usuário de uma equipe.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua estas etapas para remover um usuário de uma equipe.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**.
3. Selecione o nome da equipe na lista de equipes.
4. Selecione a guia **Usuários**.
5. Para o usuário que você deseja remover, selecione **AÇÃO** > **Remover**. Uma caixa de diálogo de confirmação é exibida.
6. Clique em **Remover usuário**.

O usuário é removido da equipe.

Remover grupos de uma equipe

Remover um grupo de uma equipe.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua estas etapas para remover um grupo de uma equipe.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**.
3. Selecione o nome da equipe na lista de equipes.
4. Selecione a guia **Grupos**.

5. Para o grupo que você deseja remover, selecione **AÇÃO** > **Remover**. Uma caixa de diálogo de confirmação é exibida.
6. Clique em **Remover grupo**.

O grupo é removido da equipe.

Remover recursos de uma equipe

Remover um recurso de uma equipe.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua essas etapas para remover um recurso de uma equipe.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**.
3. Selecione o nome da equipe na lista de equipes.
4. Selecione a guia **Recursos**.
5. Para o recurso que você deseja remover, selecione **AÇÃO** > **Remover**. Uma caixa de diálogo de confirmação é exibida.
6. Clique em **Remover recurso**.

O recurso é removido da equipe.

Remover equipes

Remover uma equipe do cluster.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Conclua essas etapas para remover uma equipe do cluster.

1. Efetue logon como administrador.
2. No menu de navegação, clique em **Gerenciar** > **Equipes**. Uma lista de equipes é exibida.
3. Para a equipe que você deseja remover, selecione **AÇÃO** > **Remover**. Uma caixa de diálogo de confirmação é exibida.
4. Clique em **Remover equipe**.

A equipe é removida do cluster.

Namespaces

Os usuários são designados a unidades organizacionais chamadas namespaces.

Os namespaces também são conhecidos como locatários ou contas. No IBM® Cloud Private, os usuários são designados a equipes. É possível designar vários namespaces para uma equipe. Os usuários de uma equipe são membros de namespaces da equipe.

Um namespace do IBM Cloud Private corresponde a um único namespace no Kubernetes. Todas as implementações, pods e volumes que são criados em um único namespace pertencem ao mesmo namespace do Kubernetes.

Os namespaces a seguir são reservados pelo IBM Cloud Private:

Tabela 1. IBM Cloud Private namespaces

Namespace	Descrição	Permissão para acessar e implementar recursos
cert-manager	Reservado para o componente do gerenciador de certificados do IBM Cloud Private .	Administrador de cluster
Padrão	Disponível quando você instala o IBM Cloud Private e é usado como o namespace padrão para objetos que não especificam um namespace. Esse namespace não deve ser usado para nenhuma carga de trabalho de produção e não deve ser excluído.	Administrador de cluster

Namespace	Descrição	Permissão para acessar e implementar recursos
icp-system	Reservado para IBM Cloud Private. Este namespace não deve ser usado para cargas de trabalho de produção.	Administrador de cluster
istio-system	Reservado para serviços de plataforma Istio.	Administrador de cluster
kube-public	Reservado pelo Kubernetes e IIBM Cloud Private para armazenar informações de referência que estão disponíveis para qualquer usuário autenticado. Este namespace não deve ser usado para cargas de trabalho de produção.	Acesso livre Apenas o administrador de cluster pode implementar recursos
kube-system	Reservado para Kubernetes, IBM Cloud Private e outras cargas de trabalho confiáveis. Este namespace não deve ser usado para cargas de trabalho de produção.	Administrador de cluster
platforma	Reservado para IBM Cloud Private. Este namespace não deve ser usado para cargas de trabalho de produção.	Administrador de cluster
serviços	Reservado para o produto IBM Cloud Automation Manager.	Administrador de cluster

A página *Visão Geral do Namespace* na console de gerenciamento exibe a lista de políticas de segurança de pod associadas a cada namespace.

- Criando um namespace

Criando um Namespace

Os aplicativos devem ser criados ou implementados sob um namespace em um cluster do IBM® Cloud Private. Os namespaces são necessários para organizar usuários e seus aplicativos.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. Efetue logon como um administrador de cluster.
2. No menu de navegação, clique em **Gerenciar** > **Namespaces**.
3. Clique em **Criar Espaço de Nomes**.
4. Insira um nome para seu namespace. O nome do namespace deve atender aos requisitos do Kubernetes para namespaces, incluindo:
   - Ser exclusivo. Não é possível usar um nome de namespace duplicado.
   - Não exceder 63 caracteres de comprimento.
   - Conter apenas letras minúsculas, números e o símbolo traço (-).
5. Selecione a política de segurança de pod a ser associada ao seu namespace. A política de segurança de pod fornece aspectos de segurança para pods que são criados por aplicativos que são instalados no namespace. O campo de política de segurança de pod aplicada padrão exibe o nome da política que é aplicado no cluster para todos os namespaces que são criados.

   Atenção: múltiplas políticas de segurança de pod podem ser associadas a um namespace ou ServiceAccount. Os pods podem ser resolvidos para qualquer uma das políticas definidas que sejam compatíveis, o que pode resultar em uma política menos restritiva. Consulte Ordem de política.
6. Clique em **Criar**.

Após a implementação ser concluída, um novo namespace é exibido na página de namespaces.

Nota: assegure-se de rolar o menu `All namespaces` para visualizar todos os namespaces.

Incluindo um rótulo

É possível usar rótulos para criar políticas de rede que são baseadas em um seletor de namespace. Não é possível incluir rótulos para um namespace na console de gerenciamento do IBM Cloud Private. Para incluir rótulos de namespace, use a CLI do Kubernetes.

1. Instale a interface da linha de comandos `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Visualize uma lista de todos os namespaces.

   ```
   kubectl get namespaces
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME          STATUS    AGE
   default       Active    6h
   dev           Active    2h
   kube-system   Active    6h
   qa            Active    2h
   ```

3. Incluir rótulo.

   ```
   kubectl label namespaces dev team=dev
   ```

Removendo um Espaço de Nomes

Remover namespaces que não são mais usados.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. Efetue logon como um administrador de cluster.
2. No menu de navegação, clique em **Gerenciar** > **Namespaces**.
3. Para o namespace que você deseja remover, selecione **ACTION** > **Remover**. Uma caixa de diálogo de confirmação é exibida.
4. Clique em **Remover namespace**.

Isolamento do LDAP

Há uma necessidade crescente para que os usuários do IBM Cloud Private possam ser capazes de se autenticar em múltiplos LDAPs. Às vezes, as grandes organizações podem ter um controlador de domínio LDAP para diferentes regiões ou subsidiárias globais.

Os usuários podem ter uma combinação de tipos de diretório, como AD, Tivoli, OpenLDAP, etc.

Os usuários podem definir múltiplos diretórios na configuração do LDAP no IBM Cloud Private. O IBM Cloud Private usa o Websphere Liberty Server OpenID Connect como um serviço de autenticação que executa a administração e a autenticação com relação ao diretório apropriado.

Múltiplos registros do LDAP

Como administrador de cluster, é possível configurar múltiplos domínios LDAP incluindo múltiplas entradas de diretório na configuração do LDAP em `server.xml`.

Em um ambiente configurado com múltiplos domínios, uma nova administração do usuário na plataforma do IBM Cloud Private cumpre uma seleção de domínios apropriados e o usuário é incluído na Equipe.

O perfil do usuário e o nome do domínio são mantidos pelo IBM Cloud Private, que é usado ainda mais para gerenciamento de usuário. A capacidade de escolher o domínio antes de selecionar usuários para uma equipe permite que o administrador isole as equipes com um domínio específico.

Nota: as credenciais do usuário são passadas pelo IBM Cloud Private para o servidor Websphere Liberty OIDC, que resolve o domínio do usuário e autentica o usuário com um domínio correspondente.

Para obter mais informações, consulte Equipes.

Isolamento de pod

Políticas de segurança de pod podem ser usadas para aplicar a segurança da imagem do contêiner para os pods em seu cluster. Uma política de segurança de pod é um recurso de nível de cluster que controla os aspectos sensíveis à segurança da especificação do pod e o conjunto de condições que devem ser atendidas para que um pod seja admitido no cluster.

As políticas de segurança de pod são usadas para configurar o controle de nível do cluster sobre o que um pod pode fazer ou o que ele pode acessar.

As seguintes políticas de segurança de pod estão disponíveis no IBM® Cloud Private:

- ibm-restricted-psp
- ibm-anyuid-psp
- ibm-anyuid-hostpath-psp
- ibm-anyuid-hostaccess-psp
- ibm-privileged-psp

Com essa nova política de segurança, o administrador de cluster pode designar as permissões necessárias para um namespace e, em seguida, autorizar o namespace a usar essa política de segurança de pod. Os usuários nesse namespace designado são capazes de criar pods com permissões elevadas. Por exemplo, um usuário no namespace `Dev` pode criar pods privilegiados e pode usar a rede do host.

Para obter mais informações sobre políticas, consulte Políticas de Segurança de Pod.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

- Segurança de pod
- Usando namespaces com políticas de segurança de pod
- Planejando-se para pods isolados
- Ativando o isolamento de pod
- Gerenciando ligações de namespace para políticas de segurança de pod
- Implementando gráficos de Cloud Paks e Helm

Segurança do pod

Saiba como proteger pods e contêineres que estão em execução em seu cluster.

O Kubernetes é uma plataforma de orquestração de contêineres. As imagens do contêiner podem vir de várias origens e ter requisitos do sistema operacional diferentes. Alguns contêineres são gravados para serem autocontidos, de modo que eles não precisam de nenhum recurso do sistema operacional do host. Outros contêineres requerem acesso ao sistema operacional do host, como a rede do host ou o sistema de arquivos.

É importante identificar quais recursos são necessários para que os contêineres em execução possam ser localizados juntamente de maneira eficiente ou isolados uns dos outros.

Políticas de segurança de pod podem ser usadas para aplicar a segurança da imagem do contêiner para os pods em seu cluster. Uma política de segurança de pod é um recurso de nível de cluster que controla os aspectos sensíveis à segurança da especificação do pod e o conjunto de condições que devem ser atendidas para que um pod seja admitido no cluster. A política de segurança de pod é aplicada ao namespace, criando um *ClusterRoleBinding* ou *RoleBinding* com a respectiva política de segurança de pod `ClusterRole` para todas as ServiceAccounts no namespace. As políticas de segurança de pod permitem que os administradores de cluster criem políticas de isolamento de pod e designe-as a namespaces e nós do trabalhador. Para obter informações adicionais sobre pods isolados, consulte Isolamento do pod. Para obter informações adicionais sobre a política de segurança de pod, consulte *Políticas de segurança de pod* na página Conceitos do Kubernetes.

Importante: Várias políticas de segurança de pod podem ser associadas a um namespace ou ServiceAccount. Os pods podem ser resolvidos para qualquer uma das políticas definidas compatíveis. Para obter informações adicionais sobre a ordem de política, consulte *Ordem de política* na página Conceitos do Kubernetes.

Um Pod do Kubernetes é um conjunto de 1 ou mais contêineres localizados juntamente. O Controlador de Admissão de Pod evita a criação de um pod quando a política de segurança do pod não permite o recurso privilegiado. O Controlador de Admissão de Pod também pode configurar valores padrão no pod e no contêiner, evitando ou permitindo o acesso no tempo de execução a recursos privilegiados.

Todos os IBM Certified Containers fornecem requisitos de segurança de contêiner detalhados. Para obter mais informações, consulte IBM Certified ContainerDefinições de política de segurança.

Consulte Isolamento no IBM Cloud Private para saber mais sobre o isolamento de pod.

Políticas de segurança de pod predefinidas

O IBM Cloud Private fornece políticas predefinidas que podem ser aplicadas em seu pod associando-as a um namespace durante a criação de namespace. Essas políticas de segurança de pod predefinidas se aplicam à maioria dos gráficos de conteúdo do IBM. A lista a seguir mostra os tipos e as descrições que variam desde as mais restritivas até as menos restritivas:

- `ibm-restricted-psp`: esta política requer que os pods sejam executados com um ID do usuário não raiz e evita que os pods acessem o host.
- `ibm-anyuid-psp`: esta política permite que os pods sejam executados com qualquer ID do usuário e ID do grupo, mas evita o acesso ao host.
- `ibm-anyuid-hostpath-psp`: esta política permite que os pods sejam executados com qualquer ID do usuário, ID do grupo e qualquer volume, incluindo o caminho do host. **Atenção:** esta política permite volumes hostPath. Assegure-se de que esse seja o nível de acesso que você deseja fornecer.
- `ibm-anyuid-hostaccess-psp`: esta política permite que os pods sejam executados com qualquer ID do usuário, ID do grupo, qualquer volume e concede acesso total ao host. **Atenção:** esta política permite acesso total ao host e à rede. Assegure-se de que esse seja o nível de acesso que você deseja fornecer.
- `ibm-privileged-psp`: esta política concede acesso a todos os recursos de host privilegiados e permite que um pod seja executado com qualquer ID do usuário, ID do grupo e qualquer volume. **Atenção:** esta política é a menos restritiva e deve ser usada apenas para administração de cluster. Use-a com cuidado.

Para obter mais detalhes sobre as definições de política, consulte as Definições de política de segurança de pod do IBM Certified Container.

Se você instalar o IBM Cloud Private versão 3.2.0 ou posterior como uma nova instalação, a configuração de política de segurança de pod padrão será a política `ibm-restricted-psp`, que é aplicada a todos os namespaces recém-criados e existentes. Se você fizer upgrade para a versão 3.2.0 ou posterior a partir de uma versão anterior, a segurança do pod padrão será a política `ibm-anyuid-hostpath-psp`, que é irrestrita e aplicada por padrão a todos os namespaces recém-criados e existentes. As configurações atualizadas são irrestritas para evitar quaisquer interrupções com problemas de acesso nos aplicativos e pods já em execução em seu cluster.

Também é possível usar os comandos da CLI do IBM® Cloud Private para visualizar as configurações atuais do cluster e para alternar a política de segurança de pod entre os modos restrito e irrestrito. Consulte Comandos da CLI cm (cm do) IBM Cloud Private para obter mais informações sobre como usar esses comandos.

É possível visualizar as informações de configurações de comando da CLI PodSecurityPolicy na console de gerenciamento do IBM Cloud Private, selecionando o ícone de configurações na página Segurança de pod.

A página *visão geral de namespace* mostra as políticas de segurança de pod para cada namespace. Consulte Namespaces para obter mais informações sobre namespaces.

Ao criar um namespace, agora é possível associar a política de segurança de pod ao namespace, que pode ser usado para implementar conteúdo ou um gráfico. Consulte Criando um namespace para obter mais detalhes sobre como associar uma política de segurança de pod a um namespace.

Os requisitos da política de segurança de pod são exibidos na página de *configuração de gráfico* para ajudar a facilitar a seleção e implementação do namespace. Consulte Implementando gráficos de Helm no Catalog.

Políticas de segurança de pod customizadas

Também é possível criar suas próprias políticas de segurança de pod.

Consulte os tópicos a seguir para gerenciar sua política de segurança de pod:

- Criando políticas de segurança de pod
- Comandos da CLI cm (cm) do IBM Cloud Private

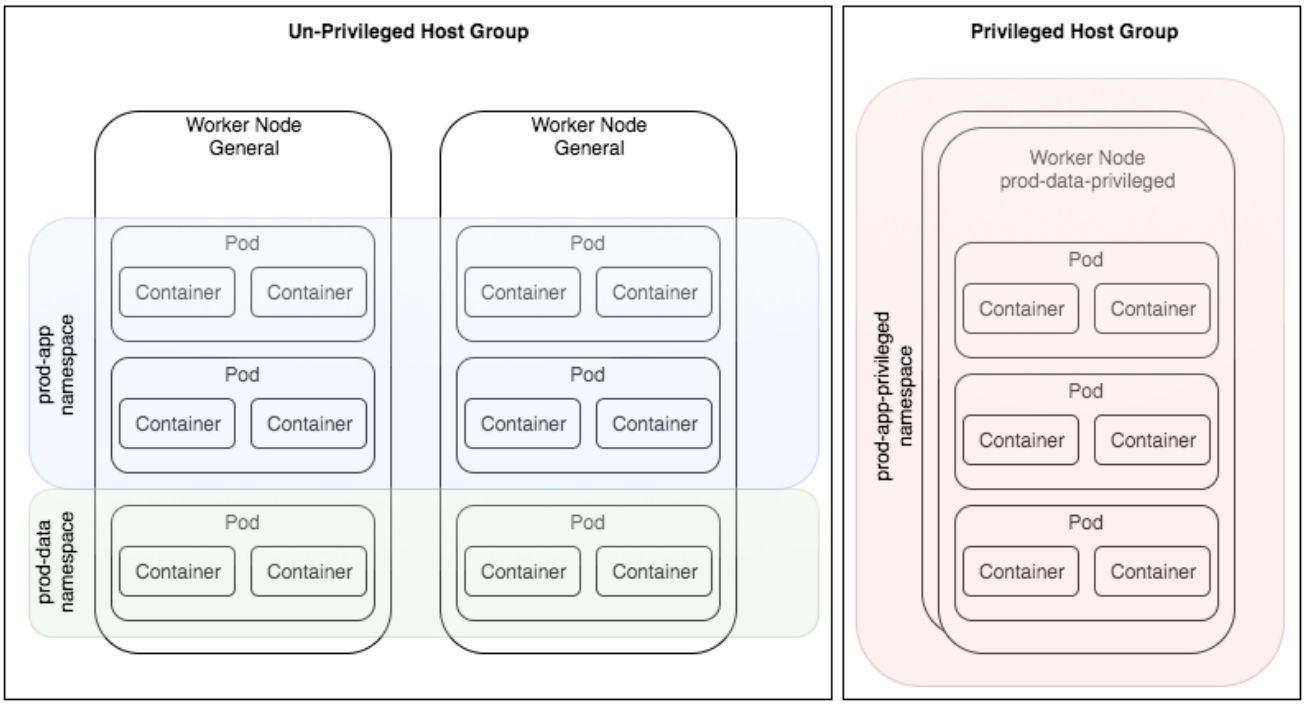
Usando namespaces com políticas de segurança de pod

Os namespaces são usados para isolar grupos de recursos uns dos outros. A segurança de um namespace é determinada pelas `ClusterRoleBindings` e `RoleBindings` dos usuários e contas de serviço.

Como as ligações são criadas para contas de serviço em funções e funções de cluster, qualquer pod no namespace pode utilizar qualquer uma das contas de serviço sem nenhuma verificação de elevação de permissão adicional.

Para simplificar esse pensamento, a estratégia é mapear uma política de segurança de pod única para todas as contas de serviço em um namespace, ou usar um pequeno número de políticas de segurança de pod que possuem uma estratégia de segurança semelhante.

O diagrama a seguir mostra os nós do trabalhador em um grupo de hosts não privilegiados e nós do trabalhador em um grupo de hosts privilegiados.



Consulte Isolamento de pod para obter mais informações.

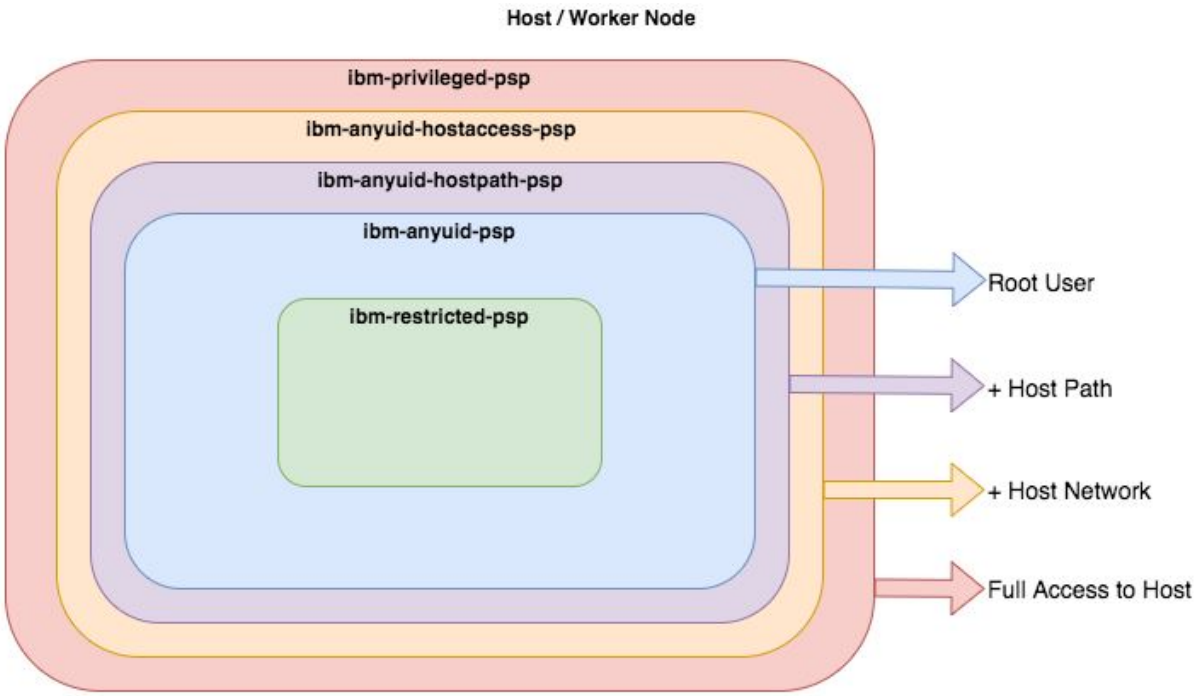
Planejando pods isolados

Configure seu cluster do IBM® Cloud Private com pods isolados.

Há várias maneiras pelas quais um contêiner pode acessar o sistema operacional do host. O IBM Cloud forneceu cinco políticas de segurança de pod pré-instaladas e funções do cluster, da menos privilegiada (`ibm-restricted-psp`) à mais privilegiada (`ibm-privileged-psp`). Essa lista não é abrangente e os administradores de cluster estão incentivados a criar suas próprias políticas para suas cargas de trabalho.

Consulte Segurança do Pod para obter detalhes sobre a política de segurança do pod e as funções de cluster incluídas.

O diagrama a seguir mostra as camadas de política de segurança de pod em um nó do host/trabalhador.



Aprenda sobre outras práticas para o isolamento de pod.

- Executar pods privilegiados separadamente de pods não privilegiados
- Ligar políticas de segurança de pod a namespaces em vez de contas de serviço
- Especificar uma política de segurança de pod por namespace
- Evitar criar pods diretamente
- Usar política de segurança de pod customizada para implementar os menos privilegiados

Para saber mais sobre isolamentos de pods, consulte Isolamento de pod.

Executando pods privilegiados separadamente de pods não privilegiados

Os pods não privilegiados são aqueles que podem ser executados com a política de segurança de pod `ibm-restricted-psp`. Esses contêineres não requerem nenhum privilégio elevado e são menos propensos a afetar outros contêineres no mesmo nó.

Você precisa separar quaisquer pods que requeiram privilégios especiais, principalmente se a carga de trabalho não for plenamente confiável ou não estiver documentada. Consulte Planejamento para pods isolados para obter outras práticas para isolar seus pods.

Ligando políticas de segurança de pod a namespaces em vez de a contas de serviço

O Kubernetes não verifica permissões elevadas de controle de acesso baseado em função (RBAC) quando um administrador de cluster designa uma conta de serviço a um pod. Ele apenas verifica se há permissões elevadas quando uma `RoleBinding` ou `ClusterRoleBinding` é criada.

Se um administrador de cluster criar várias contas de serviço em um namespace com vários níveis de privilégios, o namespace será apenas tão seguro quanto à conta do serviço com a maioria dos privilégios.

É mais fácil e seguro para um administrador de cluster examinar as configurações de segurança de um namespace quando uma política de segurança de pod é ligada a todas as contas de serviço em vez de a contas individuais. Para descobrir se um namespace está ligado a uma política de segurança de pod específica, consulte Verificando ligações existentes.

Para saber mais sobre outras práticas para o isolamento de pod, consulte Planejando pods isolados.

Especificando uma política de segurança de pod por namespace

As políticas de segurança de pod são designadas aos pods pelo Controlador de Admissão de Pod com base no usuário que cria o pod.

Para os pods criados por controladores, como `Implementações`, um usuário do namespace é a Conta de Serviço. Se mais de uma política corresponder ao contexto de segurança declarado do pod, qualquer uma das políticas poderá corresponder. Para obter mais informações, consulte a seção *Ordem de política* da Documentação do Kubernetes.

Consulte Planejamento para isolamento de pods para saber mais sobre outras práticas de isolamento de pod.

Evite criar pods diretamente

Os pods podem ser criados diretamente, com as credenciais do usuário. Isso pode contornar a política de segurança de pod que está ligada às contas de serviço no namespace de destino.

Um administrador de cluster pode executar um pod privilegiado em um namespace que esteja configurado como um namespace não privilegiado.

Consulte Planejamento para isolamento de pods para saber mais sobre outras práticas de isolamento de pod.

Usando políticas de segurança de pod customizadas para implementar a menos privilegiada

Todas as políticas de segurança de pod pré-instaladas que estão disponíveis no IBM® Cloud Private (exceto `ibm-restricted-psp`) ativam o acesso raiz.

Um IBM Certified Containers inclui uma definição de política de segurança de pod menos privilegiada e uma referência a uma política de segurança de pod pré-instalada. É mais seguro usar a política de segurança de pod menos privilegiada para implementações de produção. Para obter mais informações, consulte IBM Certified Container Pod Security Definitions] (https://github.com/IBM/cloud-pak/blob/master/spec/security/psp/README.md).

Consulte Planejando os pods de isolamento para aprender sobre outras práticas para o isolamento de pod.

Ativando o isolamento de pod

O isolamento de Pod é ativado automaticamente para o IBM® Cloud Private.

Se estiver fazendo upgrade da versão 3.1.2 para a 3.2.0, devem ser executadas etapas adicionais para ativar uma política de segurança de pod padrão global, restrita para todos os usuários e contas de serviço.

Para ativar a configuração de política de segurança de pod restrita com a linha de comandos do IBM Cloud Private, conclua as seguintes etapas:

1. Efetue login no IBM Cloud Private com o `cloudctl` como um administrador de cluster.
2. Verifique se o valor para o padrão de política de segurança de pod está configurado como `irrestrito`. Execute o comando a seguir:

   ```
   cloudctl cm psp-default-get
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   Default PSP:        unrestricted
   ```

3. Ative a política de segurança de pod padrão `restrita`. Execute o comando a seguir:

   ```
   cloudctl cm psp-default-set restricted
   OK
   ```

   As mudanças a seguir são feitas em seu cluster:

- Todos os recursos PodSecurityPolicy e ClusterRoleBinding são criados ou reparados.
- A ClusterRoleBinding não restrita `ibm-anyuid-hostpath-psp-users` é removida.
- A ClusterRoleBinding restrita `ibm-restricted-psp-users` é criada. Isso mapeia a PodSecurityPolicy `ibm-restricted-psp` para todos os usuários no cluster, incluindo todas as contas de serviço.
- Para qualquer namespace que não tenha uma RoleBinding ou uma ClusterRoleBinding explícita, uma PodSecurityPolicy é ligada ao `ibm-anyuid-hostpath-clusterrole` usando uma RoleBinding. Isso preserva a compatibilidade para quaisquer cargas de trabalho.

Para obter informações adicionais sobre pods isolados, consulte Isolamento de pod.

Gerenciando ligações de namespace para políticas de segurança de pod

As políticas de segurança de pod são mapeadas para namespaces com recursos RoleBinding e ClusterRoleBinding. Elas ligam contas de serviço individuais ou o grupo de namespaces inteiro de contas de serviço a uma política de segurança de pod com um ClusterRole.

O IBM Cloud Private usa o padrão a seguir: **RoleBinding > ClusterRole > Política de Segurança de Pod**.

Aprenda a gerenciar ligações de namespace para políticas de segurança de pod.

- Verificando ligações existentes
- Incluindo uma ligação de política de segurança de pod em um namespace
- Removendo uma ligação de política de segurança de pod de um namespace

Para obter informações adicionais sobre pods isolados, consulte Isolamento de pod.

Verificando ligações existentes

É possível validar se uma política de segurança de pod está ligada ao namespace inteiro ou ligada a uma conta de serviço.

Verificando ligações existentes para um namespace inteiro

Conclua as seguintes etapas para verificar se uma política de segurança de pod está ligada a um namespace inteiro:

1. Efetue login como um administrador de cluster.
2. Valide se uma política de segurança de pod está ligada a um namespace. Por exemplo, verifique se o namespace `kube-system` inteiro está ligado ao PodSecurityPolicy `ibm-privileged-psp`. Execute o comando a seguir:

   ```
   kubectl -n kube-system auth can-i use podsecuritypolicies/ibm-privileged-psp --as system:serviceaccount:kube-system:fake-serviceaccount
   ```

   Se uma política de segurança de pod não estiver ligada a um namespace, sua saída pode ser semelhante à seguinte mensagem:

   ```
   no
   ```

Verificando ligações existentes para uma conta de serviço

Conclua as seguintes etapas para verificar se uma política de segurança de pod está ligada a uma conta de serviço:

1. Efetue login no administrador do cluster.
2. Valide se uma conta de serviço está ligada a uma política de segurança de pod. Por exemplo, verifique se a conta de serviço `padrão` inteira está ligada ao PodSecurityPolicy `ibm-privileged-psp`. Execute o comando a seguir:

   ```
   kubectl -n kube-system auth can-i use podsecuritypolicies/ibm-privileged-psp --as system:serviceaccount:kube-system:default
   ```

   Se uma política de segurança de pod estiver ligada a um namespace, sua saída poderá ser semelhante à seguinte mensagem:

   ```
   sim
   ```

Consulte Gerenciando ligações de namespace para as políticas de segurança de pod para obter mais detalhes.

Incluindo uma ligação de política de segurança de pod em um namespace

É possível incluir uma ligação de política de segurança de pod em um namespace.

Conclua as etapas a seguir para incluir uma política de segurança de pod em um namespace:

1. Efetue login no administrador do cluster.
2. Inclua uma ligação de política de segurança de pod em um namespace. Por exemplo, crie uma Ligação de Função no namespace `appsales` para a Política de Segurança de Pod `ibm-anyuid-psp`. Execute o comando a seguir:

   ''' kubectl -n appsales create rolebinding ibm-anyuid-clusterrole-rolebinding --clusterrole=ibm-anyuid-clusterrole --group=system:serviceaccounts:appsales '''

Consulte Gerenciando ligações de namespace para as políticas de segurança de pod para obter mais detalhes.

Removendo uma ligação de política de segurança de pod de um namespace

É possível remover uma ligação de política de segurança de pod de um namespace.

Para remover uma ligação de política de segurança de pod de um namespace, conclua as etapas a seguir:

1. Efetue login como o administrador do cluster ou administrador da equipe de namespace.
2. Remova uma Ligação de Função de um namespace. Por exemplo, remova uma Ligação de Função denominada `ibm-anyuid-clusterrole-rolebindin` a partir do namespace `appsales`. Execute o comando a seguir:

   ```
   kubectl -n appsales delete rolebinding ibm-anyuid-clusterrole-rolebinding
   ```

Consulte Gerenciando ligações de namespace para as políticas de segurança de pod para obter mais detalhes.

Implementando gráficos do {{site.data.keyword.container_soft}} e do Helm

Um administrador de cluster deve configurar a implementação de gráficos do {{site.data.keyword.container_soft}} e do Helm em pods isolados para que o gráfico possa ser implementado por um operador ou administrador da equipe.

Aprenda sobre as tarefas do administrador de cluster e as tarefas do operador da equipe como a implementação e a configuração de gráficos do {{site.data.keyword.container_soft}} e do Helm em pods isolados.

- Tarefas do administrador do cluster
- Tarefas do operador da equipe

Tarefas do administrador do cluster

O administrador de cluster deve preparar o cluster antes de permitir que um administrador ou operador de equipe instale um gráfico do Helm ou IBM Certified Container.

O administrador de cluster deve criar uma política de segurança de pod customizada e namespaces com a ligação de política de segurança de pod.

A lista a seguir descreve as decisões e tarefas que um administrador de cluster deve executar ao planejar e implementar pods isolados com políticas de segurança de pod.

1. Verifique se o isolamento de pod está ativado.
2. Para cada gráfico instalado, conclua as etapas a seguir:
   1. Examine os requisitos de política de segurança de pod do gráfico.

2. Decida se você deve usar uma política de segurança de pod predefinida ou criar uma política customizada.
    1. Crie uma política de segurança de pod customizada e a função de cluster, se desejar.
3. Crie um namespace para o gráfico instalado.
4. Ligue a política de segurança de pod a um namespace com uma ligação de função.
5. Designe o namespace e o gráfico a uma equipe.

3. Comunique as informações a seguir com um administrador ou operador da equipe, descrevendo:
    - Os gráficos a serem instalados.
    - Os namespaces nos quais fazer a instalação.

Saiba planejar e implementar pods isolados em políticas de segurança de pod.

- Criando uma política de segurança de pod customizada
- Criando namespaces com ligação de política de segurança de pod

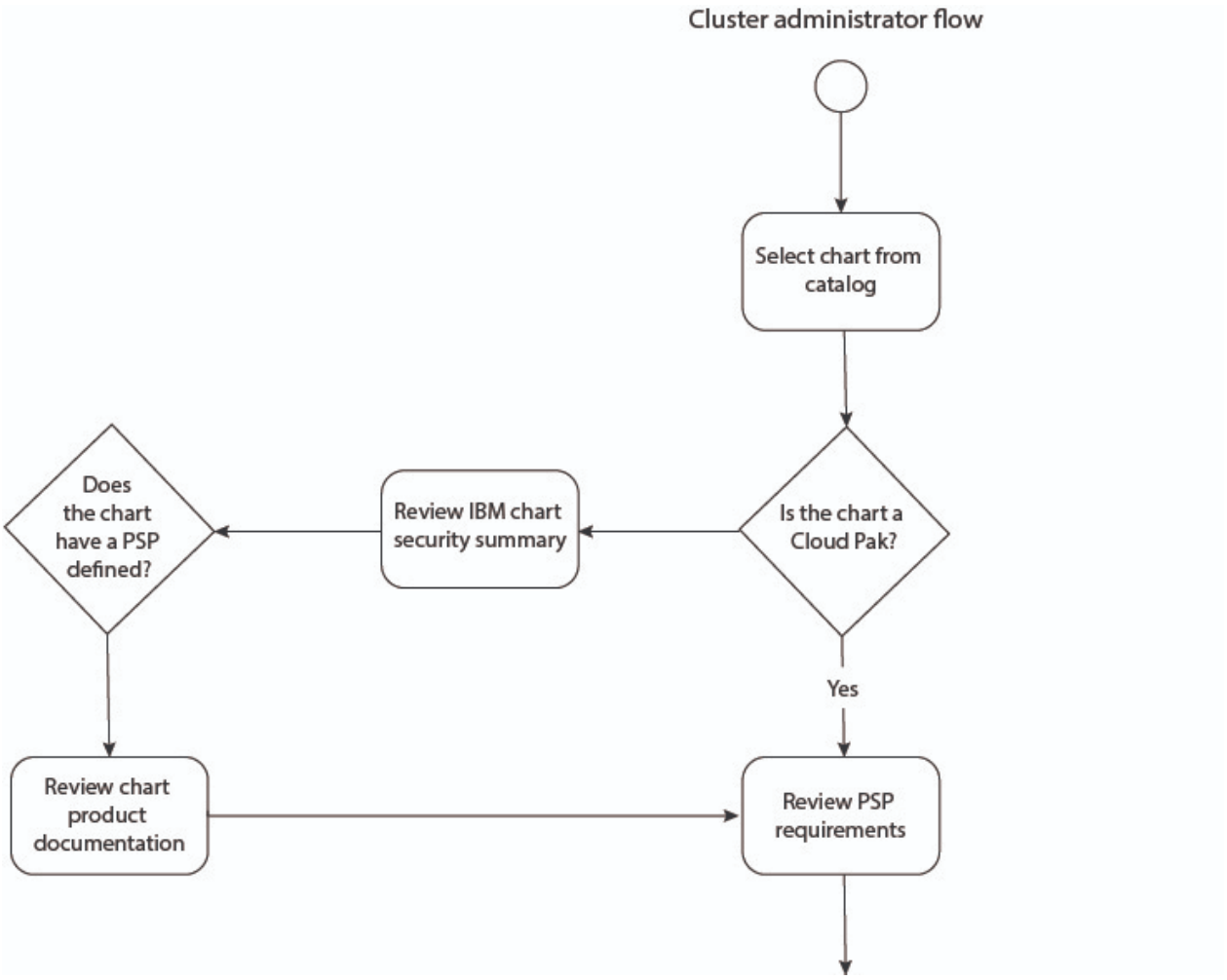
Fluxo de trabalho do administrador de cluster

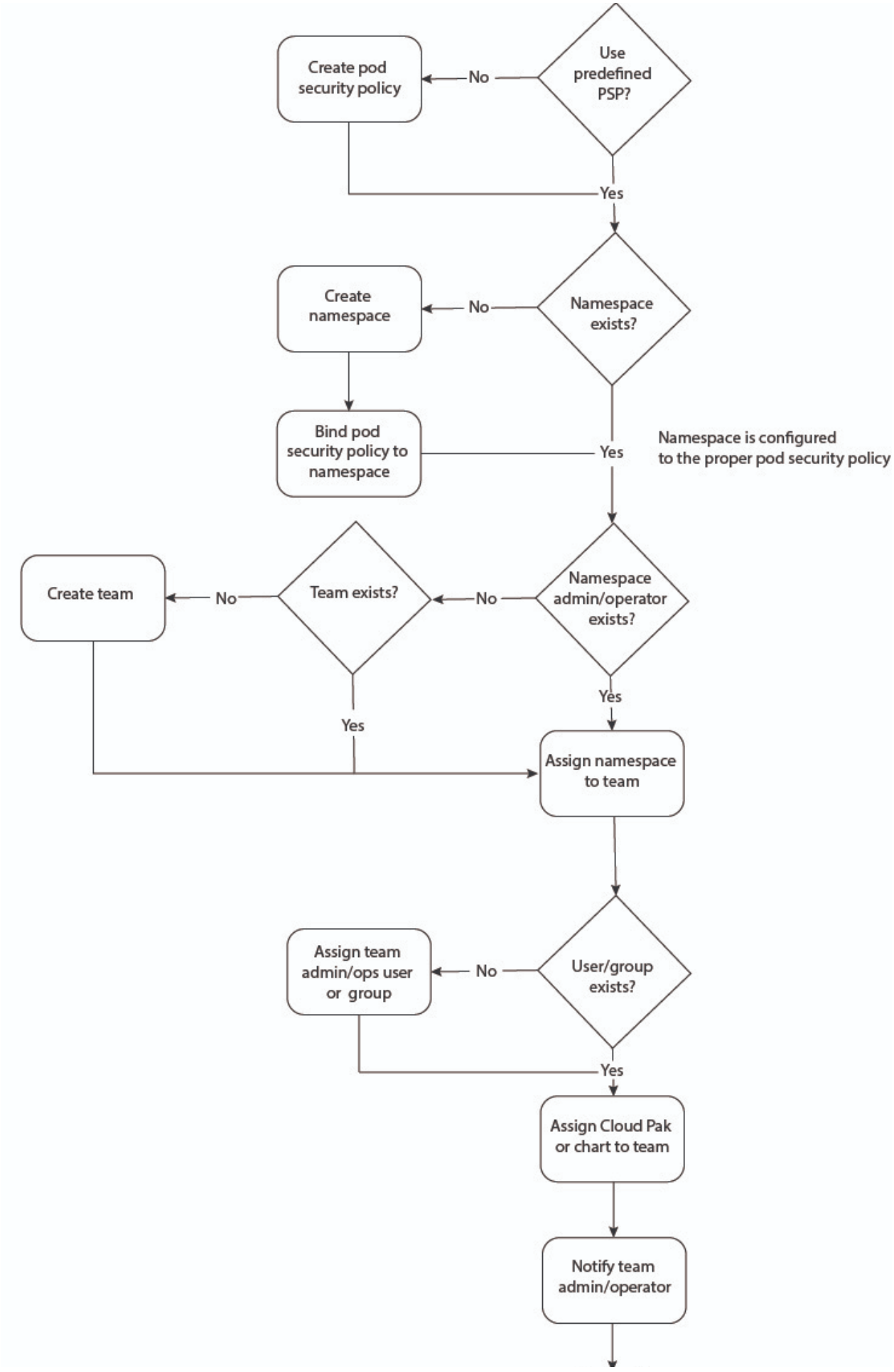
Uma visão geral do fluxo de trabalho para os administradores de cluster.

Pré-requisitos

- A segurança de pod restrita está configurada
- O LDAP está configurado

  Nota: um administrador ou operador de equipe tem a capacidade de instalar um gráfico ou um IBM Certified Container de forma independente.




Use predefined PSP?
No
Create pod security policy
Yes
Namespace exists?
No
Create namespace
Bind pod security policy to namespace
Yes
Namespace is configured to the proper pod security policy
Namespace admin/operator exists?
No
Team exists?
No
Create team
Yes
Yes
Assign namespace to team
User/group exists?
No
Assign team admin/ops user or group
Yes
Assign Cloud Pak or chart to team
Notify team admin/operator

Criando uma política de segurança de pod customizada

É possível usar uma política predefinida a partir do IBM Certified Container ou criar uma política de segurança de pod customizada no YAML. Também é possível criar uma política de segurança de pod customizada usando o IBM Cloud Private console.

Crie uma política de segurança de pod no YAML

É possível criar uma política de segurança de pod criando um arquivo YAML a partir da console de gerenciamento. Conclua as etapas a seguir para criar uma política de segurança de pod no YAML:

1. Revise o arquivo LEIA-ME de Definições de política de segurança do IBM Certified Container.
2. No LEIA-ME do IBM Certified Container, selecione uma política de segurança de pod predefinida para visualizar o conteúdo que é necessário para a política de segurança de pod.
3. Copie o conteúdo da política de segurança de pod selecionada. O conteúdo do YAML pode ser semelhante à saída a seguir:

```
cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: PodSecurityPolicy
metadata:
 name: appsales-anyuid-psp
spec:
 allowPrivilegeEscalation: false
 readOnlyRootFilesystem: false
 allowedCapabilities:
 - CHOWN
 - DAC_OVERRIDE
 - SETGID
 - SETUID
 - NET_BIND_SERVICE
 seLinux:
   rule: RunAsAny
 supplementalGroups:
   rule: RunAsAny
 runAsUser:
   rule: RunAsAny
 fsGroup:
   rule: RunAsAny
 volumes:
 - configMap
 - secret
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 name: appsales-anyuid-psp-clusterrole
rules:
- apiGroups:
 - extensions
 resourceNames:
 - appsales-anyuid-psp
 resources:
 - podsecuritypolicies
 verbs:
 - use
EOF
```

4. Na seção `ClusterRole` do arquivo YAML, você deve especificar o nome da política de segurança de pod para os parâmetros a seguir:
   - Insira o nome da política de segurança de pod com o sufixo `-clusterrole` para o parâmetro `metadata.name`.
   - Insira o nome da política de segurança de pod para o parâmetro `rules[].apiGroups.resourceNames[]`.
5. Clique em **Criar Recurso** e cole o conteúdo da política de segurança de pod.
6. Para criar a política, clique em **Criar** .

7. Crie um namespace e designe a política que você criou para o novo namespace. Para obter mais informações, consulte Criando um namespace para conhecer as etapas necessárias.

Uma política de segurança de pod que está ligada a um namespace é criada em YAML.

Crie uma política de segurança de pod customizada a partir da console de gerenciamento

Se você não usar uma política de segurança de pod predefinida, será possível criar uma política de segurança de pod customizada a partir da console de gerenciamento. Conclua as etapas a seguir para criar um PodSecurityPolicy customizado que pode ser ligado a um namespace:

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Gerenciar** > **Segurança do Recurso** > **Segurança de Pod**.
3. Na página Segurança de Pod, clique em **Criar Política de Segurança de Pod**.
4. Na caixa de diálogo Criar Segurança de Pod, forneça os detalhes da política a seguir:
    - Guia **Geral**
        - Nome - O nome da política que você deseja criar.
        - Tipo - O tipo de política que você está criando por meio das seguintes opções:
            - Privilegiado - Concede aos contêineres de pod quase o mesmo nível de acesso que o concedido a um processo executado no host.
        - Rede do host - Permite que os pods compartilhem o namespace de rede do nó.
        - PID do host - Permite que os contêineres de pod compartilhem o namespace do ID do processo de host.
        - IPC de host - Permite que os contêineres de pod compartilhem o namespace IPC do host.
        - Sistema de arquivos raiz somente leitura - Requer que contêineres sejam executados com um sistema de arquivos raiz somente leitura.
            - Guia **Portas do host**
        - Intervalo de portas do host - Inclua o intervalo de portas do host que estão acessíveis no namespace de rede do host. Não há portas acessíveis quando nada é incluído.
            - Guia **Volumes e sistemas de arquivos**
        - Volume - Forneça o nome de um volume ou entrada disponível (*) para selecionar todos os volumes disponíveis.
            - Alocando um FSGroup que possui os volumes do pod - Seleciona se você deseja especificar qual grupo pode executar o pod, ou qualquer grupo pode executá-lo.
            - Guia **Usuários e grupos**
        - Executar o contêiner como usuário - Especifique o limite do acesso do usuário para executar o contêiner.
            - Configurando grupos suplementares permitidos - Especifica se os grupos suplementares que podem acessar a política devem ser limitados.
            - Guia **SELinux**
        - Contexto de SELinux do contêiner - Especifique se opções específicas do SELinux são necessárias ou se elas podem ser executadas com qualquer opção.
            - Guia **Capacidades**
        - Especifique qualquer recurso que você deseja requerer, permitir ou descartar do contêiner.
5. Ao concluir a customização de sua política de segurança de pod, é possível clicar na régua de controle **Modo JSON** para visualizar o modo JSON de sua política de segurança de pod. A política de segurança de pod pode ser semelhante à saída a seguir:

```
{
 "kind": "PodSecurityPolicy",
 "apiVersion": "extensions/v1beta1",
 "metadata": {
    "name": "appsales-anyuid-psp"
 }
 "spec":{
   "fsGroup": {
     "rule": "RunAsAny"
   },
   "runAsUser": {
      "rule": "RunAsAny",
      "ranges": [
      {
       "min": *,
       "max": *
       }
      ]
```

```
    },
    "seLinux": {
      "rule": "RunAsAny",
      "seLinuxOptions": {
       "level": "-",
       "type": "-",
       "user": "-",
       "role": "-"
},
"supplementalGroups": {
   "rule": "RunAsAny"
    "ranges": [
   {
     "min": *,
     "max": *
   }
 ]
},

 "privileged": "false"
 "readOnlyRootFilesystem": "false"
 "allowedCapabilities": [
   "CHOWN"
 - "DAC_OVERRIDE"
 - "SETGID"
 - "SETUID"
 - "NET_BIND_SERVICE"
```

6. Clique em **Criar**.

7. Clique no ícone **Abrir e fechar lista de opções** para editar o modo JSON de sua política de segurança de pod.

Depois de criar a política de segurança de pod, é possível ligar a política a um namespace. Para obter mais informações, consulte Criando namespaces com a ligação de segurança de pod. Se você usar a console de gerenciamento para criar um novo namespace, a política de segurança de pod será automaticamente ligada ao namespace. A política de segurança de pod pode ser ligada por qualquer usuário, conta de serviço ou namespace.

Criando namespaces com a ligação de política de segurança de pod

É possível criar um namespace e ligá-lo a uma política de segurança de pod com o IBM® Cloud Privateconsole e a linha de comandos.

Crie um namespace com a ligação de política de segurança de pod com o IBM Cloud Privateconsole

Conclua as seguintes etapas para criar um novo namespace e ligá-lo a uma política de segurança de pod:

1. Efetue login em seu cluster do IBM Cloud Private como o administrador de cluster.

2. No menu de navegação, clique em **Gerenciar** > **Namespaces**.

3. Clique no botão **Criar namespace**.

4. Na caixa de diálogo Criar namespace, insira o nome do novo namespace.

5. Clique no menu suspenso *Segurança de pod* e selecione uma política de segurança de pod existente.

6. Clique em **Criar**.

Usando a linha de comandos

Para criar um namespace com uma ligação de política de segurança de pod com a linha de comandos, conclua as etapas a seguir:

1. Configure a linha de comandos `kubectl`.

2. Crie um namespace. Por exemplo, crie um namespace `appsales`. Execute este comando:

```
kubectl create namespace appsales
```

3. Ligue o PodSecurityPolicy `ibm-anyuid-psp` a todas as contas de serviço no exemplo de namespace `appsales`. Execute o comando a seguir:

```
 kubectl -n appsales create rolebinding ibm-anyuid-clusterrole-rolebinding --clusterrole=ibm-anyuid-clusterrole --group=system:serviceaccounts:appsales
```

Um namespace é criado com ligações de política de segurança de pod.

Tarefas do operador da equipe

O operador da equipe pode instalar um gráfico ou o IBM Certified Container.

Aprenda a instalar um gráfico ou o IBM Certified Container.

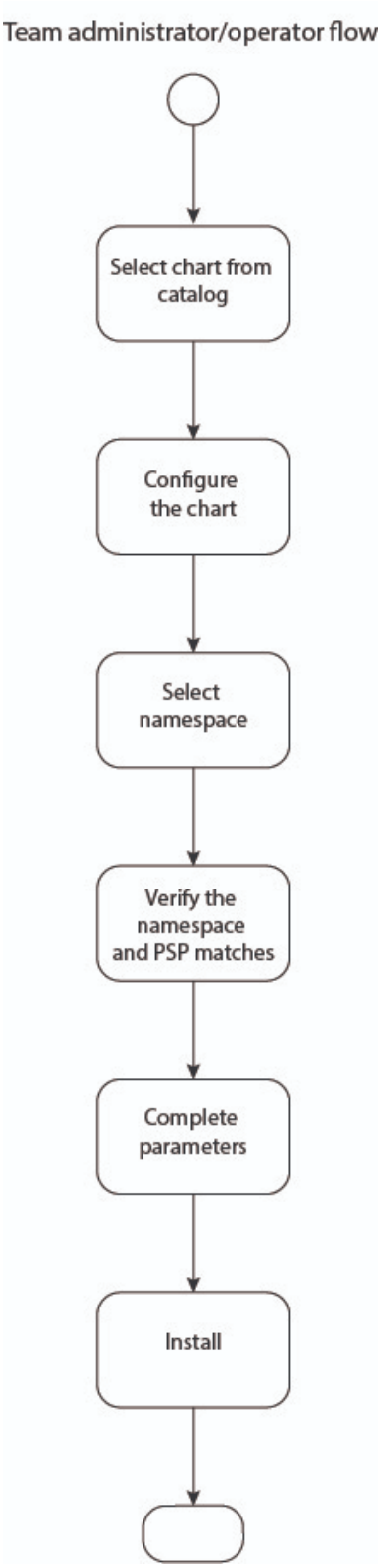
- Instalar o gráfico

Fluxo de trabalho de tarefas do operador da equipe

Uma visão geral do fluxo de trabalho para administradores e operadores da equipe.

Pré-requisitos

- Namespace criado
- A política de segurança de pod está ligada ao namespace
- O namespace está incluído na equipe
- O gráfico está incluído na equipe



Instale o gráfico

Um usuário com a função de operador da equipe pode instalar um gráfico depois que o administrador de cluster cria um namespace para uma política de segurança de pod.

Conclua as etapas a seguir para instalar os gráficos a partir da console do IBM Cloud Private:

1. Efetue login em seu IBM Cloud Private como um operador da equipe.

2. Clique em *Catalog*.
3. Selecione o IBM Certified Container ou o gráfico Helm a ser instalado.
4. Clique em *Configurar*.
5. Na guia *Visão geral*, visualize a seção *Segurança de pod* para o namespace e a Política de Segurança de Pod apropriados.
6. No menu suspenso *Namespace de destino*, selecione o namespace na seção *Segurança de pod*.
    - Se o gráfico for um IBM Certified Container, será necessário concluir uma das opções a seguir:
        1. Escolha um namespace que tenha uma política de segurança de pod predefinida.
        2. Escolha um namespace que tenha uma política de segurança de pod que seja compatível com o gráfico ou com o IBM Certified Container.
7. Conclua outros parâmetros, conforme necessário.
8. Clique em *Instalar* .
    1. Decida se deve usar uma política de segurança de pod predefinida ou criar uma política customizada.
        1. Crie uma política de segurança de pod customizada e uma Função de Cluster, se desejado.
    2. Crie um namespace para o gráfico instalado.
    3. Ligue a política de segurança de pod a um namespace com uma Ligação de Função
    4. Designe o namespace e o gráfico a uma equipe.
9. Comunique as informações a seguir com um administrador ou operador da equipe, descrevendo:
    - Os gráficos a serem instalados.
    - Os namespaces nos quais fazer a instalação.

Um gráfico está instalado.

Configurações de cluster

Aprenda a configurar seu cluster.

- Gerenciando repositórios Helm
- Configurando a cota de recurso
- Gerenciando senhas de segredos do Kubernetes com a CLI do IBM Cloud Private
- Mudando as credenciais de acesso do administrador do cluster
- Customizando a URL de acesso ao cluster
- Configurando a validade do token de acesso e de identidade
- Mudando o intervalo de tempo de atualização de mapeamentos de função de segurança que é usado durante a autorização
- Alterando valores da variável do cache de procura LDAP
- Certificados no IBM Cloud Private
- Configurando o Key Management Service
- Criptografando segredos do Kubernetes com o plug-in Key Management Service

Gerenciando repositórios Helm

Os repositório Helm internos são usados para armazenar pacotes ou gráficos Helm.

- Incluindo um repositório Helm
- Sincronizando repositórios Helm
- Fazendo Backup de Repositórios Helm
- Excluindo um repositório Helm

Incluindo um repositório Helm

Inclua um repositório.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

1. No menu de navegação, clique em **Gerenciar** > **Repositórios de Helm**. Repositórios padrão estão disponíveis.
2. Clique em **Incluir Repositório**.
3. Insira os detalhes do repositório.
   - **Nome:** Insira um nome de repositório exclusivo.
   - **URL:** insira a URL de repositório do GitHub para o seu arquivo index.yaml. É possível se conectar a ambos os repositórios `http` e `https`. Para conectar-se ao repositório de gráfico do Kubernetes de Helm, insira `https://kubernetes-charts.storage.googleapis.com/`.
   - **URL de origem:** URL do GitHub para os seus gráficos. Este campo não é necessário.
4. Clique em **Incluir**. É possível localizar os seus gráficos no Catalog.

Sincronizando repositórios do Helm

Um repositório do Helm interno é atualizado a partir de um repositório do Helm idêntico que é hospedado externamente. As versões dos gráficos do Helm que estão disponíveis no repositório do Helm local são determinadas pela última vez em que o repositório foi atualizado. Os repositórios são atualizados automaticamente apenas quando o repositório é incluído ou quando um gráfico é incluído ou transferido por upload para um repositório externo. Quaisquer atualizações fora desses tempos devem ser solicitadas manualmente para assegurar que você tenha a lista mais recente de gráficos.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Sincronizando todos os repositórios do Helm

1. No menu de navegação, clique em **Gerenciar** > **Repositórios do Helm**. Repositórios padrão estão disponíveis. A entrada da coluna *Última atualização* é a data em que esse repositório foi atualizado pela última vez.
2. Clique em **Sincronizar repositórios** . Se novos gráficos forem transferidos por download por meio dos repositórios externos, será possível localizar os novos gráficos no Catalog.

Sincronizando um único repositório do Helm

1. No menu de navegação, clique em **Gerenciar** > **Repositórios do Helm**. Repositórios padrão estão disponíveis. A entrada da coluna *Última atualização* mostra quando esse repositório foi atualizado pela última vez.
2. Selecione o menu de ações (**...**) para o repositório que você deseja sincronizar.
3. Selecione **Sincronizar repositório** para iniciar a sincronização do repositório, ou para incluí-lo na fila para ser sincronizado.

Nota: Não é possível sincronizar um repositório que já está sendo sincronizado ou está enfileirado para ser sincronizado. Uma opção para cancelar a sincronização está disponível no menu de ações (**...**) se a solicitação de sincronização existente puder ser cancelada.

Fazendo Backup de Repositórios Helm

Fazer Backup de um Repositório Helm

1. No nó de inicialização, acesse o diretório `/var/lib/icp/helmrepo`.
2. Faça uma cópia dos gráficos que estão nesse diretório. Os arquivos de gráfico são arquivos compactados que terminam em `.tgz` ou `tar.gz`.

Restaure o conteúdo de um repositório Helm

1. Certifique-se de que `helm-repo` esteja em execução e que esse diretório exista: `/var/lib/icp/helmrepo`.
2. Mova os gráficos dos quais você fez backup para o diretório `/var/lib/icp/helmrepo`.
3. Remova o pod de implementação `helm-repo` e, em seguida, o `index.yaml` é preenchido novamente com os gráficos que estão localizados no PersistentVolumeClaim (PVC).

Excluindo um repositório Helm

Remova uma conexão do repositório

Os nomes dos repositórios padrão `ibm-charts` e `local-charts` são reservados. Se você excluir esses repositórios padrão, não será possível incluí-los de volta usando esses nomes reservados. Para restaurar os repositórios padrão, deve-se usar a mesma URL, mas fornecer um nome diferente daqueles reservados.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

1. No menu de navegação, clique em **Gerenciar** > **Repositórios de Helm**.
2. Na mesma linha que o repositório a ser removido, selecione **Ação** > **Excluir**.
3. Clique em **Excluir**.

Configurando a cota de recurso

Configure cotas de recursos para limitar a soma de recursos de cálculo e armazenamento que podem ser solicitados por um aplicativo. As cotas de recursos também podem ser usadas para limitar o número de objetos de um tipo especificado que estão disponíveis em um único namespace.

Para obter informações adicionais sobre como configurar a cota de recursos, consulte *Cotas de recursos* na página de conceitos do Kubernetes.

Para visualizar uma lista de cotas, no menu de navegação, clique em **Gerenciar** > **Cotas**.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando uma cota usando a janela Criar ResourceQuota

1. No menu de navegação, clique em **Gerenciar** > **Cotas**.
2. Clique em **Criar ResourceQuota**.
3. Insira os detalhes de cota de recurso.
4. Clique em **Criar**.

Criando uma cota usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar uma cota usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/policy/resource-quotas/#viewing-and-setting-quotas.

   Ao criar uma cota por meio da janela "Criar recurso", lembre-se de especificar o namespace na seção `metadata`.

   Para as cotas de recursos de GPU, é possível configurar solicitações de recursos para namespaces específicos executando o parâmetro `requests.nvidia.com/gpu`.

   O código de amostra a seguir configura uma cota de solicitação de GPU de 2 GPUs para o namespace `myspace`.

```
apiVersion: v1
kind: ResourceQuota
metadata:
 name: compute-resources
 namespace: myspace
spec:
 hard:
   requests.cpu: "1"
   requests.memory: 1Gi
   limits.cpu: "2"
   limits.memory: 2Gi
   requests.nvidia.com/gpu: "2"
```

Nota: devido à limitação de Kubernetes, deve-se configurar `requests.nvidia.com/gpu` para permitir que a cota GPU configure solicitações de recursos.

1. Clique em **Criar**.

Após a conclusão da implementação, uma nova cota é exibida na página Cotas.

Gerenciando senhas secretas do Kubernetes com a CLI do IBM Cloud Private

É possível cumprir os requisitos de senha, mudar senhas e reiniciar o pods e contêineres necessários usando a CLI do IBM Cloud Private para um conjunto de Segredos.

Pré-requisitos

- **Tipo de usuário ou nível de acesso necessário:** administrador de cluster
- Instale e configure a ferramenta de linha de comandos do Kubernetes, kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
- Instale a CLI do IBM Cloud Private. Consulte Instalando a CLI do IBM Cloud Private para iniciar.
- Efetue login no IBM Cloud Private para gerar tokens. O comando solicita uma senha, uma conta e um namespace.

```
cloudctl login -a https://<cluster-domain-name>:8443 -u <username> --skip-ssl-validation
```

Gerenciando senhas

É possível mudar várias senhas do IBM Cloud Private que são armazenadas em Segredos do Kubernetes. A CLI do IBM Cloud Private tem um comando para mudar a senha e reiniciar os serviços que usam a senha.

1. Escolha na lista de segredos a seguir qual senha você deseja mudar:
    - Credenciais platform-auth-idp-: As credenciais de acesso do administrador de cluster
    - icp-mongodb-admin: a senha interna para serviços que estão usando o Mongo
    - platform-oidc-credentials: a senha interna para serviços de autenticação
2. Execute o comando para o Segredo que você deseja mudar. Você receberá um prompt para um valor de senha e um prompt para confirmar a mudança.
    - As credenciais platform-auth-idp-

      ```
      cloudctl pm update-secret kube-system platform-auth-idp-credentials -d admin_password
      ```

    - Icp-mongodb-admin

      ```
      cloudctl pm update-secret kube-system icp-mongodb-admin -d password
      ```

    - Credenciais-oidc-plataforma

      ```
      cloudctl pm update-secret kube-system platform-oidc-credentials -d WLP_CLIENT_SECRET
      ```

Opcional: gerenciando regras de senha

As regras de senha são expressões regulares (regex) opcionais que são usadas para configurar e validar os valores de senha gerenciados dentro de um namespace. O comando `update-secret` verifica as regras de senha antes de mudar os valores no segredo. Ele valida se as regras existem no namespace em que o segredo está e se a mudança de nome de elemento de dados secretos contém `pass` ou `pwd` em qualquer combinação de caso.

Nota: é possível usar as expressões regex a seguir na linha de comandos. Assegure-se de agrupar as expressões entre aspas simples:

Tabela 1. Lista de regras para expressões Regex

Regra	Regex
Comprimento Mín 10	`^. {10,}`
Comprimento Máx 10	`^. {0,10}$`
Comprimento intervalo	`^. {10,20}$`
Requer menor	`.*[a-z].*`
Requerer superior	`.*[A-Z].*`

Regra	Regex
Requer número	`.*[0-9].*`
Requerer spec	`.*[!@#\$%\^&\*].*`

1. Configure as regras de senha com o comando a seguir:

   ```
   cloudctl pm password-rule-set <namespace> <rule_name> <rule_regex> <rule_desc>
   ```

   Consulte o exemplo a seguir de uma expressão Regex:

   ```
   cloudctl pm password-rule-set default min_10 ' ^. {10,}' "minimum length of 10"
   OK
   ```

2. Liste as regras de senha.

   ```
   cloudctl pm password-rules <namespace>
   ```

   Consulte o seguinte exemplo:

   ```
   cloudctl pm password-regras padrão

   Nome    Descrição          Regex
   min10   minimum length 10  ^.{10,}
   OK
   ```

3. Remova a regra de senha.

   ```
   cloudctl pm password-rule-rm <namespace> <rule_name>
   ```

   Consulte o seguinte exemplo:

   ```
   cloudctl pm password-rule-rm default min_10
   OK
   ```

Mudando as credenciais de acesso do administrador de cluster

É possível atualizar o nome de usuário e a senha do administrador de cluster.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Mudando o nome de usuário do administrador de cluster

1. Efetue logon no nó principal do cluster do IBM® Cloud Private.
2. Use a CLI do IBM Cloud Private (cloudctl) para mudar seu nome de usuário e para reiniciar as implementações. Por exemplo:

   ```
   cloudctl pm update-secret kube-system platform-auth-idp-credentials -d admin_username=
   <username>
   ```

   Para obter mais informações, consulte Comandos pm da CLI do IBM Cloud Private (pm).

3. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
4. Atualize o objeto de controle de acesso baseado na função (RBAC) `clusterrolebinding` com o novo nome de usuário.

   ```
   kubectl edit clusterrolebinding oidc-admin-binding
   ```

   A seguir está um objeto RBAC `clusterrolebinding` de amostra:

   ```
   Please edit the following object. Lines beginning with a '#' will be ignored,
   # and an empty file will abort the edit. If an error occurs while saving this file will be
            # reopened with the relevant failures.
   #
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
    creationTimestamp: 2019-02-04T18:44:34Z
    name: oidc-admin-binding
    resourceVersion: "3162"
   ```

```
 selfLink: /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/oidc-admin-binding
 uid: eab9c9c9-28ac-11e9-aca2-0050569a1e29
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
 kind: User
 name: https://mycluster.icp:9443/oidc/endpoint/OP#admin        <===========
- apiGroup: rbac.authorization.k8s.io
 kind: User
 name: admin                                                    <===========
```

5. Substitua o nome do administrador em `https://mycluster.icp:9443/oidc/endpoint/OP#admin` pelo novo nome: mude `OP#admin` para `OP#<new admin user name>`.
6. Substitua o nome do administrador em `name: admin` pelo novo nome: mude `name: admin` para `name: <new admin user name>`.
7. Salve o arquivo.

Alterando a Senha do Administrador de Cluster

1. Efetue logon no nó principal do cluster do IBM Cloud Private.
2. Use a CLI do IBM Cloud Private (cloudctl) para mudar sua senha e reiniciar as implementações. O nome secreto é `platform-auth-idp-credentials` e o namespace é `kube-system`. A nova senha deve atender a uma ou mais regras de cumprimento de senha padrão que são especificadas para os parâmetros `password_rules` no arquivo `config.yaml`. Por exemplo:

   ```
   cloudctl pm update-secret kube-system platform-auth-idp-credentials -d admin_password=
   <password>
   ```

   Para obter mais informações, consulte Customizando o cluster com o arquivo config.yaml e Instalando a CLI do IBM Cloud Private.
3. Atualize o `default_admin_password` no config.yaml.
   1. Abra o arquivo `/<installation_directory>/cluster/config.yaml`.
   2. Atualize o `default_admin_password`.
   3. Salve e saia do arquivo.
4. Opcional: é possível atualizar as regras de senha executando o comando a seguir:

   ```
   cloudctl pm password-rule-set <namespace> <rule_name> <rule_regex> <rule_desc>
   ```

Customizando a URL de acesso ao cluster

Customize o Uniform Resource Locator (URL) que você usa para efetuar login na console de gerenciamento do cluster do IBM® Cloud Private.

- Formatos de customização suportados
- Customize a URL de acesso do cluster com `cloudctl`
- Customize a URL de acesso do cluster com `kubectl`

Formatos de customização suportados

Os formatos de customização a seguir são suportados:

- `https://<Public IP>:8443/console`
- `https://<Public IP>:8443/console/`
- `https://<Private IP>:8443/console/`
- `https://<Private IP>:custom-port/console/`
- `https://<host name>:8443/console`
- `https://<host name>:custom-port/console`
- `https://localhost:8443/console`

- https://localhost:<custom port>/console
- https://<Regex host name>:8443/console
- https://<Regex IP>:8443/console
- https://<Regex host name>:<custom port>/console
- https://<Regex IP>:<custom port>/console
- https://<Regex host name>:<Regex port>/console
- https://<Regex IP>:<Regex Port>/console

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Customize a URL de acesso do cluster com cloudctl

Conclua as tarefas a seguir em qualquer nó do cluster do IBM Cloud Private que tenha o `cloudctl` e o `kubectl` instalados:

- Para obter mais informações sobre como instalar a CLI `cloudctl`, consulte Instalando a CLI do IBM® Cloud Private.
- Para obter mais informações sobre como instalar a CLI `kubectl`, consulte Instalando a CLI do Kubernetes (kubectl).

1. Efetue login no `cloudctl` como um usuário com acesso de administrador de cluster.
2. Salve o ID do cliente com o comando a seguir:

```
WLP_CLIENT_ID=$(kubectl -n kube-system get secret platform-oidc-credentials -o yaml | grep
WLP_CLIENT_ID | awk '{ print $2}' | base64 --decode)
```

3. Obtenha os dados de `platform-oidc-registration.json` com o comando a seguir:

```
cloudctl iam oauth-client $WLP_CLIENT_ID  > platform-oidc-registration.json
```

4. Atualize o arquivo `platform-oidc-registration.json` com o procedimento a seguir:
   O conteúdo do arquivo `platform-oidc-registration.json` é semelhante ao texto a seguir:

```
OK
{
    "allow_regexp_redirects": false,
    "appPasswordAllowed": false,
    "appTokenAllowed": false,
    "application_type": "web",
    "client_id": "515da8f96cc161795a03b77b4129a3f3",
    "client_id_issued_at": 1559051605
    "client_name": "515da8f96cc161795a03b77b4129a3f3",
    "client_secret": "*",
    "client_secret_expires_at": 0,
    "functional_user_groupIds": [],
    "grant_types": [
        "authorization_code",
       "client_credentials",
       "password",
       "implicit",
       "refresh_token",
       "urn:ietf:params:oauth:grant-type:jwt-bearer"
    ],
    "introspect_tokens": true,
    "post_logout_redirect_uris": [
        "https://10.21.9.140:8443/console/logout",
        "https://9.46.76.19:8443/console/logout",
        "https://mycluster.icp:8443/console/logout"
    ],
    "preauthorized_scope": "openid profile email general",
    "redirect_uris": [
        "https://10.21.9.140:8443/auth/liberty/callback",
        "https://9.46.76.19:8443/auth/liberty/callback",
        "https://mycluster.icp:8443/auth/liberty/callback",
        "https://127.0.0.1:9443/oidc/endpoint/OP"
    ],
    "registration_client_uri":
"https://9.46.76.19:8443/oidc/endpoint/OP/registration/515da8f96cc161795a03b77b4129a3f3",
    "resource_ids": [],
    "response_types": [
        "code",
     "token",
     "id_token token"
    ],
```

```
    "scope": "openid profile email",
    "subject_type": "public",
    "token_endpoint_auth_method": "client_secret_basic",
    "trusted_uri_prefixes": [
        "https://10.21.9.140:8443/",
        "https://9.46.76.19:8443/",
        "https://mycluster.icp:8443/"
    ]
}
```

Atualize o conteúdo do arquivo com base nas instruções a seguir:

- Exclua as linhas que são campos somente de saída ou que são dados não JSON. Por exemplo, deve-se remover o texto a seguir da saída de amostra:
  - `OK`
  - `"client_id_issued_at": 1559051605`
  - `"registration_client_uri": "https://9.46.76.19:8443/oidc/endpoint/OP/registration/515da8f96cc161795a03b77b4129a3f3",`
- Assegure-se de que `"allow_regexp_redirects":` esteja configurado como `"true"`.
  Depois de atualizar o arquivo, o conteúdo será semelhante ao texto a seguir:

```
{
"allow_regexp_redirects": false,
"appPasswordAllowed": false,
"appTokenAllowed": false,
"application_type": "web",
"client_id": "515da8f96cc161795a03b77b4129a3f3",
"client_name": "515da8f96cc161795a03b77b4129a3f3",
"client_secret": "*",
"client_secret_expires_at": 0,
"functional_user_groupIds": [],
"grant_types": [
    "authorization_code",
   "client_credentials",
   "password",
   "implicit",
   "refresh_token",
   "urn:ietf:params:oauth:grant-type:jwt-bearer"
],
"introspect_tokens": true,
"post_logout_redirect_uris": [
    "https://10.21.9.140:8443/console/logout",
    "https://9.46.76.19:8443/console/logout",
    "https://mycluster.icp:8443/console/logout"
],
"preauthorized_scope": "openid profile email general",
"redirect_uris": [
    "https://10.21.9.140:8443/auth/liberty/callback",
    "https://9.46.76.19:8443/auth/liberty/callback",
    "https://mycluster.icp:8443/auth/liberty/callback",
    "https://127.0.0.1:9443/oidc/endpoint/OP"
],
"resource_ids": [],
"response_types": [
    "code",
 "token",
 "id_token token"
],
"scope": "openid profile email",
"subject_type": "public",
"token_endpoint_auth_method": "client_secret_basic",
"trusted_uri_prefixes": [
    "https://10.21.9.140:8443/",
    "https://9.46.76.19:8443/",
    "https://mycluster.icp:8443/"
]
}
```

5. Inclua seus URIs customizados na seção `"redirect_uris"` do arquivo `platform-oidc-registration.json`. Consulte Formatos de customização suportados para os tipos de URIs que você pode incluir. Execute o comando a seguir:

```
"<regexp>:https://<custom IP address or host name>:<custom port>/auth/liberty/callback",
```

Nota: inclua `<regexp>:` somente se você estiver usando um regex no URI customizado.

Considere os seguintes URIs de exemplo que você deseja usar para acessar o cluster:

- Use o endereço IP do nó principal e qualquer porta que inicie com `84` e, em seguida, inclua `"regexp:https://<master node IP address>:84!d!d/auth/liberty/callback"`.
- Use o nome do host `example.abc.com` e a porta `4002` e, em seguida, inclua `"https://example.abc.com:4002/auth/liberty/callback"`.
- Use um nome de host variável e uma designação de porta dinâmica e, em seguida, adicione `"regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback"`.

Se você incluiu os URIs customizados de exemplo, o código atualizado será semelhante ao texto a seguir:

```
...
"application_type":"web",
   "subject_type":"public",
   "post_logout_redirect_uris":[

"https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://m
ycluster.icp:8443/console/logout" ], "preauthorized_scope":"openid profile email general",
"introspect_tokens":true, "trusted_uri_prefixes":[
  "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443" ],
"redirect_uris":[
  "regexp:https://10.10.25.213:84!d!d/auth/liberty/callback",       <==========
  "https://example.abc.com:4002/auth/liberty/callback",         <==========
      "regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback",          <==========

"https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callba
ck","https://mycluster.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoin
t/OP"    ]
}
```

6. Execute o comando a seguir para aplicar suas mudanças:

```
cloudctl iam oauth-client-update -f platform-oidc-registration.json
```

Customize a URL de acesso do cluster com kubectl

Conclua as tarefas a seguir no nó de inicialização de seu cluster do IBM Cloud Private.

1. Efetue login no nó de inicialização como um usuário com permissões raiz.
2. Configure a CLI `kubectl` . Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

3. Copie o conteúdo que está no mapa de configuração `registration-json` para o arquivo `registration.yaml`.

```
kubectl get cm registration-json -n kube-system -o yaml > registration.yaml
```

O conteúdo do arquivo `registration.yaml` é semelhante ao código a seguir:

```
apiVersion: v1
data:
 platform-oidc-registration.json: |
   {
   "token_endpoint_auth_method":"client_secret_basic", "client_id":
"d2a00fc99163f85169ac7c6de758bad1", "client_secret": "01661d22bd0b2025fd87e26e994a4894",
"scope":"openid profile email", "grant_types":[
      "authorization_code",
      "client_credentials",
      "password",
      "implicit",
      "refresh_token",
      "urn:ietf:params:oauth:grant-type:jwt-bearer"
   ],
   "response_types":[
      "code",
      "token",
      "id_token token"
   ],
   "application_type":"web",
   "subject_type":"public",
   "post_logout_redirect_uris":[
```

```
"https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://m
ycluster.icp:8443/console/logout"    ],
   "introspect_tokens":true,
   "trusted_uri_prefixes":[
      "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443" ],
"redirect_uris":[

"https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callba
ck","https://mycluster.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoin
t/OP"    ]
   }
kind: ConfigMap
metadata:
 creationTimestamp: 2018-06-06T11:53:21Z
 name: registration-json
 namespace: kube-system
 resourceVersion: "1255"
 selfLink: /api/v1/namespaces/kube-system/configmaps/registration-json
 uid: 3620b003-6980-11e8-9420-fa163ea0dafe
```

4. Crie um arquivo `platform-oidc-registration.json`. Coloque o arquivo na pasta `<installation directory>/cluster/cfc-components/`.

5. Copie o conteúdo que está na seção `data:` do arquivo `registration.yaml` para o arquivo `platform-oidc-registration.json`. O conteúdo do arquivo `platform-oidc-registration.json` é semelhante ao código a seguir:

```
  {
   "token_endpoint_auth_method":"client_secret_basic", "client_id":
"d2a00fc99163f85169ac7c6de758bad1", "client_secret": "01661d22bd0b2025fd87e26e994a4894",
"scope":"openid profile email", "grant_types":[
      "authorization_code",
      "client_credentials",
      "password",
      "implicit",
      "refresh_token",
      "urn:ietf:params:oauth:grant-type:jwt-bearer"
   ],
   "response_types":[
      "code",
      "token",
      "id_token token"
   ],
   "application_type":"web",
   "subject_type":"public",
   "post_logout_redirect_uris":[

"https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://m
ycluster.icp:8443/console/logout" ], "preauthorized_scope":"openid profile email general",
"introspect_tokens":true, "trusted_uri_prefixes":[
      "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443" ],
"redirect_uris":[

"https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callba
ck","https://mycluster.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoin
t/OP"    ]
   }
```

6. Inclua a parte de código a seguir no arquivo `platform-oidc-registration.json`:

```
"allow_regexp_redirects":"true",
```

O código atualizado assemelha-se ao texto a seguir:

```
{
 "token_endpoint_auth_method":"client_secret_basic",
 "client_id": "d2a00fc99163f85169ac7c6de758bad1",
 "client_secret": "01661d22bd0b2025fd87e26e994a4894",
 "scope":"openid profile email",
 "allow_regexp_redirects":"true",                                  <==========
 "grant_types":[
    "authorization_code",
        "client_credentials",
        "password",
        "implicit",
```

```
        "refresh_token",
        "urn:ietf:params:oauth:grant-type:jwt-bearer"
    ...
```

7. Inclua seus URIs customizados na seção "redirect_uris" do arquivo platform-oidc-registration.json. Consulte Formatos de customização suportados para os tipos de URIs que você pode incluir.

   ```
   "<regexp>:https://<custom IP address or host name>:<custom port>/auth/liberty/callback",
   ```

   Em que você inclui <regexp>: somente se estiver usando um regex no URI customizado.

   Considere os seguintes URIs de exemplo que você deseja usar para acessar o cluster:

   - Use o endereço IP do nó principal e qualquer porta que começa com 84. Portanto, você incluiria "regexp:https://<master node IP address>:84!d!d/auth/liberty/callback".
   - Use o nome do host example.abc.com e porta 4002. Você poderia, então, incluir "https://example.abc.com:4002/auth/liberty/callback".
   - Use um nome do host da variável e uma designação de porta dinâmica. Você poderia, então, incluir "regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback".

   Se você incluiu os URIs customizados de exemplo, o código atualizado será semelhante ao texto a seguir:

   ```
   ...
   "application_type":"web",
      "subject_type":"public",
      "post_logout_redirect_uris":[

   "https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://m
   ycluster.icp:8443/console/logout" ], "preauthorized_scope":"openid profile email general",
   "introspect_tokens":true, "trusted_uri_prefixes":[
     "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443" ],
   "redirect_uris":[
     "regexp:https://10.10.25.213:84!d!d/auth/liberty/callback",       <==========
     "https://example.abc.com:4002/auth/liberty/callback",         <==========
         "regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback",          <==========

   "https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callba
   ck","https://mycluster.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoin
   t/OP"    ]
   }
   ```

8. Salve e saia do arquivo.

9. Salve o ID do cliente, o segredo do cliente e o IP de acesso nas variáveis a seguir:

   1. Salve o segredo do cliente:

      ```
      OAUTH2_CLIENT_REGISTRATION_SECRET=$(kubectl -n kube-system get secret platform-oidc-
      credentials -o yaml | grep OAUTH2_CLIENT_REGISTRATION_SECRET | awk '{ print $2}' | base64
      --decode)
      ```

   2. Salve o ID do cliente:

      ```
      WLP_CLIENT_ID=$(kubectl -n kube-system get secret platform-oidc-credentials -o yaml |
      grep WLP_CLIENT_ID | awk '{ print $2}' | base64 --decode)
      ```

   3. Salve o IP de acesso:

      ```
      FIP=<master node IP address>
      ```

10. Aplique as mudanças que você fez no arquivo platform-oidc-registration.json.

    ```
     curl -kvv -X PUT -u oauthadmin:$OAUTH2_CLIENT_REGISTRATION_SECRET -H "Content-Type:
    application/json" -d @<installation directory>/cluster/cfc-components/platform-oidc-
    registration.json https://$FIP:8443/idauth/oidc/endpoint/OP/registration/$WLP_CLIENT_ID
    ```

Editar cabeçalhos de host permitidos

Se você tiver mudado o nome do host para acessar o cluster, será necessário modificar os cabeçalhos de host permitidos no icp-management-ingress do DaemonSet.

1. Edite o icp-management-ingress do DaemonSet executando o comando a seguir:

```
kubectl edit ds -n kube-system icp-management-ingress
```

2. Edite a variável de ambiente `ALLOWED_HOST_HEADERS` no DaemonSet. Por exemplo, inclua o novo nome do host `mycluster.icp.new` na URL de acesso ao cluster e, em seguida, mude a variável de ambiente de:

```
env:
- name: ALLOWED_HOST_HEADERS
value: 10.10.25.213 9.37.239.32 mycluster.icp icp-management-ingress icp-management-
ingress.kube-system
```

Para:

```
env:
- name: ALLOWED_HOST_HEADERS
value: 10.10.25.213 9.37.239.32 mycluster.icp mycluster.icp.new icp-management-ingress icp-
management-ingress.kube-system
```

O pod de icp-management-ingresso é reinicializado.

Agora é possível acessar a console de gerenciamento com a nova URL.

Configurando a porta NodePort para instalar gráficos do Helm

O método padrão que o IBM® Cloud Private usa para instalar os gráficos do Helm com a CLI do Helm é usar uma conexão em proxy, que requer que o usuário tenha autorização para o namespace *kube-system*. O serviço <o Helm Tiller também pode ser configurado para usar um NodePort para ignorar a conexão em proxy padrão e permitir que os usuários usem a CLI do Helm sem precisar de acesso ao namespace *kube-system*. <dž

Alterando o valor da porta NodePort

O valor `tiller_nodeport` no arquivo `config.yaml` especifica a porta NodePort que é usada para conectar-se ao Tiller. Ao instalar o IBM Cloud Private, o NodePort para o Tiller é configurado como um valor padrão de `31514` no arquivo `config.yaml`. É possível atualizar o valor de NodePort após a implementação alterando o valor `nodePort` no serviço `tiller-deploy`.

```
tiller_nodeport: 31573
```

Para obter informações adicionais sobre NodePort, consulte Serviços na documentação do Kubernetes e Portas necessárias.

Comunicando com o Tiller usando a porta NodePort

Para acessar o Tiller usando a NodePort configurada, é possível configurar uma variável de ambiente `HELM_HOST` em sua janela do terminal ou especificar a opção `--host` na CLI do Helm. O formato para ambos é: *NodeIP:NodePort* . Por exemplo, se o NodeIP estiver configurado como `10.20.247.65` e a NodePort estiver configurada como `31514`, o valor será `10.20.247.65:31514`.

- A melhor prática é configurar a CLI do Helm, conforme mostrado no exemplo a seguir:

```
eval "$(cloudctl helm-init)"
helm list --tls
```

Este procedimento configura a variável ambiental `HELM_HOST para o terminal e a porta corretos do Tiller automaticamente. **Nota:** o comando para configurar a variável de ambiente deve ser inserido toda vez que você efetua login no cluster.

- É possível configurar a variável de ambiente `HELM_HOST` manualmente inserindo um comando semelhante ao exemplo a seguir:

```
export HELM_HOST=10.20.247.65:31514
helm list --tls
```

Após a variável `HELM_HOST` ser configurada, você não precisa inserir o comando `HELM_HOST` novamente durante essa sessão. **Nota:** o comando para configurar a variável de ambiente deve ser inserido toda vez que você efetua login no cluster.

- É possível incluir a opção `--host` em seu comando para usar a nodePort configurada inserindo um comando semelhante ao exemplo a seguir:

```
helm list -- host 10.20.247.65:31514 -- tls
```

Ativando a conformidade com o FIPS

A comunicação com o Tiller por meio da nodePort ***não*** é compatível com o FIPS. Para obter a comunicação compatível com o FIPS com nodePort, deve-se usar a console de gerenciamento do IBM Cloud Private para instalar seus gráficos do Helm ou ativar os conjuntos de cifras compatíveis com o FIPS para criptografar sua comunicação.

A implementação do TLS usada na linguagem Go não é compatível com o FIPS. Os clientes podem usar a console de gerenciamento do IBM Cloud Private ou usar o suporte de proxy do Kubernetes sem nodePort para permanecer em conformidade com o FIPS.

É possível configurar os conjuntos de cifras para `tiller` no arquivo `config.yaml` ao instalar o IBM Cloud Private. A configuração de `tiller_ciphersuites` fornece um conjunto de valores separados por vírgulas que definem os conjuntos de cifras que o `Tiller` pode usar para criptografar a comunicação. O valor padrão de `""` significa que o `tiller` pode usar qualquer um dos conjuntos de cifras disponíveis. Os valores suportados para conjuntos de cifras são mostrados na lista a seguir:

- , TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- , TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- , TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_HEDHE_RSA_WITH_RC4_128_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305

Um exemplo da configuração de conjuntos de cifras de tiller no arquivo `config.yaml` é mostrado no conteúdo a seguir:

```
  tiller_ciphersuites: "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHEDHE_ECDSA_WITH_AES_128_CBC_CBC_SHA256"
```

Desativando a Porta NodePort

É possível desativar a porta NodePort concluindo as etapas a seguir:

1. No menu do IBM Cloud Private, navegue para **Acesso à Rede** > **Serviços**.
2. Localize o serviço `tiller-deploy`.
3. Edite o serviço selecionando **Editar** a partir do menu *Ações* do serviço. As informações de serviço são exibidas no editor.
4. Remova o texto a seguir dos valores de `portas`:

   ```
   "nodePort": 31514
   ```

5. Mude o valor de `type` para `ClusterIP`. O conteúdo resultante é semelhante ao exemplo a seguir:

   ```
   {
     "apiVersion": "v1",
     "kind": "Service",
     "metadata": {
       "name": "tiller-deploy",
       "namespace": "kube-system",
       "resourceVersion": "7619",
       "labels": {
   ```

```
      "app": "helm",
      "name": "tiller"
    },
    "annotations": {
      "kubectl.kubernetes.io/last-applied-configuration": "
{\"apiVersion\":\"v1\",\"kind\":\"Service\",\"metadata\":{\"annotations\":{},\"labels\":
{\"app\":\"helm\",\"name\":\"tiller\"},\"name\":\"tiller-deploy\",\"namespace\":\"kube-
system\"},\"spec\":{\"clusterIP\":\"10.0.0.9\",\"ports\":
[{\"name\":\"grpc\",\"port\":44134,\"protocol\":\"TCP\",\"targetPort\":44134}],\"selector\":
{\"app\":\"helm\",\"name\":\"tiller\"}}}\n"
    }
  }, "spec": {
   "ports": [
     {
       "name": "grpc",
       "protocol": "TCP",
       "port": 44134,
       "targetPort": 44134,
       }
    ],
    "selector": {
      "app": "helm",
      "name": "tiller"
    },
    "clusterIP": "10.0.0.9",
    "type": "ClusterIP",
    "sessionAffinity": "None",
    "externalTrafficPolicy": "Cluster"
  }
}
```

6. Clique em **Submeter** para confirmar as mudanças.

Configurando a validade do token de acesso e identidade

Mude a configuração padrão da validade do token de acesso e do token de identidade.

Por padrão, o token de acesso e o token de identidade (ID) são válidos por 12 horas após sair da console de gerenciamento ou fechar o navegador. É possível mudar esse valor padrão conforme necessário.

Siga estas etapas para mudar a validade do token:

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Edite o configmap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

   O conteúdo do arquivo é semelhante ao código a seguir:

   ```
   # Please edit the following object. Lines beginning with a '#' will be ignored,
   # and an empty file will abort the edit. If an error occurs while saving this file will be
   # reopened with the relevant failures.
   #
   apiVersion: v1
   data:
    AUDIT_ENABLED_IDMGMT: "false"
    AUDIT_ENABLED_IDPROVIDER: "false"
    BASE_AUTH_URL: /v1
    BASE_OIDC_URL: https://127.0.0.1:8443/idauth/oidc/endpoint/OP
    CLUSTER_NAME: mycluster
    HTTP_ONLY: "true"
    IDENTITY_AUTH_DIRECTORY_URL: http://127.0.0.1:3100
    IDENTITY_PROVIDER_URL: http://127.0.0.1:4300
    IDTOKEN_LIFETIME: 12h
    JOURNAL_PATH: /run/systemd/journal
    MASTER_HOST: mycluster.icp
    NODE_ENV: production
    OAUTH2DB_DB_HOST: mongodb
    OAUTH2DB_DB_PORT: "3306"
    OIDC_ISSUER_URL: https://mycluster.icp:8443/oidc/endpoint/OP
    SESSION_TIMEOUT: "43200"
   ```

```
logrotate: |-
  /var/log/audit/*.log {
    su nobody root
.
.
.
```

3. Atualize os valores dos parâmetros `SESSION_TIMEOUT` e `IDTOKEN_LIFETIME`. O parâmetro `SESSION_TIMEOUT` configura a validade do token de acesso, cujo valor padrão é de `43200` segundos. O parâmetro `IDTOKEN_LIFETIME` configura a validade do token de ID, cujo valor padrão é de `12` horas. Especifique um número inteiro positivo seguido por uma unidade de tempo, que pode ser horas (h), minutos (m) ou segundos (s). Por exemplo, especifique 30 segundos como 30s. É possível incluir diversos valores em uma única entrada. Por exemplo, 1m30s é equivalente a 90 segundos.

   Depois de atualizar e salvar as mudanças, a seguinte mensagem será exibida:

   ```
   configmap "platform-auth-idp" editado
   ```

4. Reinicie o pod `auth-idp`.

   a. Obtenha o ID do pod `auth-idp`.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

   A seguir está uma saída de amostra do comando:

   ```
   auth-idp-t6sfm                              4/4     Running      0     1d
   ```

   b. Exclua o pod `auth-idp`.

   ```
   kubectl -n kube-system delete pod auth-idp-t6sfm
   ```

   Depois que o pod é excluído, a seguinte mensagem é exibida:

   ```
   pod "auth-idp-t6sfm" excluído
   ```

   Aguarde alguns minutos para que o pod seja reiniciado.

   c. Verifique o status do pod.

   ```
   kubectl -n kube-system get pods | grep auth-idp-t6sfm
   ```

   O pod obtém um novo ID após a reinicialização.

   ```
   auth-idp-5267t                                                 1/4       Running     0
   4m
   ```

Mudando o intervalo de tempo de atualização de

mapeamentos de função de segurança que é usado durante a autorização

Mude o intervalo de tempo de atualização de mapeamentos de função de segurança que é usado durante a autorização.

Por padrão, o intervalo de tempo de atualização dos mapeamentos de função de segurança é configurado como 10 minutos. É possível mudar esse valor padrão conforme necessário.

Siga estas etapas para mudar o intervalo de tempo:

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Edite o configmap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

3. Atualize o valor do parâmetro `PDP_REDIS_CACHE_DEFAULT_TTL`, conforme necessário. Especifique o valor em segundos. O valor padrão é `600`.

   Depois de atualizar e salvar as mudanças, a seguinte mensagem será exibida:

   ```
   configmap "platform-auth-idp" editado
   ```

4. Reinicie todos os pods `auth-idp` excluindo-os.

```
kubectl -n kube-system delete pod -l k8s-app=auth-pdp
```

Aguarde alguns minutos para que o pod seja reiniciado.

5. Verifique o status dos pods.

```
kubectl -n kube-system get pods | grep auth-pdp
```

O status de todos os pods deve ser mostrado como `2/2 Em execução`.

Alterando valores da variável do cache de procura LDAP

Altere os valores padrão das variáveis de cache de procura do Lightweight Directory Access Protocol (LDAP).

É possível mudar os valores das variáveis a seguir:

Tabela 1. Parâmetros do Cache LDAP

Variável	Valor Padrão	Descrição
LDAP_SEARCH_CACHE_TIMEOUT	1200 segundos	O tempo máximo que o conteúdo do cache dos resultados da procura fica disponível.
LDAP_SEARCH_CACHE_ENABLED	true	Um valor booleano para indicar que os resultados da procura devem ser armazenados em cache ou não.
LDAP_SEARCH_CACHE_SIZE	2000	O número de resultados de procura que estão armazenados no cache.
LDAP_SEARCH_CACHE_SIZELIMIT	2000	O número máximo de resultados que podem ser armazenados em cache para uma única procura LDAP.
LDAP_ATTR_CACHE_SIZE	2000	O número de entidades que podem ser armazenadas no cache.
LDAP_ATTR_CACHE_TIMEOUT	1200 segundos	O tempo máximo que o conteúdo do cache de atributo LDAP está disponível.
LDAP_ATTR_CACHE_SIZELIMIT	2000	O número máximo de atributos por entidade LDAP que são armazenados em cache.
LDAP_ATTR_CACHE_ENABLED	true	Um valor booleano para indicar que as entidades devem ser armazenadas em cache ou não.

Para obter mais informações, consulte Registro do usuário LDAP (ldapRegistry) .

Siga estas etapas para mudar os valores da variável:

1. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o configmap `platform-auth-idp` .

```
kubectl -n kube-system edit configmap platform-auth-idp
```

O conteúdo do arquivo é semelhante ao código a seguir:

```
# Please edit the following object. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
 AUDIT_DETAIL: "false"
 AUDIT_ENABLED_IDMGMT: "false"
 AUDIT_ENABLED_IDPROVIDER: "false"
 .
 .
 LDAP_ATTR_CACHE_ENABLED: "true"
 LDAP_ATTR_CACHE_SIZE: "2000"
 LDAP_ATTR_CACHE_SIZELIMIT: "2000"
 LDAP_ATTR_CACHE_TIMEOUT: 1200s
 LDAP_SEARCH_CACHE_ENABLED: "true"
 LDAP_SEARCH_CACHE_SIZE: "2000"
 LDAP_SEARCH_CACHE_SIZELIMIT: "2000"
 LDAP_SEARCH_CACHE_TIMEOUT: 1200s
 LDAP_SEARCH_CN_ATTR_ONLY: "false"
 LDAP_SEARCH_ID_ATTR_ONLY: "false"
```

```
  LDAP_SEARCH_SIZE_LIMIT: "50"
  LDAP_SEARCH_TIME_LIMIT: "5"
  MASTER_HOST: mycluster.icp
  NODE_ENV: production
  .
  .
"/tmp/kubectl-edit-i5ta2.yaml" 60L, 2204C
```

3. Altere os valores da variável LDAP, conforme necessário, e salve o ConfigMap.

   Depois de atualizar e salvar as mudanças, a seguinte mensagem será exibida:

   ```
   configmap "platform-auth-idp" editado
   ```

4. Reinicie o pod `auth-idp`.

   a. Obtenha o ID do pod `auth-idp`.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

   A seguir está uma saída de amostra do comando:

   ```
   auth-idp-5b78f                          4/4     Running                3          12d
   ```

   b. Exclua o pod `auth-idp`.

   ```
   kubectl -n kube-system delete pod auth-idp-5b78f
   ```

   Depois que o pod é excluído, a seguinte mensagem é exibida:

   ```
   pod "auth-idp-5b78f" excluído
   ```

   Aguarde alguns minutos para que o pod seja reiniciado.

   c. Verifique o status do pod.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

   O pod obtém um novo ID após a reinicialização.

   ```
   auth-idp-5267t                                                1/4       Running     0
   4m
   ```

5. Verifique se as variáveis LDAP estão atualizadas.

   a. Obtenha o ID do contêiner `platform-auth`.

   ```
   docker ps | grep platform-auth
   ```

   O seguinte é uma saída de amostra:

   ```
   2e5d416fe6d8        3ea9fbf2c199                          "/usr/bin/supervisor…"   12 days ago
   Up 12 days                              k8s_platform-auth-service_auth-idp-5b78f_kube-
   system_b83319c3-2484-11e9-8d6a-00000a29093b_1
   ```

   b. Efetue login no contêiner `platform-auth`.

   ```
   docker exec -it < container ID> bash
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   docker exec -it 2e5d416fe6d8 bash
   bash-4.4#
   ```

   c. Liste as variáveis LDAP.

   ```
   env|grep LDAP
   ```

   É possível ver o valor atualizado na saída.

   ```
   LDAP_ATTR_CACHE_SIZE=2500
   LDAP_ATTR_CACHE_ENABLED=true
   LDAP_ATTR_CACHE_TIMEOUT=1200s
   LDAP_ATTR_CACHE_SIZELIMIT=2000
   LDAP_SEARCH_CACHE_ENABLED=true
   ```

```
LDAP_SEARCH_CACHE_SIZE=2000
LDAP_SEARCH_CACHE_SIZELIMIT=2000
LDAP_SEARCH_CACHE_TIMEOUT=1200s
```

Configurando o Key Management Service

Configure o Key Management Service (KMS) para criptografar dados em repouso e em trânsito.

Use o KMS para provisionar e gerenciar chaves criptografadas para seus aplicativos e serviços. Deve-se configurar um dispositivo do Hardware Security Module (HSM) que é particionado para uso pelo KMS. O IBM® Cloud Private versão 3.2.0 suporta o SafeNet Luna Network HSM 6.2 e o nCipher nShield Connect HSM 12.40.2.

Configurando o SafeNet Luna Network HSM 6.2

Conclua estas etapas para configurar o dispositivo HSM:

1. Instale o pacote `key-management-hsm-amd64.tar.gz` do IBM Cloud Private 3.2.0 Key Management HSM seguindo as etapas em Instalando o software IBM na plataforma do IBM Cloud Private.

2. Configure a CLI do Helm. Para obter mais informações, consulte Instalando a CLI do Helm (helm).

3. Inclua o repositório `mgmt-charts` em seu cluster.

   ```
   helm repo add mgmt-charts https://<CLUSTER_NAME>.icp:8443/mgmt-repo/charts --ca-file
   ~/.helm/ca.pem --cert-file ~/.helm/cert.pem --key-file ~/.helm/key.pem
   ```

4. Atualizar repositórios Helm.

   ```
   helm repo update
   ```

5. Crie um arquivo `overrides_hsm.yaml` com o conteúdo a seguir:

   ```
   hsm:
     hsm_model: gemalto
     hsm_ip: <HSM IP ADDRESS>
     hsm_pw: <HSM PARTITION PASSWORD>
     server_ca: <BASE64 ENCODED CONTENTS OF LUNA SERVER CERTIFICATE>
     client_certs:
       <MANAGEMENT_NODE_IP>: <BASE64 ENCODED CONTENTS OF LUNA CLIENT CERTIFICATE>
       <MANAGEMENT_NODE_IP>-key: <BASE64 ENCODED CONTENTS OF LUNA CLIENT KEY>
   oss:
     image:
       repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-oss
       tag: <ICP_version>
   storage:
     image:
       repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-onboarding
       tag: <ICP_version>
   gemalto:
     image:
       repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-gemaltov6
       tag: <ICP_version>
   watcher:
     image:
       repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-oss-watcher
       tag: <ICP_version>
   ```

   Nota: para IBM Cloud Private with OpenShift, o repositório de todas as imagens deve iniciar com `docker-registry.default.svc:5000`. Por exemplo: `docker-registry.default.svc:5000/ibmcom/kms-oss-watcher`.

No parâmetro `client_certs`, deve-se listar o certificado e o par de chaves para todos os nós de gerenciamento. Todos os certificados e chaves devem ser codificados em base64.

Nota: como o arquivo `overrides_hsm.yaml` contém informações sensíveis, deve-se criptografá-lo ou excluí-lo quando não estiver em uso.

6. Execute o comando a seguir:

```
helm install mgmt-charts/key-management-hsm --tls --namespace kube-system --name kms-hsm -f
path/to/overrides_hsm.yaml
```

Configurando o nCipher nShield Connect HSM 12.40.2

Conclua estas etapas para configurar o dispositivo HSM:

1. Configure a CLI do Helm. Para obter mais informações, consulte Instalando a CLI do Helm (helm).

2. Inclua o repositório `mgmt-charts` em seu cluster.

```
helm repo add mgmt-charts https://<CLUSTER_NAME>.icp:8443/mgmt-repo/charts --ca-file
~/.helm/ca.pem --cert-file ~/.helm/cert.pem --key-file ~/.helm/key.pem
```

3. Crie um arquivo `overrides_hsm.yaml` com o conteúdo a seguir:

```
hsm:
  hsm_model: thales
  hsm_ip: <HSM IP ADDRESS>
  hsm_pw: <HSM PARTITION PASSWORD>
  rfs_ip: <RFS IP ADDRESS>
  slotId: <HSM slot ID string, i.e. "492971158">
oss:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-oss
    tag: <ICP_version>
storage:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-onboarding
    tag: <ICP_version>
thales:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-thalesv12
    tag: <ICP_version>
watcher:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-oss-watcher
    tag: <ICP_version>
```

Nota: para IBM Cloud Private with OpenShift, o repositório de todas as imagens deve iniciar com `docker-registry.default.svc:5000`. Por exemplo: `docker-registry.default.svc:5000/ibmcom/kms-oss-watcher`.

Nota: como o arquivo `overrides_hsm.yaml` contém informações sensíveis, deve-se criptografá-lo ou excluí-lo quando não estiver em uso.

4. Execute o comando a seguir:

```
helm install -f path/to/overrides_hsm.yaml mgmt-charts/key-management-hsm --tls --namespace
kube-system --name kms-hsm
```

Configurando o SoftHSM

O SoftHSM é um HSM virtualizado que exibe as funções do sistema de gerenciamento de chaves que está disponível. O SoftHSM pode ser usado apenas para propósitos de demonstração. É possível testar as APIs em um ambiente de não produção para entender o que pode ser possível com um módulo de segurança de hardware real.

Limitações

- Não é possível importar chaves raiz.
- Se um pod for reinicializado, o KMS se tornará inválido.

Conclua estas etapas para instalar o SoftHSM:

1. Configure a CLI do Helm. Para obter mais informações, consulte Instalando a CLI do Helm (helm).

2. Inclua o repositório `mgmt-charts` em seu cluster.

   ```
   helm repo add mgmt-charts https://<CLUSTER_NAME>.icp:8443/mgmt-repo/charts --ca-file
   ~/.helm/ca.pem --cert-file ~/.helm/cert.pem --key-file ~/.helm/key.pem
   ```

3. Atualizar repositórios Helm.

   ```
   helm repo update
   ```

4. Crie um arquivo `overrides_softhsm.yaml` com o conteúdo a seguir:

   ```
   hsm:
    hsm_model: softhsm
   oss:
    image:
      repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-oss
      tag: <ICP_version>
   storage:
    image:
      repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-onboarding
      tag: <ICP_version>
   softhsm:
    image:
      repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-softhsm
      tag: <ICP_version>
   watcher:
    image:
      repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-oss-watcher
      tag: <ICP_version>
   ```

   Nota: para IBM Cloud Private with OpenShift, o repositório de todas as imagens deve iniciar com `docker-registry.default.svc:5000`. Por exemplo: `docker-registry.default.svc:5000/ibmcom/kms-oss-watcher`.

5. Execute o comando a seguir:

   ```
   helm install mgmt-charts/key-management-hsm --tls --namespace kube-system --name kms-softhsm -f
   path/to/overrides_softhsm.yaml
   ```

Limpando dados do aplicativo HSM

Se você estiver alternando para um novo modelo HSM, o HSM Application Data poderá ser limpo, incluindo `clear_data: true` no arquivo de configuração `overrides.yaml` em `hsm`. Esse valor é, por padrão, configurado como false. Por exemplo:

```
hsm:
  hsm_model: <HSM MODEL>
  clear_data: true
  hsm_ip: <HSM IP ADDRESS>
  hsm_pw: <HSM PARTITION PASSWORD>
```

Gerando chaves

O KMS usa dois tipos de chaves.

- Chaves Root: chaves Root são chaves de agrupamento de chaves simétricas que você gerencia completamente. É possível usar uma chave raiz para proteger outras chaves criptográficas com criptografia avançada.
- Chaves padrão: chaves padrão são chaves simétricas que são usadas para criptografia. É possível usar uma chave padrão para criptografar e decriptografar dados diretamente.

É possível criar ou importar uma chave existente usando as APIs a seguir:

- Crie uma chave. Consulte Gerar uma chave
- Importe uma chave. Consulte Importar uma chave

Integrando um serviço para usar o KMS

Para usar o KMS para seu serviço, conclua as tarefas a seguir:

1. Configure a CLI do Kubernetes. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

2. Provisione a instância do KMS. Para obter mais informações, consulte Provisionando instâncias do KMS.
3. Crie um ID de serviço para seu serviço. É possível usar a console de gerenciamento ou as APIs para gerar um ID de serviço.
    - Para usar o console de gerenciamento, consulte Criando um ID de serviço usando o console da web do IBM Cloud Private.
    - Para usar a API, consulte Criar um ID de serviço.
4. Designe uma política de acesso para o ID de serviço para interagir com o serviço. É possível usar a console de gerenciamento ou as APIs para designar uma política de acesso.
    - Para usar o console de gerenciamento, consulte Criando um ID de serviço usando o console da web do IBM Cloud Private.
    - Para usar a API, consulte Criar uma política de acesso para um ID de serviço.

    A seguir está uma solicitação de amostra para designar uma política de acesso:

```
{
     "resources": [
       {
            "namespaceId": "kube-system",
            "serviceName": "kms",
            "serviceInstance": "<INSTANCE_ID>,
            "resource": "<KEY_ID>",
            "resourceType": "key"
       }
   ],
   "roles": [
       {
           "id": "crn:v1:icp:private:iam::::role:<ROLE>"
       }
   ]
}
```

Provisionando instâncias do KMS

O KMS suporta múltiplas instâncias. As instâncias no KMS permitem que os controles de acesso sejam designados a um grupo de chaves. Use instâncias para agrupamento lógico de chaves, por exemplo, por aplicativo ou por equipe.

Para provisionar uma instância, inclua anotação `ibm.com/kms.instanceID` em metadados de um segredo. É possível fornecer ao identificador da instância um nome conveniente para referência ao acessar os dados a partir do segredo posteriormente. Por exemplo:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: example-app-secret
  namespace: default
  annotations:
    ibm.com/kms.instanceID: "kms-instance"
```

A seção de dados é atualizada e um ID de instância provisionada é incluído no segredo com seu nome de referência selecionado. Os IDs de instância são criados no formato UUID, por exemplo, `ee8bf75d-aa46-4329-af41-9a2404d2b0eb`. O formato UUID é incluído em todas as solicitações de API para o KMS.

Instância padrão

Uma instância padrão é provisionada na implementação inicial do gráfico HSM do serviço de gerenciamento de chaves. O ID da instância é localizado no `key-management-secret` com o rótulo "kms-instance".

Criptografando segredos do Kubernetes com o plug-in Key Management Service

Criptografe segredos do Kubernetes usando o plug-in do Key Management Service (KMS).

O plug-in KMS é executado como um pod estático em um nó principal do Kubernetes. O servidor de API usa um mecanismo baseado em provedor para se comunicar com o plug-in, que, por sua vez, se comunica com o provedor KMS para obter o Data Encryption Key (DEK). O DEK é usado para criptografar os segredos ou decriptografar os segredos criptografados. Para obter mais informações, consulte Usando um provedor KMS para a criptografia de dados.

No IBM Cloud Private, o plug-in é compatível com o Federal Information Processing Standards (FIPS) 140-2. Ele usa o Advanced Encryption Standard (AES), um algoritmo criptográfico aprovado pelo FIPS, com o modo Cipher Blocker Chaining (CBC) para criptografar segredos do Kubernetes. O KMS não gera nenhuma chave nem faz cálculos criptográficos. Ele delega essas tarefas a um Hardware Security Module (HSM), que é compatível com o FIPS.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Pré-requisitos

- Deve-se fornecer um dispositivo HSM que é particionado para uso pelo KMS. O IBM Cloud Private Versão 3.2.0 suporta o SafeNet Luna Network HSM 6.2.
- Deve-se instalar o gráfico HSM do KMS.
- Deve-se obter o ID da instância e o ID do Customer Root Key (CRK) seguindo as etapas que estão documentadas na seção do Key Management Service.

Ativando o Plug-in do KMS

1. Efetue logon no nó principal como um administrador de cluster.
2. Atualize os parâmetros `API_KEY`, `INSTANCE_ID` e `CRK_ID` no arquivo `/etc/cfc/conf/kmsplugin-config.yaml`.
3. Reinicie o plug-in do KMS excluindo o contêiner de plug-in do KMS existente. Para obter informações adicionais sobre como instalar o `kubectl`, consulte Instalando a CLI do Kubernetes (kubectl).

   ```
   kubectl kill <KMS plug-in container ID>
   ```

4. Assegure-se de que o arquivo `/etc/cfc/conf/encryption-config.yaml` exista em todos os nós principais. O provedor de plug-in do KMS deve ser o primeiro provedor na lista de provedores. O conteúdo é semelhante ao código a seguir:

   ```
   kind: EncryptionConfiguration
   apiVersion: apiserver.config.k8s.io/v1
   resources:
     - resources:
         - secrets
       providers:
         - kms:
             name: KmsPlugin
             endpoint: unix:///var/run/icp/keyprotectprovider.sock
             cachesize: 100
         - identity: {}
   ```

5. Copie o arquivo `/etc/cfc/pods/master.json` para outro local, como a pasta `/root`.

   ```
   cp /etc/cfc/pods/master.json /root/
   ```

6. Abra o arquivo `/root/master.json` para edição. Inclua a sinalização `encryption-provider-config` na lista de comandos do `apiserver` no arquivo `master.json`. A sinalização deve apontar para o arquivo `/etc/cfc/conf/encryption-config.yaml`. A seguir está um arquivo `master.json` de amostra:

   ```
   {
           "name": "apiserver",
         "image": "mycluster.icp:8500/ibmcom/hyperkube:v1.11.1-ee",
         "imagePullPolicy": "IfNotPresent",
         "command": [
             "/hyperkube",
             "apiserver",
             "--encryption-provider-config=/etc/cfc/conf/encryption-config.yaml",
             "--secure-port=8001",
             "--bind-address=0.0.0.0",
             "--advertise-address=9.42.78.47",
             "--endpoint-reconciler-type=lease",
             "--insecure-port=0",
             "--etcd-servers=https://9.42.78.47:4001",
             "--etcd-cafile=/etc/cfc/conf/etcd/ca.pem",
   .
   ```

.
.

7. Copie o arquivo `/root/master.json` atualizado novamente para o local `/etc/cfc/pods/`. Esta ação reinicia o pod principal.

   ```
   cp /root/master.json /etc/cfc/pods/master.json
   ```

Verificando se o plug-in KMS está ativado

Verifique o log do contêiner de plug-in do KMS. O log deve ter mensagens que indicam que o plug-in foi iniciado com sucesso. A seguir está um log de amostra:

```
{"level":"info","ts":"Oct 29 22:55:01.615","msg":"signals.go:40: SIGTERM signal handler started.
Observando os sinais SIGTERM. " }
{"level":"info","ts":"Oct 29 22:55:01.847","msg":"server.go:140: The Provider server is started."}
```

Verificando se os segredos são criptografados usando o plug-in KMS

Os segredos são armazenados no etcd. É possível usar etcdctl para visualizar os segredos. Para obter mais informações sobre o etcdctl, consulte https://github.com/etcd-io/etcd/tree/master/etcdctl.

Considere o exemplo em que você cria um segredo que é denominado `secret1` no namespace `default`. Quando o plug-in KMS é ativado, todos os novos segredos são criptografados com o plug-in KMS.

1. Efetue logon no pod `etcd` .

2. Inclua um alias etcdctl3. O IBM Cloud Private usa a API etcdctl versão 3.

   ```
   / # alias etcdctl3="ETCDCTL_API=3 etcdctl --endpoints https://<master node IP address>:4001 --
   cacert /etc/cfc/conf/etcd/ca.pem --cert /etc/cfc/conf/etcd/client.pem --key
   /etc/cfc/conf/etcd/client-key.pem"
   ```

3. Execute o comando `etcdctl3` a seguir para recuperar o segredo. Direcione a saída para o dump hex.

   ```
   / # etcdctl3 get /registry/secrets/default/secret1 |hexdump -C
   ```

   O seguinte é uma saída de amostra:

   ```
   00000000  2f 72 65 67 69 73 74 72  79 2f 73 65 63 72 65 74  |/registry/secret|
   00000010  73 2f 64 65 66 61 75 6c  74 2f 73 65 63 72 65 74  |s/default/secret|
   00000020  31 0a 6b 38 73 3a 65 6e  63 3a 6b 6d 73 3a 76 31  |1.k8s:enc:kms:v1|
   00000030  3a 4b 6d 73 50 6c 75 67  69 6e 3a 02 74 50 50 2b  |:KmsPlugin:.tPP+|
   00000040  42 41 77 45 42 42 6c 4e  6c 59 33 4a 6c 64 41 48  |BAwEBBlNlY3JldAH|
   00000050  2f 67 67 41 42 41 77 45  4a 53 33 42 57 5a 58 4a  |/ggABAwEJS3BWZXJ|
   00000060  7a 61 57 39 75 41 51 51  41 41 51 70 58 63 6d 46  |zaW9uAQQAAQpXcmF|
   00000070  77 63 47 56 6b 52 45 56  4c 41 51 6f 41 41 51 5a  |wcGVkREVLAQoAAQZ|
   00000080  44 61 58 42 6f 5a 58 49  42 44 41 41 41 41 50 34  |DaXBoZXIBDAAAAP4|
   00000090  42 6c 2f 2b 43 41 51 49  42 2f 67 45 30 5a 58 6c  |Bl/+CAQIB/gE0ZXl|
   000000a0  4b 61 6d 46 59 51 6d 39  61 57 45 6f 77 57 6c 68  |KamFYQm9aWEowWlh|
   000000b0  6f 4d 45 6c 71 62 32 6c  53 56 6b 5a 77 57 56 56  |oMElqb2lSVkZwWVV|
   000000c0  77 56 6c 64 74 65 48 42  4e 4d 32 52 78 54 31 68  |wVldteHBNM2RxT1h|
   000000d0  4b 62 46 6c 71 57 58 70  61 52 7a 52 33 55 57 78  |KbFlqWXpaRzR3UWx|
   000000e0  76 65 6d 4a 55 5a 46 4a  56 56 6b 70 51 57 6c 68  |vemJUZFJVVkpQWlh|
   000000f0  73 65 6d 46 58 5a 48 68  61 53 46 6b 30 56 44 4a  |semFXZHhaSFk0VDJ|
   00000100  30 52 55 78 36 55 6c 4e  68 52 56 70 68 59 6d 35  |0RUx6UlNhRVphYm5|
   00000110  4f 56 6c 52 45 61 45 64  4d 4d 47 78 43 56 57 30  |OVlREaEdMMGxCVW0|
   00000120  35 54 6c 4a 75 62 46 6c  4e 51 30 6c 7a 53 57 31  |5TlJubFlNQ0lzSW1|
   00000130  6f 61 47 4d 79 5a 32 6c  50 61 55 6c 35 56 57 74  |oaGMyZ2lPaUl5VWt|
   00000140  30 52 57 56 58 64 7a 46  4f 4d 55 70 57 54 54 46  |0RWVXdzFOMUpWTTF|
   00000150  53 55 6c 45 7a 57 6d 39  5a 62 56 4a 73 5a 46 64  |SUlEzWm9ZbVJsZFd|
   00000160  4a 64 6b 31 55 55 6a 5a  4d 4d 30 35 70 56 30 5a  |Jdk1UUjZMM05pV0Z|
   00000170  73 59 55 31 36 5a 46 5a  5a 56 56 4a 4a 56 57 31  |sYU16ZFZZVVJJVW1|
   00000180  4f 56 30 35 75 53 6c 56  68 61 7a 68 79 5a 44 42  |OV05uSlVhazhyZDB|
   00000190  47 57 46 5a 47 53 6e 46  58 56 6e 42 50 59 7a 4a  |GWFZGSnFXVnBPYzJ|
   000001a0  77 4d 56 5a 74 61 7a 56  53 4d 56 5a 68 5a 46 5a  |wMVZtazVSMVZhZFZ|
   000001b0  73 65 55 31 71 54 6e 5a  6b 57 46 59 30 55 6b 55  |seU1qTnZkWFY0UkU|
   000001c0  78 59 56 56 73 51 6d 78  4c 4d 55 5a 47 55 30 52  |xYVVsQmxLMUZGU0R|
   000001d0  43 63 56 46 55 4d 44 6c  4a 61 58 64 70 59 56 68  |CcVFUMDlJaXdpYVh|
   000001e0  5a 61 55 39 70 53 6b 4a  57 4d 6c 70 7a 5a 57 74  |ZaU9pSkJWMlpzZWt|
   000001f0  47 4d 6c 46 72 64 47 78  53 61 32 52 33 5a 47 70  |GMlFrdGxSa2R3ZGp|
   00000200  4f 61 56 6c 74 53 6c 4a  4e 62 55 35 43 55 46 51  |OaVltSlJNbU5CUFQ|
   ```

```
00000210  77 61 55 78 44 53 6a 4a  61 57 45 70 36 59 56 63  |waUxDSjJaWEp6YVc|
00000220  35 64 55 6c 71 62 32 6c  4e 61 54 52 33 54 47 70  |5dUlqb2lNaTR3TGp|
00000230  42 61 57 5a 52 50 54 30  42 57 45 74 75 56 48 64  |BaWZRPT0BWEtuVHd|
00000240  4c 59 6a 52 36 65 6e 55  77 4d 57 77 79 5a 54 42  |LYjR6enUwMWwyZTB|
00000250  6d 4d 45 31 79 57 46 6c  77 54 6e 64 72 4e 55 68  |mME1yWFlwTndrNUh|
00000260  36 56 57 74 42 4d 55 6c  4c 4e 30 68 33 57 54 52  |6VWtBMUlLN0h3WTR|
00000270  59 55 46 46 44 57 6d 35  53 4e 6c 4e 56 4e 57 52  |YUFFDWm5SNlNVNWR|
00000280  6a 65 6c 5a 31 54 46 56  57 4f 48 4a 6d 52 30 6c  |jelZ1TFVWOHJmR0l|
00000290  31 55 30 52 6e 65 45 64  4c 4e 6d 5a 5a 62 45 74  |1U0RneEdLNmZZbEt|
000002a0  6b 54 33 6c 4e 5a 58 46  47 5a 33 52 6e 50 54 30  |kT3lNZXFGZ3RnPT0|
000002b0  41 e4 52 d5 5e 7f f2 f4  97 58 79 e5 3b 26 3a 97  |A.R.^....Xy.;&:.|
000002c0  b0 bc 3b e8 1d 92 82 97  aa e5 89 5d ba 98 51 9a  |..;........]..Q. |
000002d0  69 42 16 3b e7 56 7f 32  1b 89 a3 a4 7e 3e 03 7c  |iB.;.V.2....~>.||
000002e0  59 7f b5 56 ca 2c 0d 84  66 ee c0 af d8 bd 24 e2  |Y..V.,..f ..... $. |
000002f0  fc 1e b6 f6 0b a8 8b c1  b6 ee 98 45 93 85 34 2b  |...........E..4+|
00000300  1d 67 29 c5 4e dd e4 4f  92 59 29 cc 5d d7 6c c7  |.g).N..O.Y).].l.|
00000310  24 ac 97 f2 36 36 47 30  f6 5f a1 4c 9e 99 13 46  |$...66G0._.L...F|
00000320  c9 23 75 71 5f de b4 6e  5b 96 c8 44 f1 2b 4e 3e  |.#uq_..n[..D.+N>|
00000330  80 48 cd eb 08 66 50 4d  86 a0 67 66 45 19 5f af  |.H...fPM..gfE._.|
00000340  9f 0a                                            |..|
```

Para criptografar um segredo que foi incluído antes de ativar o plug-in KMS, use o comando de atualização a seguir. Deve-se configurar a CLI kubectl para executar este comando. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

```
kubectl get secret <secret name> -n <name space> -o json |kubectl replace -f -
```

Parâmetros de configuração de plug-in do KMS

Dois arquivos são usados para configurar a criptografia de plug-in do KMS. O `/etc/cfc/conf/encryption-config.yaml` é usado pelo servidor de API kubernetes e o `/etc/cfc/conf/kmsplugin-config.yaml` é usado pelo plug-in KMS.

Parâmetros no arquivo encryption-config.yaml

- `name`: Nome do plug-in KMS.
- `endpoint`: O terminal do soquete UNIX. Esse valor deve corresponder ao especificado para o parâmetro `UNIX_SOCKET_PATH` no arquivo `kmsplugin-config.yaml`.
- `cachesize`: o número de segredos armazenados em cache pelo servidor da API.

Parâmetros no arquivo kmsplugin-config.yaml

seção SERVER_CONFIG

- `REGION`: Nome da região.
- `NUM_LEN_BYTES`: o número de bytes para especificar o comprimento de DEK. Não mude esse valor antes de consultar a equipe de suporte IBM.
- `CACHE_TIMEOUT_IN_HOURS`: especifique depois de quanto tempo o cache atinge o tempo limite. Especifique um valor inteiro. Depois que o cache atinge o tempo limite, todas as entradas do DEK agrupadas no cache são limpas. Especifique o número zero se você não deseja armazenar em cache os DEKs.
- `RESTART_DELAY_IN_SECONDS`: tempo de atraso antes que o serviço de plug-in do KMS seja reiniciado. Especifique um valor inteiro.
- `UNIX_SOCKET_PATH`: O caminho do soquete UNIX gRPC. Esse caminho deve corresponder ao que você especificou no arquivo `encryption-config.yaml`. Não mude esse valor antes de consultar a equipe de suporte IBM.
- `TOKEN_URL`: a URL do serviço de token do IAM. Não altere esse valor.
- `TOKEN_URL_CA_FILE`: o arquivo de certificado TLS que é usado para comunicação segura com o serviço de token do IAM. Não altere esse valor.
- `API_KEY`: a chave API que é usada para obter o token de acesso. Especifique a chave API que você usa.
- `HEALTHZ_PATH`: caminho que é usado para construir a URL de funcionamento. Não altere esse valor.
- `HEALTHZ_PORT`: caminho que é usado para construir a URL de funcionamento. O valor padrão é 20358. Esse valor deve corresponder ao valor de porta que é configurado nas seções `livenessProbe` e `readinessProble` do arquivo `kmsplugin.json`.
- `NEW_DEK`: Especifique um valor booleano. Um valor igual a `true` indica que um novo DEK é usado para criptografar cada segredo. Um valor de `false` indica que um DEK é reutilizado para criptografar cada segredo.
- `MAX_RETRIES`: o número máximo de novas tentativas de conexão, caso o plug-in do KMS não possa se conectar com o serviço do KMS.

Seção KP_CONFIG:

- `VERSION`: o valor padrão é 1. Não mude esse valor.
- `CRK_ID`: o ID do Customer Root Key. Inclua o ID de CRK que você usa.
- `URL`: a URL para o serviço da API do KMS. Não altere esse valor.
- `KP_CA_FILE`: o certificado TLS que é usado para comunicação segura com o serviço da API KMS. Não altere esse valor.
- `INSTANCE_ID`: o ID da instância do HSM. Inclua o ID do HSM que você usa.

Seção LOGGER:

- `LOG_LEVEL`: configure o nível de log em `info` ou em `debug`.
- `LOG_ALL_VALUES`: Especifique um valor booleano. Um valor igual a `true` indica que todos os valores são registrados. Esses logs podem incluir informações sensíveis. Se você não deseja incluir informações confidenciais nos logs, configure o valor como `false`.

Guia de adoção do Key Management Service (KMS)

O KMS no IBM Cloud Private ajuda a manter os dados seguros. Ele se integra aos módulos de segurança de hardware (HSM) pertencentes ao usuário. Uma chave raiz é usada para criptografia de envelope para proteger as chaves de criptografia de dados usadas dentro de seus aplicativos.

A incorporação do KMS em seus aplicativos inclui as tarefas a seguir:

- Provisionar uma instância do KMS
- Configurar um ID de serviço com privilégios de `Administrator` para a instância
- Configurar um ID de serviço com privilégios de `Viewer` para seu aplicativo
- Gerar ou importar uma chave raiz no KMS
- Use a chave raiz para criptografar e decriptografar as chaves de criptografia de dados que são usadas em seu aplicativo.

Introdução

Uma configuração inicial é necessária para que tudo esteja pronto para que seu aplicativo se integre ao KMS.

- Configurando uma instância
- Incluindo chaves raiz na instância

Configurando uma instância

As instâncias no KMS permitem que os controles de acesso sejam designados a um grupo de chaves. Uma instância pode ser usada para qualquer agrupamento lógico de chaves, como por aplicativo ou por equipe.

Provisionando uma instância

Para provisionar uma instância, inclua a anotação `ibm.com/kms.instanceID` nos metadados de um segredo. É possível fornecer ao identificador da instância um nome conveniente para referência ao acessar os dados a partir do segredo posteriormente.

Por exemplo, para provisionar uma instância com os dados armazenados na referência `kms-instance`, forneça um segredo com:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: example-app-secret
  namespace: default
  annotations:
    ibm.com/kms.instanceID: "kms-instance"
```

A seção `data` é atualizada e um ID de instância provisionada é incluído no segredo com o nome de referência selecionado. O ID da instância está no formato UUID (por exemplo, `ee8bf75d-aa46-4329-af41-9a2404d2b0eb`). Ele é incluído em todas as solicitações de API para o KMS.

Configurando IDs de Serviço

Pelo menos dois IDs de serviço devem ser configurados. Um ID de serviço é designado a uma função administrativa para uma instância do KMS. O outro ID é designado a uma função de visualizador a ser usada pelo aplicativo ou serviço.

Para obter mais informações, consulte Criar um ID de serviço.

Para cada ID de serviço, uma política de serviço deve ser incluída pelo administrador de cluster. Inclua o ID da instância provisionada na política. Para que o ID do serviço do visualizador seja usado pelo serviço, deve-se incluir a chave raiz na instância. Para obter mais informações, consulte Incluindo chaves raiz na instância.

O corpo da política é semelhante ao código a seguir. Para o administrador da instância, não forneça os parâmetros `resource` ou `resourcetype`.

```
{
     "resources": [
          {
           "namespaceId": "kube-system",
              "serviceName": "kms",
              "serviceInstance": "<INSTANCE_ID>,
              "resource": "<KEY_ID>",
              "resourceType": "key"
          }
   ],
      "roles": [
          {
           "id": "crn:v1:icp:private:iam::::role:<ROLE>"
          }
   ]
}
```

Incluindo chaves raiz em instâncias

As chaves raiz são armazenadas e gerenciadas no KMS. As chaves usam criptografia de envelope para proteger as chaves de criptografia de dados que são usadas pelos seus aplicativos.

Gerando chaves raiz

O KMS usa a chamada da API a seguir para gerar uma chave raiz:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -H 'correlation-id: <correlation_ID>' \
   -d '{
  "metadata": {
    "collectionType": "application/vnd.ibm.kms.key+json", "collectionTotal": 1
  }, "resources": [ {
    "type": "application/vnd.ibm.kms.key+json",
    "name": "<key_alias>",
    "description": "<key_description>",
    "expirationDate": "<YYYY-MM-DDTHH:MM:SS.SSZ>",
    "extractable": <key_type>
    }
  ]
}'
```

Para obter mais informações, consulte Gerar uma chave.

Importando chaves raiz (Bring Your Own Key)

Talvez você já tem uma chave ou prefere gerar a chave sozinho. É possível usar a chamada da API a seguir para importar seu material de chave para o KMS:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
```

```
   -H 'correlation-id: <correlation_ID>' \
   -d '{
  "metadata": {
    "collectionType": "application/vnd.ibm.kms.key+json", "collectionTotal": 1
  }, "resources": [ {
    "type": "application/vnd.ibm.kms.key+json",
    "name": "<key_alias>",
    "description": "<key_description>",
    "expirationDate": "<YYYY-MM-DDTHH:MM:SS.SSZ>",
    "payload": "<key_material>",
    "extractable": <key_type>
    }
  ]
}'
```

Para obter mais informações, consulte Importar uma chave.

Incorporando chaves raiz em seus aplicativos

Com a sua instância provisionada e sua chave raiz incluída, você está pronto para aproveitar o KMS em seu aplicativo.

Agrupando Data Encryption Keys (DEK)

Em vez de armazenar a chave de criptografia de dados usada em seu aplicativo dentro do próprio aplicativo, deve-se criptografar o DEK com sua chave raiz. O DEK agrupado deve ser armazenado em seu aplicativo. Por exemplo:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID}?action=wrap \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.kms.key_action+json' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -d '{
     'plaintext': '<data_key>',
     'aad': ['<additional_data>', '<additional_data>']
   }'
```

Para obter mais informações, consulte Agrupar uma chave.

Desagrupando Data Encryption Keys (DEK)

Seu aplicativo usa a API a seguir para desagrupar (decriptografar) a chave de criptografia de dados agrupada quando ela for necessária.

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID}?action=unwrap \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.kms.key_action+json' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -d '{
     'ciphertext': '<data_key>',
     'aad': ['<additional_data>', '<additional_data>']
   }'
```

Para obter mais informações, consulte Desagrupar uma chave.

Gerando Data Encryption Keys (DEK)

Se você preferir que o KMS gere seu DEK, siga as instruções para agrupar uma chave. Omita a carga útil.

Gerenciando o ciclo de vida de suas chaves

O KMS pode ser usado para gerenciar o ciclo de vida de suas chaves raiz.

Expiração

Você pode ter incluído uma data de expiração quando a chave raiz é incluída no KMS. Essa chave é inutilizável para criptografia quando a data de expiração chega. Deve-se incluir uma nova chave para que seu aplicativo continue o uso do KMS. Deve-se

também atualizar a configuração do aplicativo para usar o novo ID de chave. A chave expirada ainda pode ser usada para decriptografar os dados existentes.

Giro

A rotação de chave permite que o material de chave raiz seja substituído, enquanto mantém o mesmo ID de chave. Essa solução é perfeita se você acreditar que sua chave pode estar comprometida. Também é uma boa prática trocar materiais de chave periodicamente.

Uma chave raiz gerada deve ser girada com uma chave raiz gerada recentemente. De forma semelhante, uma chave raiz deve ser fornecida para girar uma chave que foi importada anteriormente. Ainda é possível usar o material de chave antigo para decriptografar os dados existentes. Use a chamada da API a seguir para girar uma chave:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID}?action=rotate \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.kms.key_action+json' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -d '{
     'payload': '< data_key>'
   }'
```

Para obter mais informações, consulte Girar uma chave.

Vulnerability Advisor

Use o consultor para obter o status de segurança para imagens de contêiner em seu registro privado do IBM® Cloud Private. O Vulnerability Advisor também executa verificações de segurança em contêineres em execução em seu ambiente.

Para obter informações adicionais sobre o Vulnerability Advisor, consulte a seção *Sobre o Vulnerability Advisor* nos Documentos do IBM Cloud.

O recurso Vulnerability Advisor é suportado para clusters de múltiplos nós somente das edições Cloud Native e Enterprise do IBM Cloud Private.

Visualize a tabela a seguir para obter uma lista de sistemas operacionais que o Vulnerability Advisor suporta:

Tabela 1. Sistemas operacionais que o Vulnerability Advisor suporta.

Sistema Operacional	Versão
Ubuntu	• 18.04 • 17.10 • 16.10 • 16.04 • 15.10 • 15.04 • 14.04 • 12.04 • 13.10 • 13.04 • 12,10 • 11.10 • 11.04 • 10.10 • 10.04
Alpine	2.7-3.8
Red Hat Enterprise Linux	todas as imagens base
Centos	todas as imagens base
Debian	• 7 • 8 • 9

Para obter uma lista dos componentes do Vulnerability Advisor, consulte Componentes.

Ative o Vulnerability Advisor durante ou após a instalação do cluster do IBM Cloud Private. Para obter informações adicionais, consulte Ativando o Vulnerability Advisor.

Para ativar a pós instalação do Orientador de vulnerabilidade do cluster, conclua as etapas nas seções a seguir:

Configurando o Vulnerability Advisor

Configurando o crawler do contêiner do Vulnerability Advisor

1. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
2. Na caixa de procura digite "live-crawler".
3. Para o ConfigMap `vulnerability-advisor-live-crawler`, selecione **Ação** > **Editar**. O arquivo JSON `vulnerability-advisor-live-crawler` é exibido.
4. Modifique o valor do parâmetro `enabled`.
   - Para desativar o crawler, configure o parâmetro `enabled` como `false`.
   - Para ativar o crawler, configure o parâmetro `enabled` como `true`.
5. (Opcional) Também é possível configurar o intervalo de tempo para varrer os contêineres no host. Para configurar o intervalo de tempo, modifique o valor do parâmetro `crawl-interval`. O valor padrão é 86400 (segundos por dia).
6. Clique em **Enviar**.
7. Deve-se reiniciar o contêiner do crawler. O crawler do contêiner é implementado como um DaemonSets denominado `vulnerability-advisor-live-crawler`. Reinicie o contêiner do crawler executando o seguinte comando:

```
 kubectl delete pods -n kube-system $(kubectl get pods -n kube-system |  awk '{print $1}' |
grep live-crawler)
```

Configurando o crawler de imagem do Vulnerability Advisor

1. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
2. Na caixa de procura digite "registry-crawler".
3. Para o ConfigMap `vulnerability-advisor-registry-crawler`, selecione **Ação** > **Editar**. O arquivo JSON `vulnerability-advisor-registry-crawler` é exibido.
4. Modifique o valor do parâmetro `enabled`.
   - Para desativar o crawler, configure o parâmetro `enabled` como `false`.
   - Para ativar o crawler, configure o parâmetro `enabled` como `true`.
5. Clique em **Enviar**.

Configurando o crawler de imagem do Vulnerability Advisor para varrer novamente as imagens

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Na caixa de procura digite "registry-crawler".
3. Para a implementação `vulnerability-advisor-registry-crawler`, selecione **Ação** > **Editar**. O arquivo JSON `vulnerability-advisor-registry-crawler` é exibido.
4. Modifique o valor dos parâmetros a seguir.
   - Para varrer novamente as imagens que foram varridas com êxito, configure a opção `RESET_WHITELIST` para `true`.
   - Para varrer novamente as imagens que falharam ao serem varridas, configure a opção `RESET_BLACKLIST` para `true`.
5. Clique em **Enviar**.

Configurando o número de linhas para visualizações de lista de contêineres e imagens

1. No menu de navegação, clique em **Tools** > **Consultor de vulnerabilidade**.
2. Selecione um namespace da tabela. A janela **Vulnerability Advisor (Listar Contêineres)** é exibida. Cada linha na tabela inclui um relatório para cada contêiner. Há 50 linhas que são exibidas por página com um máximo de 100 linhas.
3. Para configurar o número de linhas, inclua o parâmetro `max` na URL da página. Por exemplo, quando você inclui o parâmetro `&max=200` na URL, um máximo de 200 linhas no total é exibido.
4. Para aumentar o número de linhas exibidas em cada página, inclua o parâmetro `count` na URL da página. Por exemplo, quando você inclui o parâmetro `&count=100` na URL, cada página inclui um máximo de 100 linhas.
5. É possível configurar os parâmetros `max` e `count`. Por exemplo, quando você inclui `&max=300 &count=100` na URL, cada página exibe um máximo de 100 linhas e um máximo de 300 linhas (máximo de 3 páginas) no total.

   ```
   https://xxx.xxx.xxx.xxx:8443/va/ui/list?access_group=kube-system&max=300&count=100
   ```

   Os parâmetros de URL `max` e `count` são ativados para as tarefas a seguir:

   - Vulnerability Advisor (Listar Contêineres)
   - Vulnerability Advisor (Listar Imagens)
   - Orientador de Mutação (Listar Contêineres)

Gerenciamento de logs e relatório

Os componentes Vulnerability Advisor, o log do Kafka e os dados do Minio consomem uma grande quantidade de espaço em disco em nós do VA. Por padrão, o Kafka retém 600 minutos (10 horas) de logs e o Minio retém 30 dias de dados. Esses dados incluem relatórios do contêiner.

Configurando o intervalo de curadoria de dados do limpador VA Minio

1. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
2. Para o ConfigMap `vulnerability-advisor-minio-cleaner-config`, selecione **Ação** > **Editar**. O arquivo JSON `vulnerability-advisor-minio-limp-config` é exibido.
3. Modifique o valor de cada depósito Minio `vacos:30 vacos-hf:5 vacos-ma:30 vacos-summary:30` na seção `data.clean.sh`. A unidade é dias.
4. Clique em **Enviar**.

Orientador de Mutação

É possível visualizar os alertas de modificação de arquivos do sistema, arquivos de configuração, arquivos de conteúdo ou o processo do sistema operacional. No menu de navegação, clique em **Ferramentas** > **Vulnerability Advisor** > **namespaces**. Selecione o botão **Acessar o Mutation Advisor** para visualizar alertas.

Configurando o crawler de processo do Mutation Advisor

1. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
2. Na caixa de procura, digite "ma-crawler".
3. Para o ConfigMap `vulnerability-advisor-process-ma-crawler`, selecione **Ação** > **Editar**. O arquivo JSON `vulnerability-advisor-process-ma-crawler` é exibido.
4. Modifique o valor do parâmetro `enabled`.
   - Para desativar o crawler, configure o parâmetro `enabled` como `false`.
   - Para ativar o crawler, configure o parâmetro `enabled` como `true`.
5. (Opcional) Também é possível configurar o intervalo de tempo para varrer os contêineres no host. Para configurar o intervalo de tempo, modifique o valor do parâmetro `crawl-interval`. O valor padrão é 300 (segundos por 5 minutos).
6. Clique em **Enviar**.
7. Deve-se reiniciar o contêiner do crawler. O crawler do contêiner é implementado como um DaemonSets denominado `vulnerability-advisor-process-ma-crawler`. Reinicie o contêiner do crawler executando o seguinte comando:

   ```
    kubectl delete pods -n kube-system $(kubectl get pods -n kube-system |  awk '{print $1}' |
   grep vulnerability-advisor-process-ma-crawler)
   ```

Configurando o crawler de arquivo do Mutation Advisor

A Mutação de Arquivo também é implementada pelo crawler de contêiner do Vulnerability Advisor. Para obter informações, consulte Configurando o crawler de contêiner do Vulnerability Advisor.

Configurando a lista de desbloqueio do Orientador de Mu

O Mutation Advisor suporta a configuração de listas de desbloqueio de mutações de arquivos e de processos comuns para redução de alarmes falsos. O sistema gera listas de desbloqueio de candidatos que podem ser ativadas ou desativadas na console de gerenciamento.

Conclua as etapas a seguir para configurar as listas de desbloqueio do Mutation Advisor.

1. Na página do Mutation Advisor, clique em **Gerenciar lista de desbloqueio**.
2. Decida se você deseja uma lista de desbloqueio para mutação de arquivo ou para mutação de processo. Na seção **Escopo**, selecione **Arquivo** ou **Processo** no menu suspenso.
3. Associe contêineres a uma lista de desbloqueio de uma das maneiras a seguir:
   - Selecione **Todos os contêineres no namespace atual**.
   - Selecione **Somente contêineres criados usando a imagem abaixo** e forneça o nome da imagem completo conforme definido no `.yaml spec` do Kubernetes.
   - Selecione uma imagem na lista suspensa para usar uma imagem com uma lista de desbloqueio existente.
4. Atualize as regras. Na seção **Regras**, clique em **Nova regra**. Na janela pop-up, insira um nome **Padrão** e selecione uma **Ação** para as regras de correspondência. Clique em **Criar**.
5. Sua nova regra é incluída na tabela **Regras**.
6. Ative ou desative uma regra alternando os botões de opções **ON/OFF**.
7. É possível remover uma regra. Na coluna Excluir, marque a caixa associada à regra que você deseja remover.
8. Clique em **Salvar Lista de Desbloqueio** para salvar as configurações.

Configurando o intervalo de limpeza de log do cluster Kafka

1. Configure a CLI do `kubectl`. Consulte Acessando o cluster do IBM Cloud Private usando a CLI kubectl.

2. Edite o objeto StatefulSet `vulnerability-advisor-kafka` para reconfigurar o Kafka.

   ```
   Kubectl -- namespace = kube-system edit StatefulSet vulnerabilidade a consultor kafka
   ```

3. Modifique o valor da variável de ambiente `KAFKA_LOG_RETENTION_MINUTES`. O valor padrão é 600 minutos (10 horas).

4. Salve as alterações.

Visualizando relatórios de segurança

Na console de gerenciamento, é possível visualizar relatórios de segurança para contêineres e imagens organizados por namespace. Esses relatórios de segurança são gerados usando uma política padrão.

1. No menu de navegação, clique em **Tools** > **Consultor de vulnerabilidade**.
2. Selecione o namespace que você deseja visualizar. O painel do Consultor de vulnerabilidade é exibido. Nesse painel, é possível revisar os relatórios para contêineres e imagens no namespace selecionado. O relatório detalha as informações a seguir em cada contêiner ou imagem:
   - Nome - nome do contêiner ou imagem
   - Proprietário - o namespace ao qual a imagem ou contêiner pertence.
   - Varredura mais recente - o registro de data e hora quando a imagem ou o contêiner foi varrido.
   - Tipo - especifica se o objeto é um contêiner ou imagem
   - Políticas Organizacionais - a política de segurança que está sendo usada. Isso é configurado na página Gerenciando políticas.
   - Pacotes Vulneráveis - vulnerabilidades atuais que são identificadas para o contêiner ou imagem.
   - Configurações do Contêiner - resumo de problemas de segurança e conformidade em potencial. As recomendações para segurança também são apresentadas aqui.

Gerenciando políticas

1. No menu de navegação, clique em **Tools** > **Consultor de vulnerabilidade**.
2. Selecione o namespace para o qual você deseja visualizar relatórios. O painel do Vulnerability Advisor é exibido.
3. No menu de navegação horizontal do painel do Vulnerability Advisor, selecione **Gerenciar políticas**.
4. Na página Gerenciar políticas, selecione as mudanças de política que você deseja fazer alternando os botões de opções **ON/OFF**.

5. Clique em **Enviar política**.

Política de sincronização

Crie uma política de sincronização para planejar um tempo específico para varrer os pods do Vulnerability Advisor (VA), imagens do VA e o processo do Mutation Advisor (MA).

Conclua as etapas a seguir:

1. Atualize o arquivo YAML para a varredura de pod do VA executando o seguinte comando:

   ```
   kubectl edit cm vulnerability-advisor-live-crawler -nkube-system
   ```

2. Edite o parâmetro `live-crawler.crontab` no arquivo YAML do pod do VA. Seu arquivo YAML pode ser semelhante ao seguinte mapa de configuração:

   ```
   apiVersion: v1
   data:
     enabled: "true"
     live-crawler.crontab: 59 22 * * *
     kind: ConfigMap
   ```

3. Atualize o arquivo YAML para a varredura de imagem do VA executando o seguinte comando:

   ```
   kubectl edit cm vulnerability-advisor-registry-crawler -nkube-system
   ```

4. Edite o parâmetro `reg-crawler.crontab` no arquivo YAML de imagem do VA. Seu arquivo YAML pode ser semelhante ao seguinte mapa de configuração:

   ```
   apiVersion: v1
   data:
     enabled: "true"
     reg-crawler.crontab: 0 10 * * *
   kind: ConfigMap
   metadata:
   ```

5. Atualize o arquivo YAML para a varredura de processo do MA executando o seguinte comando:

   ```
   kubectl edit cm vulnerability-advisor-process-ma-crawler -nkube-system
   ```

6. Edite o parâmetro `live-crawler.crontab` no arquivo YAML de processo do MA. Seu arquivo YAML pode ser semelhante ao seguinte mapa de configuração:

   ```
   apiVersion: v1
   data:
     enabled: "true"
     live-crawler.crontab: '*/15 * * * *'
   kind: ConfigMap
   metadata:
   ```

É criada uma política de sincronização para a varredura de pods do Vulnerability Advisor (VA), a varredura de imagens do VA e a varredura de processo do Mutation Advisor (MA).

Atualizando os avisos de segurança para os componentes do Vulnerability Advisor

Os avisos de segurança para toda a distribuição suportada do Linux® são pré-carregados no cluster do Elasticsearch para o Vulnerability Advisor. No entanto, os avisos de segurança para cada distribuição do Linux são atualizados periodicamente na Internet.

O IBM publica avisos de segurança enviando por push uma nova imagem do `usnloader` para o Docker Hub à 0h E.S.T. diariamente. Novas imagens do `usnloader` são marcadas com um registro de data e hora. Por exemplo, os avisos de segurança que são liberados em 10 de maio de 2018 são identificados como `cloudviz/usnloader: 20180510`. Uma imagem identificada como `latest` também é enviada por push diariamente quando a compilação é concluída à 0h E.S.T. Cada versão de registro de data e hora da imagem `usnloader` fica disponível no Docker Hub por 7 dias.

Pré-requisitos

Se seu ambiente não tiver acesso à Internet, é preciso extrair manualmente a imagem `usnloader` do Docker Hub diariamente. Para configurar um pull manual, conclua as etapas a seguir:

1. Crie uma Tarefa cron do Linux em um host que tenha acesso à Internet. Planeje a Tarefa Cron para fazer pull da imagem `usnloader` todos os dias às 17h E.S.T.
2. Envie por push a imagem `usnloader` mais recente para seu registro privado do IBM Cloud Private. Consulte Enviando por push e efetuando pull de imagens.
3. Conclua o procedimento para atualizar os avisos de segurança. Assegure-se de atualizar a especificação de `image` no Kubernetes CronJob `usnloader.yaml` para apontar para a imagem no registro privado do IBM Cloud Private. Por exemplo `image: mycluster.icp: 8500/services/usnloader:latest`.

Procedimento

Para atualizar os avisos de segurança para o cluster do IBM Cloud Private, conclua as etapas a seguir:

1. Configure a CLI do `kubectl`. Consulte Acessando o cluster do IBM Cloud Private usando a CLI kubectl.

2. Crie um Kubernetes CronJob `usnloader.yaml` usando as especificações a seguir.

```
---
  apiVersion: batch/v1beta1
  kind: CronJob
  metadata:
    labels:
      app: usnloader
      component: vulnerability-advisor
    name: usnloader
    namespace: kube-system
  spec:
    concurrencyPolicy: Replace
    failedJobsHistoryLimit: 1
    successfulJobsHistoryLimit: 3
    schedule: '0 6 * * *'
 suspend: false
 jobTemplate:
   spec:
     template:
       spec:
         containers:
         - command: ["python2.7", "/opt/usnloader/usnloader.py",
                     "--elasticsearch-urls", "https://elasticsearch:9200", "--ca-file",
"/tls/ca.crt",
                     "--client-cert", "/tls/curator.crt", "--client-key", "/tls/curator.key"]
           image: cloudviz/usnloader:latest
           imagePullPolicy: Always
           name: usnloader
           volumeMounts:
           - mountPath: /var/log/cloudsight/
             name: log
           - mountPath: /tls
             name: certs
             readOnly: true
         nodeSelector:
           va: "true"
         restartPolicy: OnFailure
         tolerations:
         - effect: NoSchedule
           key: "dedicated"
           operator: "Exists"
         - key: "CriticalAddonsOnly"
           operator: "Exists"
         volumes:
         - name: certs
           secret:
             defaultMode: 420
             secretName: logging-elk-certs
         - emptyDir: {}
           name: log
```

Para carregar os avisos de segurança para uma data específica, é possível criar uma tarefa em lote do Kubernetes, `usnloader.yaml` e especificar a imagem para a data desejada. A tarefa em lote pode ser semelhante ao código a seguir:

```
---
apiVersion: batch/v1
kind: Job
```

```
metadata:
  name: usnloader
  namespace: kube-system
  labels:
    app: usnloader
    component: vulnerability-advisor
spec:
  template:
    metadata:
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ""
      name: vulnerability-advisor-usncrawler
    spec:
      containers:
      - command:
        - python2.7
        - /opt/usnloader/usnloader.py
        - -- elasticsearch-urls
        - https://elasticsearch: 9200
        - -- ca-arquivo
        - /tls/ca.crt
        - -- client-cert
        - /tls/curator.crt
        - -- client-key
        - /tls/curator.key
        image: "cloudviz/usnloader:latest"
        imagePullPolicy: Always
        name: usnloader
        volumeMounts:
        - mountPath: /var/log/cloudsight/
          name: log
        - mountPath: /tls
          name: certs
          readOnly: true
      dnsPolicy: ClusterFirst
      nodeSelector:
        va: "true"
      priorityClassName: system-cluster-critical
      restartPolicy: OnFailure
      terminationGracePeriodSeconds: 30
      tolerations:
      - effect: NoSchedule
        key: dedicated
        operator: Exists
      volumes:
      - name: certs
        secret:
          defaultMode: 420
          secretName: logging-elk-certs
      - emptyDir: {}
        name: log
```

3. Ative a Tarefa usnloader.

```
kubectl aplicar -f usnloader.yaml
```

4. Verifique a tarefa.

```
kubectl -n kube-system get cronjob | grep usnloader
```

A saída se assemelha ao código a seguir:

```
NAME                                          SCHEDULE      SUSPEND   ACTIVE    LAST SCHEDULE
AGE
usnloader                                     0 6 * * *     False     1         29s
4m
```

O CronJob puxa a imagem mais recente do Docker Hub e carrega os avisos de segurança mais recentes para o componente Elasticsearch do seu Vulnerability Advisor.

```
kubectl -n kube-system get job | grep usnloader
```

A saída se assemelha ao código a seguir:

```
usnloader-1526436600                                  1         0           33s
```

```
kubectl -n kube-system get pods --show-all | grep usnloader
```

A saída se assemelha ao código a seguir:

```
apiVersion: batch/v1beta1
usnloader-1526436600-846nf                                        0/1       Completed   0
59s

kubectl -n kube-system logs -f usnloader-1526436600-846nf
```

A saída se assemelha ao código a seguir:

```
2018-05-16 02:10:20,581 INFO 63 usnloader: Arguments received from the command line
2018-05-16 02:10:20,582 INFO 66 usnloader: {'elastic_search': 'vulnerability-advisor-
elasticsearch:9200', 'elastic_search_password': '**********'}
2018-05-16 02:10:42,731 INFO 79 usnloader: No new usns
2018-05-16 02:10:42,744 INFO 58 log_update_status: [
  {
    "latest_advisory": "deb-2018-msg00126.html",
    "index_load_time": "2018-05-16T02:10:07.866827",
    "distro": "debian"
  },
  {
    "latest_advisory": "alpine_git_commit:",
    "index_load_time": "2018-05-15T03:02:11.375949",
    "distro": "alpine"
  },
  {
    "latest_advisory": "RHSA-2018:0998",
    "index_load_time": "2018-05-16T02:10:07.744258",
    "distro": "redhat"
  },
  {
    "latest_advisory": "centos-2018-May.txt.gz",
    "index_load_time": "2018-05-16T02:10:07.832857",
    "distro": "centos"
  },
  {
    "latest_advisory": "FEDORA-2018-05",
    "index_load_time": "2018-05-16T02:10:07.656827",
    "distro": "fedora"
  },
  {
    "latest_advisory": "ubuntu-2018-May.txt.gz",
    "index_load_time": "2018-05-16T02:10:07.551024",
    "distro": "ubuntu"
  }
]
```

Agora você está pronto para usar o Vulnerability Advisor com os avisos de segurança atualizados. Também é possível fazer a varredura de registros de imagem externa com o Vulnerability Advisor. Consulte Varrendo registros externos com o Vulnerability Advisor (VA) para obter mais detalhes.

Varrendo registros externos de imagem com o Vulnerability Advisor (VA)

É possível varrer registros externos de imagem com o VA.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Configurando o Vulnerability Advisor para varrer um registro de imagem externo

Configure o Vulnerability Advisor para incluir, editar e remover um registro de imagem externo para ser varrido.

Pré-requisito:

- Assegure-se de que o nó do VA possa se conectar ao registro quando quiser fazer a varredura executando o seguinte comando:

  ```
  docker login $registry_addr -u $username -p $password
  ```

- Inclua um registro de imagem externo com as seguintes etapas:

1. Crie um segredo de pull de imagem que o IBM Cloud Private Kubernetes possa usar para fazer pull de uma imagem a partir de um registro de imagem externo:

   ```
   kubectl -n kube-system create secret docker-registry $secret_name --docker-server=$registry_addr --docker-username=$username --docker-password=$password  --docker-email=$your_mail
   ```

2. No menu de navegação na interface com o usuário do VA, clique em **Ferramentas** > **Vulnerability Advisor** > **kube-system** > **Registros**.

3. Clique no botão **Incluir Novo Registro** para abrir a janela Configuração de Registro.

4. Insira o endereço do registro na coluna *Registro*.

5. Selecione o tipo de registro no menu *Tipo*. Suas opções são `dockerhub, artifactory, icp ou harbor`.

6. Insira o nome do segredo do pull de imagem que você criou.

7. Defina quais imagens desejar varrer inserindo uma sequência de escopo com formato regex. Sua sequência de escopo pode ser semelhante ao seguinte conteúdo:

   ```
   ".*"   means scan all images in the registry
   "^library"  means only scan images under path library/ in the registry
   "^library/va-crawler"  means only scan image `va-crawler` under path library/ in the registry
   ```

8. Selecione True ou False no menu *Ativar* para ativar a varredura do seu registro externo.

9. Salve a configuração clicando-se no botão **Salvar**.

   Um registro de imagem externo é incluído.

- Edite o registro de imagem externo com as seguintes etapas:

  1. Selecione um registro existente no menu suspenso *Selecionar um Registro*.

  2. Edite seu registro clicando no botão *Editar Registro Selecionado*.

  3. Clique no botão *Salvar* após editar sua configuração.

     Seu registro de imagem externo é editado.

- Remova um registro de imagem externo com as seguintes etapas:

  1. Selecione um registro existente no menu suspenso *Selecionar um Registro*.

  2. Edite seu registro clicando no botão *Editar Registro Selecionado*.

  3. Remova sua configuração clicando no botão *Excluir*.

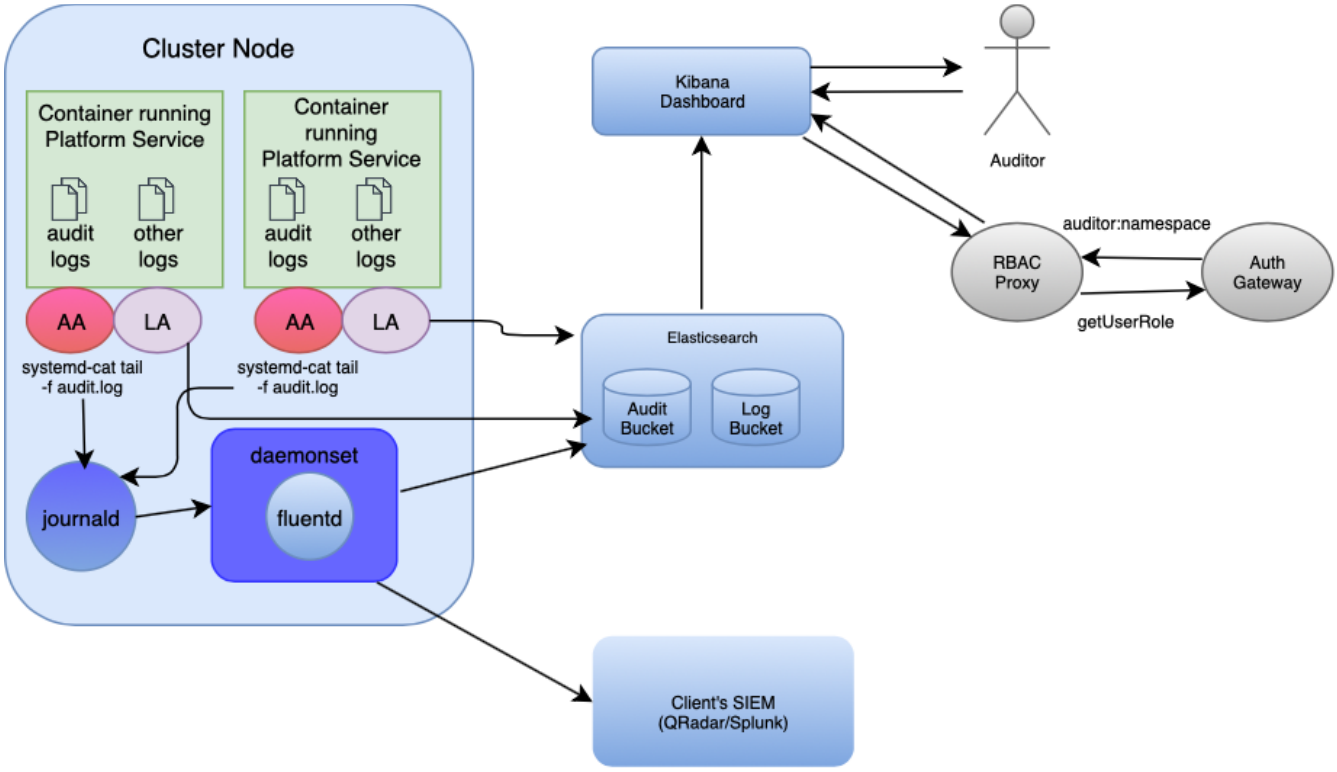
     Seu registro de imagem externo é removido.

O Vulnerability Advisor é configurado para a varredura de registros de imagem externos.

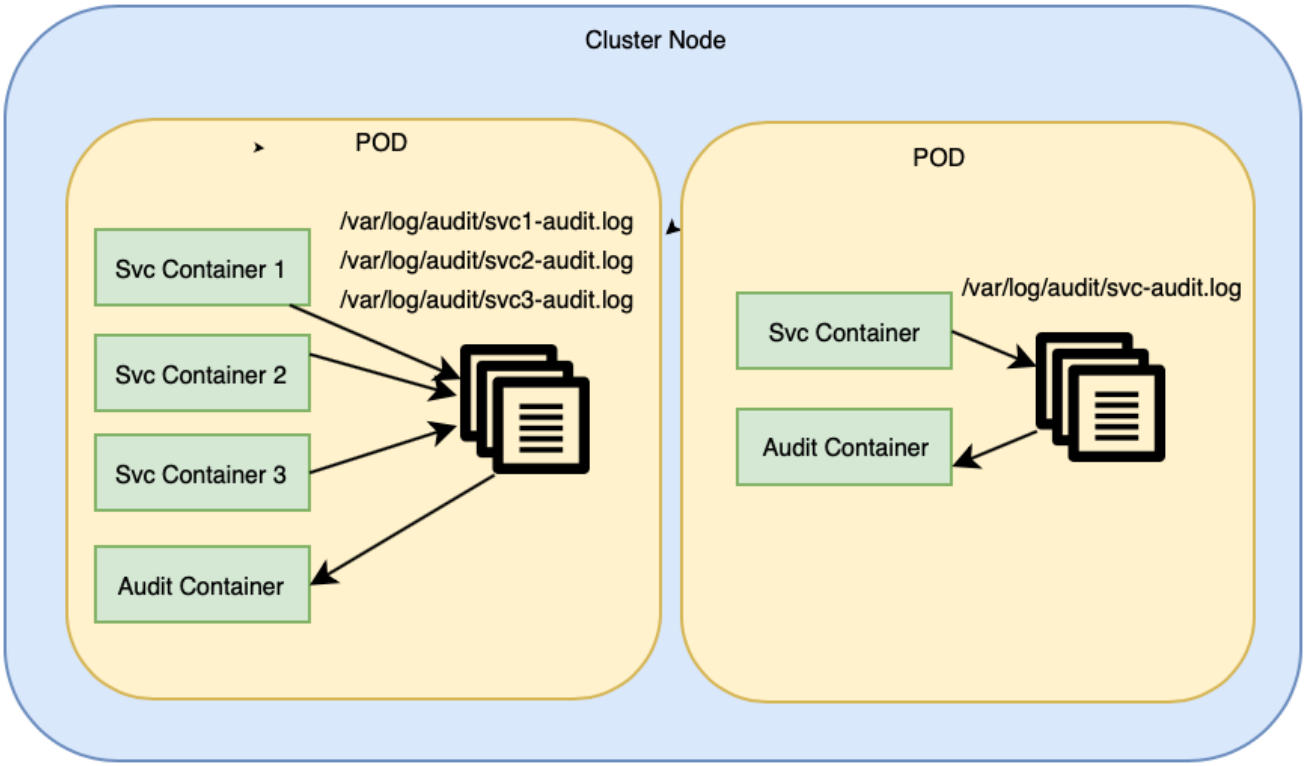
Guia de adoção de criação de log de auditoria

Configure o cluster do IBM® Cloud Private para gerar logs de auditoria e rotear os logs para o gerenciamento de informações de segurança e eventos (SIEM).

Arquitetura de criação de log de auditoria do IBM Cloud Private



AA - Audit Agent; LA - Application Log Agent



A seguir estão os componentes principais da arquitetura de criação de log de auditoria:

Contêiner de auditoria

O contêiner de auditoria é um contêiner sidecar. Ele acompanha o arquivo `audit.log` e canaliza-o usando o comando `system-cat`. Ele envia os logs de auditoria para o diário `systemd`.

Cada serviço que gera logs de auditoria grava os logs no arquivo `/var/log/audit/<service_name>-audit.log`. Um contêiner de serviço deve compartilhar `/var/log/audit` com o contêiner de auditoria.

Um volume `emptyDir` é usado para compartilhar o diretório `/var/log/audit` entre os contêineres de serviço para auditar o contêiner sidecar em um pod.

Como o contêiner de auditoria deve gravar no diário `systemd`, ele também precisa montar o sistema de arquivos host em que o diário do sistema existe.

A ferramenta logrotate é usada para monitorar os logs no diretório `/var/log/audit` para tamanho, período de rotação e outros parâmetros, e para reciclar os logs de auditoria conforme especificado na configuração.

Journald (diário systemd)

`systemd` é um serviço que é executado em um nó. Os logs de auditoria que são gerados por serviços do IBM Cloud Private que são executados em pods que possuem contêiner sidecar de auditoria são enviados para o diário `systemd`. O diário `systemd` armazena os dados em formato binário. Os dados podem ser apenas anexados. Depois que o diário `systemd` recebe os dados de auditoria, ele é selecionado por `fluentd` e, em seguida, enviado para o Elasticsearch ou SIEM.

Fluentd

O Fluentd é um coletor do log que usa plug-ins de entrada e saída para coletar dados de várias origens e para distribuir ou enviar dados para vários destinos.

Fluentd coleta logs de auditoria do diário `systemd` usando o plug-in de entrada `fluent-plugin-systemd`. Esse plug-in tem capacidade de filtragem de dados e coleta somente logs de auditoria do diário `systemd`.

Os contêineres fluentd montam um sistema de arquivos host no qual os dados do log de diário são armazenados. O local padrão é `/run/log/journal`.

Fluentd, por padrão, envia os logs de auditoria para a pilha Elasticsearch Logstash Kibana (ELK) usando o plug-in de saída `fluent-plugin-elasticsearch`. O Fluentd pode ser configurado para enviar logs para uma ferramenta SIEM corporativa, como o QRadar.

As seguintes opções de configuração são importantes para o plug-in de saída do ponto de vista da criação de log de auditoria:

- Opções de certificado de cliente ou de host. Essas opções ativam a autenticação de Infraestrutura de Chave Pública (PKI) com o ELK.

  ```
  ca_file /path/to/your/ca/cert

  client_cert /path/to/your/client/cert
  client_key /path/to/your/private/key
  client_key_pass password

  ssl_version TLSv1_2
  ```

  Nota: é possível alimentar a senha a partir de um segredo usando a seguinte instrução:

  ```
  client_key_pass "#{ENV["APP_KEYSTORE_PASSWORD"]}"
  ```

- Opção Logstash. Esta opção coloca dados de criação de log de auditoria em seu depósito para isolar os logs de auditoria de outros logs de depuração.

  ```
  logstash_format true # defaults to false
  logstash_prefix mylogs # defaults to "logstash"
  logstash_prefix_separator _ # defaults to "-"
  logstash_dateformat %Y.%m. # defaults to "%Y.%m.%d"
  ```

ELK

A pilha de ELK é usada para armazenar, indexar e representar logs do IBM Cloud Private.

O ELK também é usado para criação de log de auditoria. Os dados que são recebidos pelo ELK a partir do Fluentd são analisados e, em seguida, colocados em depósitos separados: um para criação de log e outro para dados de auditoria.

Eventos de log de auditoria

Os eventos a seguir podem ser registrados como eventos de auditoria:

- Eventos de autenticação (tentativa de login bem-sucedida ou com falha)

- Eventos de autorização (tentativa bem-sucedida ou com falha de acessar recursos ou dados)
- Modificação de configuração do sistema
- Criar, ler, excluir ou atualizar recursos ou dados
- Falha de gerenciamento de sessões
- Mudanças nos privilégios de usuário
- Acesso ao banco de dados (sucesso ou falha)
- Eventos de firewall de rede ou de aplicativo, de Sistema de Detecção de Intrusão (IDS) ou de Sistema de Prevenção de Intrusão (IPS)
- Quantidade de uso do sistema
- Eventos de início, encerramento ou reinicialização do sistema
- Eventos de falha do aplicativo ou do sistema

Os dados a seguir não devem ser incluídos nos logs de auditoria:

- Informações confidenciais
- Credenciais do usuário
- Senhas
- Detalhes de conta bancária
- Token de Acesso
- Token de Autenticação
- Caminho ou informações do sistema de arquivos
- Consulta ou sequência de banco de dados
- Chaves de criptografia ou decriptografia

Formato de criação de log de auditoria

O IBM Cloud Private segue os padrões de Cloud Auditing Data Federation (CADF). O padrão define um modelo de evento para coletar os dados necessários para auditoria. O IBM Cloud Private inclui alguns campos customizados para gerar logs abrangentes.

Para obter informações adicionais sobre o CADF, consulte Cloud Auditing Data Federation.

Os campos a seguir são importantes:

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    "eventType": "activity",
    "id": "icp:e97c7b00-e215-11e8-abf8-79cb75b57820",
    "action": "create",
    "requestPath": "/identity/api/v1/teams",
    "initiator": {
        "typeURI": "service/security/account/user",
        "name": "admin",
        "credential": {
            "type": "token"
        },
        "host": {
            "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14) AppleWebKit/605.1.15
(KHTML, like Gecko) Version/12.0 Safari/605.1.15",
            "address": "icp-management-ingress:8443"
        }
    },
    "target": {
        "id": "c4e8170e90a7c01a228fbef74c22245d2665cefffac1662907a6a75e82319a74",
        "name": "icp-testing-audit-logs",
        "actions": {
            "name": "icp-testing-audit-logs",
            "teamId": "icp-testing-audit-logs",
            "users": [],
            "usergroups": [],
            "directoryList": []
        },
        "typeURI": "service/security/group"
    },
    "observer": {
        "id": "target"
    },
    "severity": "normal",
    "outcome": "success",
    "reason": {
        "reasonType": "HTTP",
```

```
        "reasonCode": 200
    },
    "eventTime": "2018-11-06T22:47:17.424Z",
    "kubernetes.container_id": "c4e8170e90a7c01a228fbef74c22245d2665cefffac1662907a6a75e82319a74",
    "kubernetes.container_name": "platform-identity-management",
    "kubernetes.pod": "auth-idp-mw2x9",
    "kubernetes.namespace": "kube-system",
    "origination": "ui",
    "version": "v1.0"
}
```

Acessar estrutura do CADF de linguagem

```
type CADF struct {
    TypeURI   string `json:"typeURI"`
    Action    string `json:"action"`
    ID        string `json:"id"`
    Initiator struct {
        Name       string `json:"name"`
        TypeURI    string `json:"typeURI"`
        Credential struct {
            Type string `json:"type"`
        } `json:"credential"`
    } `json:"initiator"`
    Target struct {
        ID      string `json:"id"`
        Name       string `json:"name"`
        TypeURI    string `json:"typeURI"`
    } `json:"target"`
    RequestPath             string `json:"requestPath"`
    EventType               string `json:"eventType"`
    Severity                string `json:"severity"`
    Outcome                 string `json:"outcome"`
    EventTime               string `json:"eventTime"`
    KubernetesContainerID   string `json:"kubernetes.container_id"`
    KubernetesContainerName string `json:"kubernetes.container_name"`
    KubernetesPod           string `json:"kubernetes.pod"`
    KubernetesNamespace     string `json:"kubernetes.namespace"`
    Observer                struct {
        ID string `json:"id"`
    } `json:"observer"`
    Origination string `json:"origination"`
    Version     string `json:"version"`
}
```

Estrutura do CADF NodeJs

```
 let cadf = {
      "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
      "eventType": "activity",
      "id": "icp:"+uuid1,
      "action": action,
      "requestPath": path,
      "initiator": {
    "typeURI": (user ? "service/security/account/user": ''),
        "name": user,
    "credential": {
      "type": "token"
        },
        "host": {
          "user-agent": req.headers['user-agent'],
          "address": req.headers['host']
    }
      },
    "target": {
    "id": cont_id,
    "name": res,
    "actions": actions,
        "typeURI": (map ? map: parseUrl(path)) // pretend this app is a service
      },
      "observer": {
        "id": "target"
      },
      "severity" : severity,
```

```
  "outcome": outcome,
  "reason": {
"reasonType":"HTTP",
    "reasonCode": status, // like 200 or 400
  },
  "eventTime": expT,
  "kubernetes.container_id": cont_id,
  "kubernetes.container_name": process.env.SERVICE_NAME,
  "kubernetes.pod": process.env.POD_NAME || process.env.HOSTNAME,
  "kubernetes.namespace": process.env.POD_NAMESPACE,
  "origination": identifyOrig(req.headers['referer'] || req.headers['user-agent']),
  "version": "v1.0"
};
```

Modificação do arquivo de implementação

- Inclua um `contêiner sidecar de criação de log de auditoria` na primeira posição sob a seção `contêiner` no arquivo de implementação:

```
- name: icp-audit-service
  image: mycluster.icp:8500/ibmcom/icp-audit-service:3.1.1
  imagePullPolicy: IfNotPresent
  env:
  - name: POD_NAME
    valueFrom:
      fieldRef:
        apiVersion: v1
        fieldPath: metadata.name
  resources: {}
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
  volumeMounts:
  - mountPath: /run/systemd/journal
    name: journal
  - mountPath: /var/log/audit
    name: shared
```

Nota: O caminho da imagem depende da versão e do tipo de instalação do IBM Cloud Private. O exemplo a seguir é de uma instalação offline no IBM Cloud Private versão 3.1.1:

- Monte o volume compartilhado `/var/log/audit/` em todos os contêineres em um pod:

```
  volumeMounts:
  - mountPath: /var/log/audit
    name: shared
```

- Alguns dos campos customizados no formato CADF podem ser fornecidos em um contêiner usando as variáveis de ambiente `kubernetes.container_name`, `kubernetes.pod` e `kubernetes.namespace`.

```
- env:
  - name: SERVICE_NAME
    value: key-management-lifecycle
  - name: POD_NAME
    valueFrom:
      fieldRef:
        apiVersion: v1
        fieldPath: metadata.name
  - name: POD_NAMESPACE
    valueFrom:
      fieldRef:
        apiVersion: v1
        fieldPath: metadata.namespace
  - name: CONFIG_PATH
    value: /opt/keyprotect/config/
  - name: ICP_NAMESPACE
    value: kube-system
  - name: CLUSTER_NAME
    valueFrom:
      configMapKeyRef:
        key: CLUSTER_NAME
        name: platform-auth-idp
```

- Em um contêiner de serviço, o aplicativo que está em execução cria um arquivo `<service_name>-audit.log` no diretório que está montado em `/var/log/audit/`.
- Grave ou anexe todos os logs de auditoria que são gerados pelo aplicativo ou serviço ao arquivo `<service_name>-audit.log`. O contêiner sidecar de auditoria encaminha os logs para o `journald` e o `journald` encaminha para o `fluentd`.
- Os logs de auditoria devem estar no formato JSON para que sejam analisados corretamente pelo Fluentd e por uma ferramenta SIEM. Consulte a amostra a seguir:


```
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:16d
5af60-4cc2-11e9-9451-
57b95a7e8968","action":"update","requestPath":"/identity/api/v1/teams/test-
team/resources","initiator":
{"typeURI":"service/security/account/user","name":"admin","credential":{"type":"token"},"host":
{"user-agent":"curl/7.47.0","address":"<cluster-ip>:8443"}},"target":
{"id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07436122671","name":"test-
team","actions":"crn:v1:icp:private:k8:mycluster.icp:n/default:::","typeURI":"service/security/
group"},"observer":{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:46:50.198Z","kubernetes.container_id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2
b4a07436122671","kubernetes.container_name":"platform-identity-
management","kubernetes.pod":"auth-idp-8jdlx","kubernetes.namespace":"kube-
system","origination":"cli","version":"v1.0"}
```


- If there are multiple service containers in a single pod, there is a separate audit log file for each service container in the `/var/log/audit/` directory and only one file for audit sidecar container. The volume of the `/var/log/audit/` directory is shared by all containers in a pod.
- You can enable or disable audit logs for a service. Add the `AUDIT_ENABLED` flag in the ConfigMap. The flag can be added as an environment variable so that the application code has access to enable or disable generation of audit records.

A seguir está um exemplo de arquivo de implementação:

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
  # and an empty file will abort the edit. If an error occurs while saving this file will be
  # reopened with the relevant failures.
#
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.17.0 ()
  creationTimestamp: null
  labels:
    io.kompose.service: think-blue-demo-app
  name: think-blue-demo-app
  namespace: jkstore
spec:
  replicas: 1
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        io.kompose.service: think-blue-demo-app
    spec:
      containers:
      - name: icp-audit-service
        image: mycluster.icp:8500/ibmcom/icp-audit-service:3.1.2
        imagePullPolicy: IfNotPresent
        env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.name
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
        volumeMounts:
```

```
        - mountPath: /run/systemd/journal
          name: journal
        - mountPath: /var/log/audit
          name: shared
      - name: think-blue-demo-app
        image: mycluster.icp:8500/kube-system/think-blue-demo_app:0.1
        imagePullPolicy: IfNotPresent
        args:
        - npm
        - start
        ports:
        - containerPort: 30003
        env:
        - name: CLIENT_ID
          value: "9b570f23952a45099966ffa2cf7b6355"
        - name: CLIENT_SECRET
          value: "3rQwon8BrfpOEqzX2RttUXAt4CVkf8S2WuBQKiE5wPHKudMGX0FIlARejVf9"
        - name: AUTH_URL
          value: "https://<CLUSTER_IP>:8443/idprovider/v1/auth/authorize"
        - name: TOKEN_URL
          value: "https://<CLUSTER_IP>:8443/v1/auth/token"
        - name: ISSUER_ID
          value: "https://mycluster.icp:9443/oidc/endpoint/OP"
        - name: CALLBACK_URL
          value: "https://bobcat.rtp.raleigh.ibm.com/auth/liberty/callback"
        - name: LOGOUT_URL
          value: "https://<CLUSTER_IP>:8443/v1/auth/logout"
        - name: AUDIT_ENABLED
          value: "true"
        - name: CONTAINER_ID
          value: "think-blue-demo-app"
        - name: SERVICE_NAME
          value: "think-blue-demo-app"
        - name: AUTHZ_URL
          value: "https://<CLUSTER_IP>:8443/iam-pdp/v1/authz"
        - name: PAYMENT_URL
          value: "https://<CLUSTER_IP>:8443/payments/payment/"
        - name: AUDIT_ENABLED
          valueFrom:
            configMapKeyRef:
              name: "think-blue-demo-ConfigMap"
              key: AUDIT_ENABLED
        resources: {}
        volumeMounts:
        - mountPath: /var/log/audit
          name: shared
      restartPolicy: Always
      volumes:
      - hostPath:
         path: /run/systemd/journal
         type: ""
        name: journal
      - emptyDir: {}
        name: shared
status: {}
```

Note: `icp-audit-service` is at first position. The `/var/log/audit` volume is mounted and shared between application container and audit sidecar container. `AUDIT_ENABLED` flag is imported from the ConfigMap file.

- Authentication and authorization audit logs
- Audit logging in IBM Cloud Private
- Audit logging data statistics
- Audit logging Kibana dashboard
- IBM Cloud Private audit logging integration with enterprise SIEM tools

Autenticação e logs de auditoria de autorização

IBM Cloud Private gera logs de auditoria de autenticação e autorização. O serviço `auth-idp` gera eventos de auditoria de autenticação e o serviço `auth-pdp` gera logs de auditoria de autorização.

Os logs de auditoria contêm os campos a seguir.

Tabela 1. Dados do log de auditoria de autenticação e autorização

Nome de Campo	Dados salvos	Descrição	Exemplo
Initiator.id	Origem de uma ação	ID da origem que iniciou a ação	ID do LDAP; ID de uma chave API
Initiator.typeURI	URI de origem	URI da origem de ação	Serviço; usuário
Initiator.credential.type	Tipo de ID	Tipo de ID da origem de ação	Token; chave de API
target.name	Alvo de uma ação	O terminal no qual a ação é iniciada.	Serviço; recurso
Target.id	ID do destino	O valor de Cloud Resource Name (CRN) do serviço ou recurso	Crn:v1 :icp:private:platform-service :::core:service:metering de serviço
Target.typeURI	URI do destino	URI do destino no qual a ação é iniciada.	Recurso; chave de API; segredo
.	Ação que é solicitada	A ação que aciona um evento.	Criar, atualizar, excluir; implementar; autenticar
Resultado	Resultado da ação		Sucesso; pendente; falha
Reason.reasonCode	código de resposta HTTP	O código de resposta do resultado.	200 para sucesso
Gravidade	Nível de Severidade	O nível de gravidade do evento.	Crítica; normal
eventTime	Registro de data e hora	A hora, data e fuso horário do evento.	2018-04-20 20 :15:00.32 +0000 UTC

Todas as operações criar, ler, atualizar e excluir (CRUD) que estão relacionadas a um diretório, usuário, grupo de usuários e equipe são registradas.

Cada serviço que gera dados de auditoria grava registros de auditoria em um arquivo `/var/log/audit/<service_name>-audit.log` dentro do contêiner sidecar de auditoria no respectivo pod. Por exemplo:

- Os logs de autenticação são salvos no arquivo `/var/log/audit/platform-identity-management-audit.log`.
- Os logs de autorização são salvos no arquivo `/app/logs/audit/pdp-audit.log`.

O diretório `/var/log/audit` é compartilhado com o contêiner de auditoria, que é um contêiner sidecar. Um volume `emptyDir` é usado para compartilhar o diretório `/var/log/audit` entre os dois contêineres. O contêiner de auditoria (também conhecido como agente automático) envia os dados para o diário `systemd`.

Observe que a criação de log está desativada por padrão. Para ativar a criação de log, deve-se configurar a variável `AUDIT_ENABLED` no configmap para `true`. Para obter informações sobre como gerar logs de auditoria, consulte Configurando os serviços do IBM Cloud Private para gerar logs de auditoria.

É possível usar uma ferramenta security information and event management (SIEM) de sua escolha para visualizar esses logs.

Registro de Auditoria no IBM Cloud Private

O recurso de criação de log de auditoria no IBM Cloud Private fornece a capacidade de coletar logs de auditoria gerados por vários serviços de plataforma e pelo servidor da API do Kubernetes e de enviá-los para o Elasticsearch ou o Gerenciamento de informações de segurança e de eventos (SIEM).

Há dois tipos de logs de auditoria:

- Logs de auditoria que são gerados pelo servidor da API do Kubernetes
- Logs de auditoria que são gerados por serviços de plataforma

Formato do log de auditoria

Os dados de auditoria que são gerados em serviços de plataforma estão em conformidade com o padrão Cloud Auditing Data Federation (CADF). O evento CADF é registrado no formato JSON. Os dados de auditoria que são gerados pelo servidor da API do Kubernetes usam o recurso "AdvancedAuditing" e também estão no formato JSON.

Local dos logs de auditoria

Os dados de auditoria que são gerados em cada serviço são enviados primeiramente para o diário systemd no nó em que o serviço está em execução. Os dados de auditoria que são gerados pelo servidor da API do Kubernetes são salvos em `/var/log/k8saudit/audit.log` no nó. Um daemonset fluentd é implementado como parte da criação de log de auditoria. Em cada nó, o fluentd recupera os dados de auditoria do log de diário systemd e também do log de auditoria do Kubernetes e envia os dados para o Elasticsearch ou o SIEM. O serviço do Elasticsearch ou SIEM que recebe os dados de auditoria é o mesmo serviço que foi implementado para coletar logs do aplicativo. Um depósito separado, como um índice, é criado no Elasticsearch ou SIEM para dados de auditoria.

Ativando e desativando a criação de log de auditoria para serviços IBM Cloud Private

Conclua as etapas a seguir para ativar ou desativar a criação de log de auditoria.

1. A partir do menu de navegação, clique em **Configuração** > **ConfigMap**
2. Procure o ConfigMap do serviço para o qual você deseja ativar a criação de log. Clique em **Editar**.
3. Configure a chave relacionada à auditoria como `true` ou `false` para ativar ou desativar a criação de log de auditoria para esse serviço. Clique em **Enviar**.
4. Remova todos os pods que pertencem ao serviço. Os pods são recriados com a auditoria ativada ou desativada. É possível visualizar serviços nos locais a seguir:
    - A partir do menu de navegação, clique em **Carga de Trabalho** > **DaemonSets**.
    - A partir do menu de navegação, clique em **Carga de Trabalho** > **Implementações**.

    Para obter informações adicionais, consulte Tabela 1. Serviços do IBM Cloud Private e os ConfigMaps nos quais as chaves relacionadas à auditoria estão configuradas..

Visualizando Dados de Auditoria em Painéis do Kibana

O acesso aos dados de auditoria no Elasticsearch ou SIEM é fornecido por meio do Kibana. Somente os usuários que são designados à função de `Auditor` ou à função de `administrador de cluster` podem visualizar os dados de auditoria. Outras restrições baseadas nos namespaces também são aplicáveis. Os usuários designados à função de `Auditor` ou à função de `administrador de cluster` podem visualizar somente dados de auditoria que pertencem aos namespaces aos quais eles têm acesso. Para obter informações detalhadas sobre o acesso a dados de auditoria, consulte Criação de log do IBM Cloud Private.

Para obter informações sobre como ativar a auditoria do Kubernetes, consulte Gerando logs de auditoria do Kubernetes.

Estatísticas de dados de criação de log de auditoria

Há dois tipos de dados de criação de log de auditoria gerados em clusters do IBM Cloud Private:

- Auditoria do Kubernetes
- Auditoria de serviços do IBM Cloud Private Platform.

Este artigo fornece uma perspectiva sobre a quantia de dados de auditoria que é gerada. Use as informações para ajudar a ajustar políticas de auditoria, alocar espaço em disco e preparar ELK ou SIEM para manipular registros de auditoria.

- Criação de log de auditoria do Kubernetes
- Criação de log de auditoria de serviços da plataforma

Criação de log de auditoria do Kubernetes

O servidor de API do Kubernetes gera registros de auditoria para diferentes grupos de API. Os grupos de API possuem seus próprios recursos. É possível usar os seguintes fatores para customizar seus registros de auditoria do Kubernetes: Nível de log (metadata, request, requestResponse), apiGroups, recursos de apiGroup, usuários, verbos e assim por diante.

Políticas de auditoria do Kubernetes

É possível customizar o arquivo `audit-policy.yaml` para gerar e filtrar registros de auditoria apropriados. O Kubernetes pode gerar um número grande de registros de auditoria que requerem espaço em disco extra no nó e tráfego excessivo para SIEM e ELK. Consulte sua equipe de segurança e conformidade para ajustar suas políticas.

A tabela a seguir lista o número de grupos de API e seus recursos relacionados:

Tabela 1. Grupos de API

Número	apiGroup	Recursos
1	networking.k8s.io	networkpolicies
2	events.k8s.io	eventos
3	scheduling.k8s.io	priorityclasses
4	monitoringcontroller.cloud.ibm.com	alertrules
		monitoringdashboards
5	storage.k8s.io	volumeattachments
		storageclasses
6	authentication.k8s.io	tokenreviews
7	apps	implementações
		replicasets
		daemonsets
		controllerrevisions
		statefulsets
8	apiregistration.k8s.io	apiservices
9	Política	poddisruptionbudgets
		podsecuritypolicies
10	mcm.ibm.com	clusterstatuses
11	core	configmaps
		persistentvolumeclaims
		podtemplates
		resourcequotas
		segredos
		replicationcontrollers
		persistentvolumes
		limitranges
		nós
		namespaces
		serviços
		serviceaccounts
		terminais
		pods
		eventos
12	metrics.k8s.io	pods
13	coordination.k8s.io	leases
14	autoscaling	horizontalpodautoscalers
15	icp.ibm.com	imagens
16	rbac.authorization.k8s.io	clusterrolebindings
		Roleligações
		função
		clusterroles
17	servicecatalog.k8s.io	servicebindings
		serviceinstances
		clusterserviceclasses
		clusterservicebrokers
		clusterserviceplans
18	certmanager.k8s.io	certificados
		desafios
		clusterissuers
		pedidos
		emissores
19	authorization.k8s.io	subjectaccessreviews

Número	apiGroup	Recursos
20	apiextensions.k8s.io	customresourcedefinitions
21	Lote	cronjobs
		tarefas
22	extensões	ingresses
		podsecuritypolicies
		daemonsets
		implementações
		networkpolicies
		replicasets
23	admissionregistration.k8s.io	initializerconfigurations
		mutatingwebhookconfigurations
		validatingwebhookconfigurations
24	securityenforcement.admission.cloud.ibm.com	imagepolicies
		clusterimagepolicies

Estatísticas de criação de log de auditoria do Kubernetes

Gerar logs de auditoria para todas as solicitações.

Use um dos métodos a seguir para gerar logs de auditoria para todas as solicitações.

- Use uma política que não aplique nenhum filtro. Por exemplo:

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
- RequestReceived
rules:
- level: Metadata
```

- Use uma política que inclua todos os grupos de API. Alguns registros de auditoria não pertencem a nenhum grupo de API. A última linha na política gera logs de auditoria para esses tipos de solicitações. Por exemplo:

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
- RequestReceived
rules:
- level: Metadata #Request #RequestResponse #None
  resources:
  - group: "" #core group
  - group: admissionregistration.k8s.io
  - group: authorization.k8s.io
  - group: batch
  - group: apiregistration.k8s.io
  - group: apps
  - group: servicecatalog.k8s.io
  - group: metrics.k8s.io
  - group: extensions
  - group: policy
  - group: authentication.k8s.io
  - group: rbac.authorization.k8s.io
  - group: certmanager.k8s.io
  - group: storage.k8s.io
  - group: monitoringcontroller.cloud.ibm.com
  - group: securityenforcement.admission.cloud.ibm.com
  - group: apiextensions.k8s.io
  - group: autoscaling
  - group: networking.k8s.io
  - group: scheduling.k8s.io
  - group: events.k8s.io
  - group: coordination.k8s.io
  - group: mcm.ibm.com
  - group: icp.ibm.com
- level: Metadata  #Request #RequestResponse #None
```

É possível customizar políticas de auditoria usando nível de log, nível do grupo de API, nível do recurso, usuários, verbos e assim por diante. Para obter mais informações, consulte https://kubernetes.io/docs/tasks/debug-application-cluster/audit/.

Resultados estatísticos

Um experimento foi executado em um cluster do IBM Cloud Private Versão 3.1.2 que foi configurado com 1 nó principal, 1 nó de gerenciamento e 2 nós do trabalhador. Registros de auditoria são gerados para cada nível de log (`Metadata`, `Request` e `RequestResponse`) e para cada grupo de API individual incluindo todos os recursos desse grupo de API. O estágio `RequestReceived` é omitido. Se você não omitir o estágio `RequestReceived`, o servidor de API gerará múltiplos registros de auditoria para cada solicitação do cliente. Ele gera registros de auditoria por estágios. Isso leva a um aumento significativo no tamanho do registro de auditoria. Se você deseja gerar registros de auditoria para todos os estágios, remova o campo `omitStages` do arquivo `audit-policy.yaml`.

O arquivo `audit-policy.yaml` a seguir era usado para cada `apiGroup`.

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
- RequestReceived
rules:
- level: <Metadata OR Request OR RequestResponse>
  resources:
  - group: <apiGroup>
```

A tabela a seguir lista os tamanhos de geração de dados de auditoria para intervalos de tempo de 15 minutos. Os tamanhos de dados são mostrados em kilobytes (KB).

Tabela 2. Tamanhos de geração de dados do registro de auditoria

apiGroup	Meta (KB)	Request (KB)	RequestResponse (KB)
"" (core group)	6600	7600	55000
admissionregistration.k8s.io-metadata-2019-03-08T10:55:12Z/	2400	2400	3000
authorization.k8s.io-metadata-2019-03-08T11:10:29Z/	3100	4500	5000
batch-metadata-2019-03-08T11:25:45Z/	464	884	3600
apiregistration.k8s.io-metadata-2019-03-08T11:41:01Z/	1100	1800	2700
apps-metadata-2019-03-08T11:56:18Z/	420	632	884
servicecatalog.k8s.io-metadata-2019-03-08T12:11:35Z/	96	108	108
metrics.k8s.io-metadata-2019-03-08T12:26:51Z/	72	68	68
extensions-metadata-2019-03-08T12:42:07Z/	76	80	888
policy-metadata-2019-03-08T12:57:23Z/	24	20	40
authentication.k8s.io-metadata-2019-03-08T13:12:39Z/	60	120	200
rbac.authorization.k8s.io-metadata-2019-03-08T13:27:56Z/	136	136	644
certmanager.k8s.io-metadata-2019-03-08T13:43:11Z/	64	72	88
certificates.k8s.io-metadata-2019-03-08T13:58:27Z/	20	28	172
storage.k8s.io-metadata-2019-03-08T14:13:42Z/	24	28	40
monitoringcontroller.cloud.ibm.com-metadata-2019-03-08T14:28:58Z/	28	28	1700
securityenforcement.admission.cloud.ibm.com-metadata-2019-03-08T14:44:13Z/	16	16	16
apiextensions.k8s.io-metadata-2019-03-08T14:59:29Z/	12	12	52
autoscaling-metadata-2019-03-08T15:14:45Z/	12	12	20
networking.k8s.io-metadata-2019-03-08T15:30:00Z/	12	12	12
scheduling.k8s.io-metadata-2019-03-08T15:45:17Z/	12	16	16
events.k8s.io-metadata-2019-03-08T16:00:32Z/	8	8	312
coordination.k8s.io-metadata-2019-03-08T16:15:49Z/	8	8	8
mcm.ibm.com-metadata-2019-03-08T16:31:06Z/	4	4	4
icp.ibm.com-metadata-2019-03-08T16:46:22Z/	8	8	52
Total	14776	18600	74624

Registros de auditoria de amostra para cada nível de log

No exemplo a seguir, os registros de auditoria de amostra `Metadata` e `Request` são semelhantes. O registro de auditoria `Solicitação de resposta` contém um campo adicional, `responseObject`. Quando os tamanhos dos registros de auditoria aumentam com cada nível, a quantia de informações em cada registro também aumenta.

- Metadados
- Solicitação
- Resposta da solicitação

`Metadados

```
`

{
    "kind": "Event",
    "apiVersion": "audit.k8s.io/v1beta1",
    "metadata": {
        "creationTimestamp": "2019-03-08T10:55:11Z"
    },
    "level": "Metadata",
    "timestamp": "2019-03-08T10:55:11Z",
    "auditID": "980770af-d30b-43e4-be1a-5d64d276aacc",
    "stage": "ResponseComplete",
    "requestURI": "/apis/admissionregistration.k8s.io/v1alpha1/initializerconfigurations",
    "verb": "list",
    "user": {
        "username": "system:apiserver",
        "uid": "2594919c-6436-454c-9b11-a1e498f184be",
        "groups": [
            "system:masters"
        ]
    },
    "sourceIPs": [
        "::1"
    ],
    "userAgent": "hyperkube/v1.12.4+icp (linux/amd64) kubernetes/d03f642",
    "objectRef": {
        "resource": "initializerconfigurations",
        "apiGroup": "admissionregistration.k8s.io",
        "apiVersion": "v1alpha1"
    },
    "responseStatus": {
        "metadata": {},
        "code": 200
    },
    "requestReceivedTimestamp": "2019-03-08T10:55:11.593615Z",
    "stageTimestamp": "2019-03-08T10:55:11.594683Z",
    "annotations": {
        "authorization.k8s.io/decision": "allow",
        "authorization.k8s.io/reason": ""
    }
}
```

`Solicitação

```
`

{
    "kind": "Event",
    "apiVersion": "audit.k8s.io/v1beta1",
    "metadata": {
        "creationTimestamp": "2019-03-08T17:01:48Z"
    },
    "level": "Request",
    "timestamp": "2019-03-08T17:01:48Z",
    "auditID": "8f48a2b7-0609-462f-9337-1547189a7a59",
    "stage": "ResponseComplete",
    "requestURI": "/apis/admissionregistration.k8s.io/v1alpha1/initializerconfigurations",
    "verb": "list",
    "user": {
        "username": "system:apiserver",
        "uid": "eb5062c6-2f1a-4a22-abbe-e8b788a68aa2",
```

```
            "groups": [
                "system:masters"
            ]
        },
        "sourceIPs": [
            "::1"
        ],
        "userAgent": "hyperkube/v1.12.4+icp (linux/amd64) kubernetes/d03f642",
        "objectRef": {
            "resource": "initializerconfigurations",
            "apiGroup": "admissionregistration.k8s.io",
            "apiVersion": "v1alpha1"
        },
        "responseStatus": {
            "metadata": {},
            "code": 200
        },
        "requestReceivedTimestamp": "2019-03-08T17:01:48.152445Z",
        "stageTimestamp": "2019-03-08T17:01:48.153558Z",
        "annotations": {
            "authorization.k8s.io/decision": "allow",
            "authorization.k8s.io/reason": ""
        }
    }
```

Resposta de Solicitação

```
    {
        "kind": "Event",
        "apiVersion": "audit.k8s.io/v1beta1",
        "metadata": {
            "creationTimestamp": "2019-03-09T09:27:38Z"
        },
        "level": "RequestResponse",
        "timestamp": "2019-03-09T09:27:38Z",
        "auditID": "57cd7c16-9312-4efd-83ca-bc3b18a227cf",
        "stage": "ResponseComplete",
        "requestURI": "/apis/admissionregistration.k8s.io/v1alpha1/initializerconfigurations",
        "verb": "list",
        "user": {
            "username": "system:apiserver",
            "uid": "2ce622b7-d7a1-4a45-b54e-58e6319a2312",
            "groups": [
                "system:masters"
            ]
        },
        "sourceIPs": [
            "::1"
        ],
        "userAgent": "hyperkube/v1.12.4+icp (linux/amd64) kubernetes/d03f642",
        "objectRef": {
            "resource": "initializerconfigurations",
            "apiGroup": "admissionregistration.k8s.io",
            "apiVersion": "v1alpha1"
        },
        "responseStatus": {
            "metadata": {},
            "code": 200
        },
        "responseObject": {
            "kind": "InitializerConfigurationList",
            "apiVersion": "admissionregistration.k8s.io/v1alpha1",
            "metadata": {
                "selfLink": "/apis/admissionregistration.k8s.io/v1alpha1/initializerconfigurations",
                "resourceVersion": "496099"
            },
            "items": []
        },
        "requestReceivedTimestamp": "2019-03-09T09:27:38.797081Z",
        "stageTimestamp": "2019-03-09T09:27:38.798617Z",
        "annotations": {
            "authorization.k8s.io/decision": "allow",
            "authorization.k8s.io/reason": ""
```

```
    }
}
```

Criação de log de auditoria de serviços de plataforma

Múltiplos serviços de plataforma no IBM Cloud Private geram registros de auditoria. Alguns dos serviços, como autenticações (`auth-idp`), autorização (`auth-pdp`) e consultor de mutação geram um volume grande de dados de auditoria.

A tabela a seguir lista informações sobre cada serviço de plataforma:

Tabela 3. Logs de serviço de plataforma

Serviço	O nome do pod começa com	Serviços de geração de logs de auditoria	Contêineres em execução	Local do diretório de log de auditoria
Autenticações	`auth-idp`	platform-identity-provider, platform-identity-manager	platform-identity-provider, platform-identity-manager, platform-auth-service, icp-audit-service	`/var/log/audit/`
Autorização	`auth-pdp`	auth-pdp	auth-pdp, icp-audit-service	`/app/logs/audit/`
Anotador de arquivo do consultor de mutação	`vulnerability-advisor-ma-file-annotator`	vulnerability-advisor-ma-file-annotator	vulnerability-advisor-ma-file-annotator, icp-audit-service	`/var/log/audit/`
Anotador do processo do consultor de mutação	`vulnerability-advisor-process-ma-annotator`	process-ma-annotator	process-ma-annotator, icp-audit-service	`/var/log/audit/`

Logs de auditoria de amostra para cada serviço

- platform-identity-management-audit.log
- platform-identity-provider-audit.log
- pdp-audit.log
- mutation-advisor-audit.log (anotação de arquivo)
- mutation-advisor-audit.log (anotação de processo)

platform-identity-management-audit.log


```
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:168ac3b0
-4cc2-11e9-9451-57b95a7e8968","action":"create","requestPath":"/identity/api/v1/teams","initiator":
{"typeURI":"service/security/account/user","name":"admin","credential":{"type":"token"},"host":
{"user-agent":"curl/7.47.0","address":"<Cluster-IP>:8443"}},"target":
{"id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07436122671","name":"Test
Team","actions":{"teamId":"test-team","name":"Test Team","directoryList":
[]},"typeURI":"service/security/group"},"observer":
{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:46:49.707Z","kubernetes.container_id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07
436122671","kubernetes.container_name":"platform-identity-management","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:16b2be10
-4cc2-11e9-9451-57b95a7e8968","action":"update","requestPath":"/identity/api/v1/teams/test-
team","initiator":{"typeURI":"service/security/account/user","name":"admin","credential":
{"type":"token"},"host":{"user-agent":"curl/7.47.0","address":"<Cluster-IP>:8443"}},"target":
{"id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07436122671","name":"test-
team","actions":{"teamId":"test-team","name":"Test Team","users":
[{"userId":"testuser","userBaseDN":"uid=testuser,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}],"usergroups":
[{"name":"security","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com",
"roles":[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}],"directoryList":
[null]},"typeURI":"service/security/group"},"observer":
{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:46:49.969Z","kubernetes.container_id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07
436122671","kubernetes.container_name":"platform-identity-management","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:16d5af60
-4cc2-11e9-9451-57b95a7e8968","action":"update","requestPath":"/identity/api/v1/teams/test-
```

team/resources","initiator":{"typeURI":"service/security/account/user","name":"admin","credential":
{"type":"token"},"host":{"user-agent":"curl/7.47.0","address":"<Cluster-IP>:8443"}},"target":
{"id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07436122671","name":"test-
team","actions":"crn:v1:icp:private:k8:mycluster.icp:n/default:::","typeURI":"service/security/group
"},"observer":{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:46:50.198Z","kubernetes.container_id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07
436122671","kubernetes.container_name":"platform-identity-management","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:16f74120
-4cc2-11e9-9451-57b95a7e8968","action":"read","requestPath":"/identity/api/v1/teams/test-
team","initiator":{"typeURI":"service/security/account/user","name":"admin","credential":
{"type":"token"},"host":{"user-agent":"curl/7.47.0","address":"<Cluster-IP>:8443"}},"target":
{"id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07436122671","name":"test-
team","typeURI":"service/security/group"},"observer":
{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:46:50.418Z","kubernetes.container_id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07
436122671","kubernetes.container_name":"platform-identity-management","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:1a1e71c0
-4cc2-11e9-9451-57b95a7e8968","action":"delete","requestPath":"/identity/api/v1/teams/test-
team","initiator":{"typeURI":"service/security/account/user","name":"admin","credential":
{"type":"token"},"host":{"user-agent":"curl/7.47.0","address":"<Cluster-IP>:8443"}},"target":
{"id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07436122671","name":"test-
team","typeURI":"service/security/group"},"observer":
{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:46:55.708Z","kubernetes.container_id":"4b1871f2f163856e3e3f56723fa16c543af3b1386588d311c2b4a07
436122671","kubernetes.container_name":"platform-identity-management","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}

platform-identity-provider-audit.log

{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:d3f579f0
-4cc1-11e9-8103-
77a98aa80e0c","action":"authenticate","requestPath":"/v1/auth/identitytoken","initiator":
{"typeURI":"service/security/account/user","name":"admin","credential":{"type":"token"},"host":
{"user-agent":"Go-http-client/1.1","address":"<Cluster-IP>:8443"}},"target":
{"id":"5342e7942a91434bebc79a2683e6ad4a426348f079c1d87480d1ada5e39a4706","typeURI":"service/security
/credential"},"observer":{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:44:57.999Z","kubernetes.container_id":"5342e7942a91434bebc79a2683e6ad4a426348f079c1d87480d1ada
5e39a4706","kubernetes.container_name":"platform-identity-provider","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:d423b5e0
-4cc1-11e9-8103-
77a98aa80e0c","action":"authenticate","requestPath":"/v1/auth/identitytoken","initiator":
{"typeURI":"","name":"","credential":{"type":"token"},"host":{"user-agent":"Go-http-
client/1.1","address":"<Cluster-IP>:8443"}},"target":
{"id":"5342e7942a91434bebc79a2683e6ad4a426348f079c1d87480d1ada5e39a4706","typeURI":"service/security
/credential"},"observer":{"id":"target"},"severity":"normal","outcome":"success","reason":
{"reasonType":"HTTP","reasonCode":200},"eventTime":"2019-03-
22T16:44:58.302Z","kubernetes.container_id":"5342e7942a91434bebc79a2683e6ad4a426348f079c1d87480d1ada
5e39a4706","kubernetes.container_name":"platform-identity-provider","kubernetes.pod":"auth-idp-
8jdlx","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}

pdp-audit.log

{"typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event", "initiator": {"typeURI":
"service/security/account/user", "host": {"address": "iam-pdp.kube-system.svc.cluster.local:7998",
"user-agent": "lua-resty-http/0.11 (Lua) ngx_lua/10013"}, "name": "admin", "credential": {"type":
"token"}}, "kubernetes.namespace": "kube-system", "kubernetes.pod": "auth-pdp-vvtbt", "requestPath":
"/v1/authz", "observer": {"id": "initiator"}, "eventType": "activity", "origination": "cli",
"eventTime": "2019-03-22T16:46:50.851788", "kubernetes.container_id":
"f6ffd930a71856089866433b9f943e3af2b0f638a99d082fd15155f840517b3d", "severity": "normal", "reason":
{"reasonCode": "200", "reasonType": "HTTP"}, "version": "v1.0", "action": "authorize", "outcome":
"success", "id": "icp:1739793c-4cc2-11e9-8903-e6096785c0b3", "kubernetes.container_name": "iam-
policy-decision", "target": {"typeURI": "security/policy", "id":
"f6ffd930a71856089866433b9f943e3af2b0f638a99d082fd15155f840517b3d", "name": "iam-policy-decision"}}
{"typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event", "initiator": {"typeURI":
"service/security/account/user", "host": {"address": "iam-pdp.kube-system.svc.cluster.local:7998",
"user-agent": "lua-resty-http/0.11 (Lua) ngx_lua/10013"}, "name": "admin", "credential": {"type":
"token"}}, "kubernetes.namespace": "kube-system", "kubernetes.pod": "auth-pdp-vvtbt", "requestPath":

"/v1/authz", "observer": {"id": "initiator"}, "eventType": "activity", "origination": "cli", "eventTime": "2019-03-22T16:46:51.085948", "kubernetes.container_id": "f6ffd930a71856089866433b9f943e3af2b0f638a99d082fd15155f840517b3d", "severity": "normal", "reason": {"reasonCode": "200", "reasonType": "HTTP"}, "version": "v1.0", "action": "authorize", "outcome": "success", "id": "icp:175d307a-4cc2-11e9-8903-e6096785c0b3", "kubernetes.container_name": "iam-policy-decision", "target": {"typeURI": "security/policy", "id": "f6ffd930a71856089866433b9f943e3af2b0f638a99d082fd15155f840517b3d", "name": "iam-policy-decision"}} {"typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event", "initiator": {"typeURI": "service/security/account/user", "host": {"address": "iam-pdp.kube-system.svc.cluster.local:7998", "user-agent": "lua-resty-http/0.11 (Lua) ngx_lua/10013"}, "name": "admin", "credential": {"type": "token"}}, "kubernetes.namespace": "kube-system", "kubernetes.pod": "auth-pdp-vvtbt", "requestPath": "/v1/authz", "observer": {"id": "initiator"}, "eventType": "activity", "origination": "cli", "eventTime": "2019-03-22T16:46:55.608122", "kubernetes.container_id": "f6ffd930a71856089866433b9f943e3af2b0f638a99d082fd15155f840517b3d", "severity": "normal", "reason": {"reasonCode": "200", "reasonType": "HTTP"}, "version": "v1.0", "action": "authorize", "outcome": "success", "id": "icp:1a0f3c8c-4cc2-11e9-9566-e6096785c0b3", "kubernetes.container_name": "iam-policy-decision", "target": {"typeURI": "security/policy", "id": "f6ffd930a71856089866433b9f943e3af2b0f638a99d082fd15155f840517b3d", "name": "iam-policy-decision"}}

mutation-advisor-audit.log (anotação de arquivo)

{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:2bd64a33-db41-4c96-9910-7e9ecb7a1b76","action":"update","requestPath":"/opt/ibm/identity-provider/logs/identity_provider.log.1.gz","observer":{"id":"target"},"initiator":{"id":"crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/auth-idp-gk76f/platform-identity-provider/421a6c6850d2e259a91e25b222b06efa4e3e22320820d7f3fbc2a1924675984e","credential":{"type":"container"}},"target":{"id":"crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/auth-idp-gk76f/platform-identity-provider/421a6c6850d2e259a91e25b222b06efa4e3e22320820d7f3fbc2a1924675984e","name":"/opt/ibm/identity-provider/logs/identity_provider.log.1.gz","typeURI":"service/data/file"},"severity":"critical","outcome":"success","eventTime":"2019-05-19T22:26:30.625Z","kubernetes.container_id":"3297527d587232e539315eca371f95c62a8b1a1c2f7ea1f5896444582b7093ad","kubernetes.container_name":"mutation-advisor","kubernetes.pod":"vulnerability-advisor-ma-file-annotator-58c5bdcbd5-5r65p","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:fd44c075-6a2a-46ff-9d11-1c9a8929c8d0","action":"create","requestPath":"/opt/ibm/identity-provider/logs/identity_provider.log.2.gz","observer":{"id":"target"},"initiator":{"id":"crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/auth-idp-gk76f/platform-identity-provider/421a6c6850d2e259a91e25b222b06efa4e3e22320820d7f3fbc2a1924675984e","credential":{"type":"container"}},"target":{"id":"crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/auth-idp-gk76f/platform-identity-provider/421a6c6850d2e259a91e25b222b06efa4e3e22320820d7f3fbc2a1924675984e","name":"/opt/ibm/identity-provider/logs/identity_provider.log.2.gz","typeURI":"service/data/file"},"severity":"critical","outcome":"success","eventTime":"2019-05-19T22:26:30.625Z","kubernetes.container_id":"3297527d587232e539315eca371f95c62a8b1a1c2f7ea1f5896444582b7093ad","kubernetes.container_name":"mutation-advisor","kubernetes.pod":"vulnerability-advisor-ma-file-annotator-58c5bdcbd5-5r65p","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:caf11dcf-8996-481a-b064-478b7ac47c60","action":"delete","requestPath":"/var/lib/prometheus/data/01DB3SHSTH94KTG4RF2KF5M6E1","observer":{"id":"target"},"initiator":{"id":"crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/mcm-prometheus-5894b6655f-nrlbk/prometheus/90daf5abeb0678691284047365429 1d799893a53c7137074e0be11216b848574","credential":{"type":"container"}},"target":{"id":"crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/mcm-prometheus-5894b6655f-nrlbk/prometheus/90daf5abeb0678691284047365429 1d799893a53c7137074e0be11216b848574","name":"/var/lib/prometheus/data/01DB3SHSTH94KTG4RF2KF5M6E1","typeURI":"service/data/file"},"severity":"critical","outcome":"success","eventTime":"2019-05-19T22:28:28.749Z","kubernetes.container_id":"3297527d587232e539315eca371f95c62a8b1a1c2f7ea1f5896444582b7093ad","kubernetes.container_name":"mutation-advisor","kubernetes.pod":"vulnerability-advisor-ma-file-annotator-58c5bdcbd5-5r65p","kubernetes.namespace":"kube-system","origination":"cli","version":"v1.0"}

mutation-advisor-audit.log (anotação de processo)

{"typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event", "initiator": {"credential": {"type": "container"}, "id": "crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/key-management-lifecycle-56b76dc775-dqd4f/icp-audit-

service/17128f445f8a7402f22ed4ff571eea637ac3a6292024e2309a9c83d53fb0cef8"}, "kubernetes.namespace": "kube-system", "kubernetes.pod": "vulnerability-advisor-process-ma-annotator-6966664857-m9xgs", "requestPath": "cron", "observer": {"id": "target"}, "eventType": "activity", "origination": "cli", "eventTime": "2019-05-20T06:26:14+0000", "kubernetes.container_id": "3298edd78b411e2bab1311329da58a85e4a30cf2d844e2b8eceb45e34ab58b0f", "severity": "critical", "version": "v1.0", "action": "create", "outcome": "success", "id": "icp:c231a79c-10a1-4225-a226-51d0c9f070c4", "kubernetes.container_name": "mutation-advisor", "target": {"typeURI": "service/compute/process", "id": "crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/key-management-lifecycle-56b76dc775-dqd4f/icp-audit-service/17128f445f8a7402f22ed4ff571eea637ac3a6292024e2309a9c83d53fb0cef8", "name": "cron"}}
{"typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event", "initiator": {"credential": {"type": "container"}, "id": "crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/think-blue-demo-app-68b77bdc57-dj7lq/icp-audit-service/f6854681151bda2d2335dd6206e632d535ae789fe9f3a0a071f92fe193ed918f"}, "kubernetes.namespace": "kube-system", "kubernetes.pod": "vulnerability-advisor-process-ma-annotator-6966664857-m9xgs", "requestPath": "cron", "observer": {"id": "target"}, "eventType": "activity", "origination": "cli", "eventTime": "2019-05-20T06:35:44+0000", "kubernetes.container_id": "3298edd78b411e2bab1311329da58a85e4a30cf2d844e2b8eceb45e34ab58b0f", "severity": "critical", "version": "v1.0", "action": "delete", "outcome": "success", "id": "icp:81762981-f340-4bed-a193-4bac86085cf6", "kubernetes.container_name": "mutation-advisor", "target": {"typeURI": "service/compute/process", "id": "crn:v1:icp:private:k8:172.16.26.19:n/kube-system::container:kube-system/think-blue-demo-app-68b77bdc57-dj7lq/icp-audit-service/f6854681151bda2d2335dd6206e632d535ae789fe9f3a0a071f92fe193ed918f", "name": "cron"}}

Experimento e estatísticas

Um experimento foi executado em um cluster do IBM Cloud Private Versão 3.1.2 com 1 nó principal, 1 nó de gerenciamento e 2 nós do trabalhador. Um script de segundo plano executou tarefas para acionar os serviços de destino para gerar logs de auditoria. A configuração padrão foi usada para `auth-idp` e `auth-pdp`. Na configuração do consultor de mutação, o tempo de crawler do arquivo foi atualizado de 24 horas para 5 minutos. O crawler do processo não foi mudado. O experimento foi executado por 5 horas para cada serviço. O tamanho dos dados de auditoria foi coletado conforme o experimento era executado.

A tabela a seguir ilustra um padrão de crescimento de tamanho de dados em intervalos de tempo de 15 minutos. Os tamanhos de dados são mostrados em kilobytes (KB).

Tabela 4. Tamanhos de dados de auditoria

Minutes	auth-idp-8jdlx (KB)	auth-pdp-vvtbt (KB)	vulnerability-advisor-ma-file-annotator-b9d746f9-nzpnj (KB)	vulnerability-advisor-process-ma-annotator-5fcf6dccbc-lc7ws (KB)
1	172	80	508	0
15	2400	1100	324000	0
30	4500	2000	473000	0
45	6700	3000	583000	0
60	9000	4000	828000	4
75	12000	5000	965000	4
90	14000	5900	1006000	4
105	16000	6900	1100000	8
120	19000	8000	1100000	8
135	21000	8900	1200000	12
150	23000	9900	1200000	12
165	26000	11000	1300000	12
180	28000	12000	1500000	16
195	30000	13000	1600000	16
210	33000	14000	1600000	16
225	35000	15000	1700000	16
240	37000	16000	1900000	16
255	40000	17000	2000000	16
270	42000	18000	2000000	16
285	43000	19000	2200000	16
300	46000	20000	2300000	16

Painel do Kibana de criação de log de auditoria

Aprenda como incluir painéis customizados no Kibana para que seja possível analisar os logs de auditoria.

É possível incluir ou excluir gráficos de visualização nos painéis. É possível modificar a representação de dados ou o layout. Os exemplos de painel a seguir são fornecidos:

- um painel que está focado nos logs de auditoria `platform services`
- um painel que está focado nos logs de auditoria do `Kubernetes`.

Importando painéis no Kibana

O processo a seguir é aplicável a ambos os painéis. É possível importar um ou ambos os painéis, um de cada vez. Antes de incluir o painel, certifique-se de que o cluster do IBM Cloud Private tenha gerado logs de auditoria e que os logs de auditoria sejam encaminhados para o ELK.

Para obter mais informações sobre como ativar a criação de log de auditoria, consulte Configurando os serviços do IBM Cloud Private para gerar logs de auditoria.

Para obter mais informações sobre como encaminhar logs de auditoria para o ELK, consulte Configurando o IBM Cloud Private para encaminhar logs de auditoria.

Conclua as etapas a seguir para importar um painel no Kibana:

1. Copie o conteúdo do painel a seguir e salve-o em um arquivo `<file-name>.json`.
   - Painel e visualizações de criação de log de auditoria de serviço de plataforma
   - Painel e visualizações de criação de log de auditoria do Kubernetes
2. Abra o console da web do Kibana (no menu de navegação, clique em `Plataforma > Criação de log`)
3. No Kibana, navegue para `Gerenciamento > Objetos salvos`
4. Clique em `Importar` no canto superior direito
5. Localize o arquivo `<file-name>.json` salvo e importe-o
6. É possível localizar o painel importado no menu de navegação do Kibana em `Dashboard`

Painel e visualizações de criação de log de auditoria de serviço de plataforma

```
[
  {
    "_id": "2d1bc1a0-f886-11e8-94e8-63db1f1e8f5c",
    "_type": "dashboard",
    "_source": {
      "title": "audit-logging-dashboard",
      "hits": 0,
      "description": "",
      "panelsJSON": "
[{\"size_x\":6,\"size_y\":5,\"panelIndex\":1,\"type\":\"visualization\",\"id\":\"a3841760-f882-11e8-
94e8-63db1f1e8f5c\",\"col\":1,\"row\":1},
{\"size_x\":6,\"size_y\":5,\"panelIndex\":2,\"type\":\"visualization\",\"id\":\"79f77cf0-1e45-11e9-
bc71-473f395cd7d0\",\"col\":7,\"row\":1},
{\"size_x\":6,\"size_y\":5,\"panelIndex\":3,\"type\":\"visualization\",\"id\":\"ff209890-1e3f-11e9-
9b3c-fbc41e168e2a\",\"col\":1,\"row\":6},
{\"size_x\":6,\"size_y\":5,\"panelIndex\":4,\"type\":\"visualization\",\"id\":\"5e771c90-1e42-11e9-
9b3c-fbc41e168e2a\",\"col\":7,\"row\":6}]",
      "optionsJSON": "{\"darkTheme\":false}",
      "uiStateJSON": "{}",
      "version": 1,
      "timeRestore": false,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"filter\":[{\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}}}],\"highlightAll\":true,\"version\":true}"
      }
    }
  },
  {
    "_id": "a3841760-f882-11e8-94e8-63db1f1e8f5c",
    "_type": "visualization",
    "_source": {
      "title": "audit-logging-team-operations-group-by-actions",
      "visState": "{\"title\":\"audit-logging-team-operations-group-by-
actions\",\"type\":\"pie\",\"params\":
```

```
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"action.keyword\",\"exclude\":\"revoke\",\"size\":10,\"order\":\"desc\",\"orderBy\":\"1\
",\"customLabel\":\"Action\"}}],\"listeners\":{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "79f77cf0-1e45-11e9-bc71-473f395cd7d0",
    "_type": "visualization",
    "_source": {
      "title": "audit-logging-group-by-container-name",
      "visState": "{\"title\":\"audit-logging-group-by-container-name\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"kubernetes.container_name.keyword\",\"exclude\":\"platform-identity-
provider\",\"size\":15,\"order\":\"desc\",\"orderBy\":\"1\",\"customLabel\":\"container-
name\"}}],\"listeners\":{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "ff209890-1e3f-11e9-9b3c-fbc41e168e2a",
    "_type": "visualization",
    "_source": {
      "title": "audit-logging-team operation-group-by-initiator-name",
      "visState": "{\"title\":\"audit-logging-team operation-group-by-initiator-
name\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"initiator.name.keyword\",\"exclude\":\"\\\"\\\"\",\"size\":20,\"order\":\"desc\",\"orde
rBy\":\"1\",\"customLabel\":\"Initiator Name\"}}],\"listeners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "5e771c90-1e42-11e9-9b3c-fbc41e168e2a",
    "_type": "visualization",
    "_source": {
      "title": "audit-logging-group-by-syslog-identifier",
      "visState": "{\"title\":\"audit-logging-group-by-syslog-
identifier\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"syslog_identifier.keyword\",\"size\":5,\"order\":\"desc\",\"orderBy\":\"1\",\"customLab
el\":\"Syslog Identifier\"}}],\"listeners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
```

```
    }
  }
]
```

Painel e visualizações de criação de log de auditoria do Kubernetes

```
[
  {
    "_id": "f9f11070-3b6d-11e9-b42d-ab168fec400a",
    "_type": "dashboard",
    "_source": {
      "title": "k8s-audit-logging-dashboard",
      "hits": 0,
      "description": "",
      "panelsJSON": "
[{\"size_x\":6,\"size_y\":5,\"panelIndex\":1,\"type\":\"visualization\",\"id\":\"ade39170-3ad8-11e9-
a4b7-37e35b0b26d6\",\"col\":1,\"row\":7},
{\"size_x\":6,\"size_y\":3,\"panelIndex\":2,\"type\":\"visualization\",\"id\":\"d4762410-3ad8-11e9-
a4b7-37e35b0b26d6\",\"col\":7,\"row\":4},
{\"size_x\":6,\"size_y\":3,\"panelIndex\":3,\"type\":\"visualization\",\"id\":\"52bdd3f0-3ad8-11e9-
a4b7-37e35b0b26d6\",\"col\":1,\"row\":12},
{\"size_x\":6,\"size_y\":3,\"panelIndex\":4,\"type\":\"visualization\",\"id\":\"788b08d0-3ada-11e9-
a4b7-37e35b0b26d6\",\"col\":7,\"row\":7},
{\"size_x\":6,\"size_y\":3,\"panelIndex\":5,\"type\":\"visualization\",\"id\":\"04fb0f20-3ad8-11e9-
a4b7-37e35b0b26d6\",\"col\":1,\"row\":1},
{\"size_x\":6,\"size_y\":12,\"panelIndex\":6,\"type\":\"visualization\",\"id\":\"20dfe890-3ad9-11e9-
a4b7-37e35b0b26d6\",\"col\":7,\"row\":10},
{\"size_x\":6,\"size_y\":3,\"panelIndex\":7,\"type\":\"visualization\",\"id\":\"9ec2d8a0-3ad7-11e9-
a4b7-37e35b0b26d6\",\"col\":1,\"row\":4},
{\"size_x\":6,\"size_y\":7,\"panelIndex\":8,\"type\":\"visualization\",\"id\":\"d4d5b3c0-3ad9-11e9-
a4b7-37e35b0b26d6\",\"col\":7,\"row\":22},
{\"size_x\":6,\"size_y\":6,\"panelIndex\":9,\"type\":\"visualization\",\"id\":\"7e8b3c50-3ad5-11e9-
a4b7-37e35b0b26d6\",\"col\":1,\"row\":15},
{\"size_x\":6,\"size_y\":8,\"panelIndex\":10,\"type\":\"visualization\",\"id\":\"bdd21e60-3ad5-11e9-
a4b7-37e35b0b26d6\",\"col\":1,\"row\":21},
{\"size_x\":6,\"size_y\":3,\"panelIndex\":11,\"type\":\"visualization\",\"id\":\"c91e1360-3ad4-11e9-
a4b7-37e35b0b26d6\",\"col\":7,\"row\":1}]",
      "optionsJSON": "{\"darkTheme\":false}",
      "uiStateJSON": "{\"P-10\":{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}},\"P-8\":
{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}},\"P-1\":{\"spy\":{\"mode\":
{\"name\":null,\"fill\":false}}},\"P-6\":{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}},\"P-
9\":{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}}}",
      "version": 1,
      "timeRestore": false,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"filter\":[{\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}}}],\"highlightAll\":true,\"version\":true}"
      }
    }
  },
  {
    "_id": "ade39170-3ad8-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-api-group",
      "visState": "{\"title\":\"k8s-audit-group-by-api-group\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"objectRef.apiGroup.keyword\",\"size\":115,\"order\":\"desc\",\"orderBy\":\"1\",\"custom
Label\":\"api-group\"}}],\"listeners\":{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":\"table\",\"fill\":false}}}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "d4762410-3ad8-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
```

```
      "title": "k8s-audit-group-by-api-version",
      "visState": "{\"title\":\"k8s-audit-group-by-api-version\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"objectRef.apiVersion.keyword\",\"size\":15,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"li
steners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "52bdd3f0-3ad8-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-log-level",
      "visState": "{\"title\":\"k8s-audit-group-by-log-level\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"level.keyword\",\"size\":15,\"order\":\"desc\",\"orderBy\":\"1\",\"customLabel\":\"log-
level\"}}],\"listeners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "788b08d0-3ada-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-log-stage",
      "visState": "{\"title\":\"k8s-audit-group-by-log-stage\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"stage.keyword\",\"size\":5,\"order\":\"desc\",\"orderBy\":\"1\",\"customLabel\":\"stage
\"}}],\"listeners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "04fb0f20-3ad8-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-namespace",
      "visState": "{\"title\":\"k8s-audit-group-by-namespace\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"kubernetes.namespace.keyword\",\"size\":15,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"li
steners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
```

```
  },
  {
    "_id": "20dfe890-3ad9-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-resources",
      "visState": "{\"title\":\"k8s-audit-group-by-resources\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"objectRef.resource.keyword\",\"size\":115,\"order\":\"desc\",\"orderBy\":\"1\",\"custom
Label\":\"resources\"}}],\"listeners\":{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":\"table\",\"fill\":false}}}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "9ec2d8a0-3ad7-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-response-status-code",
      "visState": "{\"title\":\"k8s-audit-group-by-response-status-
code\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"responseStatus.code\",\"size\":15,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"listeners\"
:{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "d4d5b3c0-3ad9-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-source-ip",
      "visState": "{\"title\":\"k8s-audit-group-by-source-ip\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"sourceIPs.keyword\",\"size\":115,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"listeners\":
{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":\"table\",\"fill\":false}}}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "7e8b3c50-3ad5-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-user",
      "visState": "{\"title\":\"k8s-audit-group-by-user\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"user.username.keyword\",\"size\":125,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"listener
s\":{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":null,\"fill\":false}}}",
      "description": "",
```

```
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "bdd21e60-3ad5-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "k8s-audit-group-by-user-agent",
      "visState": "{\"title\":\"k8s-audit-group-by-user-agent\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"userAgent.keyword\",\"size\":115,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"listeners\":
{}}",
      "uiStateJSON": "{\"spy\":{\"mode\":{\"name\":\"table\",\"fill\":false}}}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  },
  {
    "_id": "c91e1360-3ad4-11e9-a4b7-37e35b0b26d6",
    "_type": "visualization",
    "_source": {
      "title": "kube-audit-group-by-verb",
      "visState": "{\"title\":\"kube-audit-group-by-verb\",\"type\":\"pie\",\"params\":
{\"addTooltip\":true,\"addLegend\":true,\"legendPosition\":\"right\",\"isDonut\":false},\"aggs\":
[{\"id\":\"1\",\"enabled\":true,\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"2\",\"enabled\":true,\"type\":\"terms\",\"schema\":\"segment\",\"params\":
{\"field\":\"verb.keyword\",\"size\":15,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"listeners\":{}}",
      "uiStateJSON": "{}",
      "description": "",
      "version": 1,
      "kibanaSavedObjectMeta": {
        "searchSourceJSON": "{\"index\":\"audit-*\",\"query\":{\"query_string\":
{\"query\":\"*\",\"analyze_wildcard\":true}},\"filter\":[]}"
      }
    }
  }
]
```

Usando seu painel do Kibana

Os recursos do Kibana a seguir são úteis para agrupar tipos de logs semelhantes, supervisionar as estatísticas de cada tipo de log e aplicar filtros para localizar os logs desejados. Navegue para `Painel > <imported-dashboard>`.

- É possível visualizar vários gráficos de pizza representando logs de auditoria agrupados de modo diferente. Cada campo no gráfico é representado com uma cor exclusiva.
- Passe o mouse sobre um gráfico para visualizar a contagem e os nomes dos campos.
- Clique em qualquer campo para visualizar qual filtro é aplicado.
- Passe o mouse sobre um filtro para visualizar as opções disponíveis. Por exemplo:
  - Opção `pin`. É possível fixar ('pin') o filtro para utilizá-lo no `Discover`. Se você fixar (`pin`) o filtro e navegar para `Discover`, verá que o filtro está aplicado. Isso ajuda a localizar logs brutos para filtros específicos.
  - opção para incluir ou excluir filtros
  - opção para remover a opção de filtro
- É possível aplicar diversos filtros.

  Para obter mais informações sobre os casos de uso de visualização e de painel, consulte https://www.elastic.co/guide/en/kibana/current/visualize.html.

Resolução de problemas

Erro de importação do painel

- Verifique se seu cluster do IBM Cloud Private está gerando logs de auditoria e encaminhando-os para o ELK.
- Verifique se o padrão de índice `audit-*` foi criado:
  - Abra o console da web do Kibana
  - Navegue para `Gerenciamento > Padrões de índice`
  - Clique em `Criar padrão de índice`
  - Configure o nome ou o padrão do índice como `audit-*`. Mantenha o nome do campo `Time Filter` como `@timestamp`.
- Atualize a lista de campos de índice `audit-*`.
  - Abra o console da web do Kibana.
  - Navegue para `Gerenciamento > Padrões de índice`.
  - Clique no padrão de índice `audit-*`.
  - Clique em `Atualizar lista de campo`.

Integração de criação de log de auditoria do IBM Cloud Private

com ferramentas do SIEM corporativas

O IBM Cloud Private fornece recursos de log de auditoria.

O IBM Cloud Private fornece uma nuvem privada baseada em Kubernetes que pode ser implementada e gerenciada por um cliente dentro de sua empresa. O IBM Cloud Private permite que os desenvolvedores de aplicativos transformem seus aplicativos corporativos para usar recursos de nuvem, como a arquitetura de elasticidade e de microsserviços. As equipes corporativas de segurança, conformidade e risco precisam de vários controles de segurança implementados dentro do IBM Cloud Private para obedecer as políticas de segurança e de conformidade corporativas que atendam aos requisitos de auditoria e regulamentares internos e externos.

Um dos controles de segurança que precisa atender a esses requisitos é a criação de log de auditoria. Especificamente, o IBM Cloud Private precisa ser configurado para gerar logs de auditoria. Esses logs de auditoria precisam ser roteados para a ferramenta SIEM corporativa existente do cliente para gerenciamento de incidente de segurança pelo centro de operações de segurança do cliente. Essas informações esboçam como o IBM Cloud Private pode ser configurado para gerar vários logs de auditoria e como esses logs podem ser roteados para o SIEM de um cliente usando o IBM QRadar como um exemplo.

Log de auditoria de amostra


```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    "eventType": "activity",
    "id": "icp:f14704b0-a9dd-11e8-817b-89bcae80625c",
    "action": "read",
    "requestPath": "/identity/api/v1/directory/ldap/56027bb0-a9dd-11e8-9573-
0b442b44e932/fetchUsergroups?searchString=%2Ac%2A",
    "initiator": {
        "typeURI": "service/security/account/user",
        "name": "admin",
        "credential": {
            "type": "token"
        },
        "host": {
            "address": "icp-management-ingress:8443"
        }
    },
    "target": {
        "id": "6917371c373f3eb2098a9d7bcd5026052a1c665721a80596c74295ddd9f39ee9\n",
        "name": "platform-identity-management",
        "typeURI": "service/storage/directory"
    },
    "observer": {
        "id": "target"
    },
    "severity": "normal",
    "outcome": "success",
    "reason": {
        "reasonType": "HTTP",
        "reasonCode": 200
    },
```

```
    "eventTime": "2018-08-27T09:45:33.563Z",
    "kubernetes.container_id": "6917371c373f3eb2098a9d7bcd5026052a1c665721a80596c74295ddd9f39ee9\n",
    "kubernetes.container_name": "platform-identity-management",
    "kubernetes.pod": "auth-idp-vfmzh",
    "kubernetes.namespace": "kube-system",
    "origination": "cli",
    "version": "v1.0"
}
```


- Configurando o IBM Cloud Private para gerar vários logs de auditoria
- Configurando o IBM Cloud Private para encaminhar logs de auditoria
- Integrando o IBM Cloud Private ao IBM QRadar
- Integrando o IBM Cloud Private ao Splunk

Configurando os serviços do IBM Cloud Private para gerar logs de auditoria

É possível configurar seu IBM® Cloud Private para gerar logs de auditoria.

O IBM Cloud Private fornece dois tipos de logs de auditoria: um, `icp-audit` gerado pelos serviços de plataforma IBM Cloud Private, e dois, `kube-audit`, gerado pelo Kubernetes. É possível ativar ou desativar somente `kube-audit` durante a instalação de seu cluster, atualizando o arquivo `config.yaml`. É possível ativar ou desativar ambos os tipos de logs de auditoria após a instalação. Por padrão, os serviços de plataforma não geram nenhum log de auditoria. Deve-se ativar a criação de log de auditoria para cada serviço para o qual você precisa dos logs.

O IBM Cloud Private gera logs `icp-audit` para todas as operações de criação, leitura, atualização e exclusão. Não é possível configurar o nível de log para logs `icp-audit`. No entanto, é possível configurar o nível de log para logs `kube-audit` atualizando o arquivo `audit-policy.yaml`.

Ativando e desativando a criação de log de auditoria para vários serviços do

IBM Cloud Private

1. Navegue para ConfigMap: **Menu de navegação** > **Configuração** > **ConfigMap**.
2. Procure o ConfigMap do serviço para o qual a criação de log de auditoria precisa ser ativada.
3. Clique em **Editar**.
4. Configure a chave relacionada à auditoria como `true` ou `false` para ativar ou desativar a criação de log de auditoria para esse serviço.
5. Clique em **Enviar**.
6. Remova todos os pods que pertencem a esse serviço. Os pods são recriados com a auditoria ativada ou desativada. Os serviços podem ser localizados em DaemonSets: **Menu de navegação**> **Carga de trabalho** > **DaemonSets** ou em Implementações: **Menu de navegação** > **Carga de trabalho** > **Implementações**.

A tabela a seguir lista os serviços do IBM Cloud Private e os ConfigMaps nos quais as chaves relacionadas à auditoria são configuradas.

Nome do serviço	ConfigMap	Chave	Local do pod
`auth-idp` (Logs de auditoria de autenticação)	`platform-auth-idp`	`AUDIT_ENABLED_IDMGMT`, `AUDIT_ENABLED_IDPROVIDER`	`Menu de navegação > Carga de trabalho > DaemonSets > auth-idp`
`auth-pdp` (Registros de auditoria de autorização)	`auth-pdp`	`AUDIT_ENABLED`	`Menu de navegação > Carga de trabalho > DaemonSets > auth-pdp`
`auth-pap` (Logs de auditoria do ponto de administração de política)	auth-pap	AUDIT_ENABLED	Menu de navegação > Carga de trabalho > DaemonSets > auth-pap
platform-api	platform-api	AUDIT_ENABLED	Menu de navegação > Carga de trabalho > Implementações > platform-api
helm-api	helm-api	AUDIT_ENABLED	Menu de navegação > Carga de trabalho > Implementações > helm-api
helm-repo	helm-repo	AUDIT_ENABLED	Navigation Menu > Workload > Deployments > helm-repo
vulnerability-advisor-ma-file-annotator	vulnerability-advisor-audit-config	MUTATION_ADVISOR_AUDIT_ENABLED, SAS_API_SERVER_AUDIT_ENABLED	Menu de navegação > Carga de trabalho > Implementações > vulnerability-advisor-ma-file-annotator
vulnerability-advisor-ma-process-annotator	vulnerability-advisor-audit-config	MUTATION_ADVISOR_AUDIT_ENABLED, SAS_API_SERVER_AUDIT_ENABLED	Menu de navegação > Carga de trabalho >

Implementações > vulnerability-advisor-ma-process-annotator |
| key-management-lifecycle | lifecycle-config | AUDIT_ENABLED | Menu de navegação > Carga de trabalho > Implementações > key-management-lifecycle ` |

Nota: alguns serviços do IBM Cloud Private, como `auth-idp`, `auth-pdp` e `auth-pap`, podem gerar dados de auditoria mais detalhados. Essa criação de log detalhada pode ser ativada configurando a chave `AUDIT_DETAIL` como true no respectivo ConfigMap.

Ativando a auditoria do Kubernetes

A auditoria do Kubernetes (`kube-audit`) pode ser ativada atualizando o arquivo `master.json`.

1. Use o Shell Seguro (SSH) para conectar-se ao nó principal como um usuário raiz.
2. Copie o arquivo `master.json` no local `tmp`.

   ```
   cp /etc/cfc/pods/master.json /tmp/
   ```

3. Edite o arquivo `master.json` copiado usando qualquer editor.

   ```
   vim /tmp/master.json
   ```

4. Inclua `audit policy file path` e `audit log file path` na seção de configuração `apiserver` na lista `command` após o último elemento.

   ```
   "--audit-policy-file=/etc/cfc/conf/audit-policy.yaml",
   "--audit-log-path=/var/log/k8saudit/audit.log",
   "--audit-log-maxage=3",
   "--audit-log-maxbackup=10",
   "--audit-log-maxsize=10"
   ```

   Nota: Coloque uma vírgula "," após o último elemento da lista de comandos antes de incluir os dois campos anteriores.

5. Substitua o `master.json` original pelo atualizado.

   ```
   cp /tmp/master.json /etc/cfc/pods/master.json
   ```

6. O pod principal seleciona as mudanças e o kube-apiserver é reiniciado com a auditoria ativada.

Para obter informações adicionais, consulte Auditando o .

Desativando a auditoria do Kubernetes

A auditoria do Kubernetes (`kube-audit`) pode ser desativada atualizando o arquivo `master.json`.

1. Use SSH para se conectar ao nó principal como um usuário raiz.
2. Copie o arquivo `master.json` no local `tmp`.

   ```
   cp /etc/cfc/pods/master.json /tmp/
   ```

3. Edite o arquivo `master.json` copiado usando qualquer editor.

   ```
   vim /tmp/master.json
   ```

4. Remova `"--audit-policy-file=/etc/cfc/conf/audit-policy.yaml"` da seção de configuração `apiserver`.
   Nota: Para reativar `kube-audit`, siga todas as etapas na seção Ativando a auditoria do Kubernetes.

5. Substitua o `master.json` original por um atualizado.

   ```
   cp /tmp/master.json /etc/cfc/pods/master.json
   ```

6. O pod principal seleciona as mudanças e o `kube-apiserver` é reiniciado com a auditoria desativada.

Para obter informações adicionais, consulte Política de auditoria .

Configurando o IBM Cloud Private para encaminhar logs de auditoria

É possível ativar a criação de log de auditoria para serviços individuais para encaminhar seus logs de auditoria para o ELK ou SIEM.

Para obter mais informações sobre como gerar logs de auditoria, consulte Configurando os serviços do IBM Cloud Private para gerar logs de auditoria.

Ativando e desativando o encaminhamento para criação de log de auditoria

Por padrão, o encaminhamento está desativado. Cada plug-in possui um ConfigMap separado. Consulte a tabela a seguir para obter mais informações sobre ConfigMaps de criação de log de auditoria:

ConfigMap	Descrição
audit-logging-fluentd-ds-config	Este ConfigMap é o ConfigMap primário para criação de log de auditoria. Os

plug-ins de origem e os plug-ins de saída são importados neste ConfigMap.| |audit-logging-fluentd-ds-source-config| ConfigMap de plug-in de origem| |audit-logging-fluentd-ds-elk-config| ConfigMap de plug-in de saída ELK| |audit-logging-fluentd-ds-remote-syslog-config| ConfigMap de plug-in de saída do IBM QRadar| |audit-logging-fluentd-ds-splunk-hec-config| ConfigMap de saída do Splunk|

Ative e desative o encaminhamento para criação de log de auditoria a partir do console de gerenciamento com as etapas a seguir:

1. Efetue login em seu cluster do IBM® Cloud Private.
2. No menu de navegação, clique em **Configuração** > **ConfigMap**.
3. Selecione o ConfigMap `audit-logging-fluentd-ds-config`.
4. Clique no ícone **Abrir e fechar opções** e clique em **Editar**.
5. Ative o encaminhamento para a criação de log de auditoria configurando o valor de parâmetro `ENABLE_AUDIT_LOGGING_FORWARDING` como `true`.
6. Desative o encaminhamento para a criação de log de auditoria configurando o valor de parâmetro `ENABLE_AUDIT_LOGGING_FORWARDING` como `false`. Se você desativar o encaminhamento, ignore a etapa 7.
7. Encaminhe seus logs de auditoria para o ELK ou SIEM.

   Nota: há um arquivo de configuração de plug-in de entrada e múltiplos arquivos de configuração de plug-in de saída em seu ConfigMap. Certifique-se de usar apenas um plug-in de saída por vez.

   - Edite o arquivo `audit-logging-fluentd-ds-config`. Remova o comentário `@include /fluentd/etc/elk.conf` a partir do parâmetro `fluent.conf` para encaminhar para o ELK. Deve-se manter outros plug-ins de saída comentados. **Nota**: assegure-se de que o serviço `logging` do IBM Cloud Private esteja implementado.
   - Edite o arquivo `audit-logging-fluentd-ds-config` para encaminhar logs de auditoria para o IBM QRadar com o SIEM removendo o comentário `@include /fluentd/etc/remoteSyslog.conf`. Deve-se manter outros plug-ins de saída comentados.
     - Edite o `audit-logging-fluentd-ds-remote-syslog-config` e inclua as informações a seguir para IBM QRadar com o SIEM: nome do host do servidor, número da porta e identificador de log do IBM QRadar. Para obter mais informações sobre como atualizar os arquivos `audit-logging-fluentd-ds` e `audit-logging-fluentd-ds-remote-syslog-config`, consulte Configurando o cluster do IBM Cloud Private para enviar logs de auditoria sobre TLS para o IBM QRadar.
   - Edite o arquivo `audit-logging-fluentd-ds-config` para encaminhar para o Splunk removendo o comentário do `@include /fluentd/etc/splunkHEC.conf`. Deve-se manter outros plug-ins de saída comentados.
     - Edite o `audit-logging-fluentd-ds-splunk-hec-config` e inclua as informações a seguir para o Splunk: nome do host do servidor, número da porta e token do HEC do Splunk. Para obter mais informações para atualizar os arquivos `audit-logging-fluentd-ds` e `audit-logging-fluentd-ds-splunk-hec-config`, consulte Integrando o IBM Cloud Private com o Splunk.
8. Clique em **Enviar**
9. Remova todos os pods do Daemonset `audit-logging-fluentd-ds`. Seus pods são recriados automaticamente.
   - Remova os pods da console de gerenciamento:
     1. Efetue login em seu cluster do IBM Cloud Private.

2. No menu de navegação, clique em **Carga de trabalho** > **Daemonsets**.
3. Localize e clique no Daemonset `audit-logging-fluentd-ds`.
4. Na seção *Pods*, exclua cada pod clicando no ícone **Abrir e fechar opções de lista**.
5. Clique em **Remover**.

- Remova os pods com a CLI do Kubernetes executando o comando a seguir:

```
kubectl get pod -n kube-system -o wide | grep audit-logging-fluentd-ds- | awk '{print $1}' | xargs kubectl delete pod -n kube-system
```

Integrando o IBM Cloud Private com o IBM QRadar

Integrar o IBM Cloud Private com o IBM QRadar.

- Extensão de origem de log do IBM QRadar para analisar logs de auditoria do IBM Cloud Private
- Configurando o IBM QRadar para receber logs de auditoria do IBM Cloud Private sobre TLS
- Configurando o cluster do IBM Cloud Private para enviar logs de auditoria sobre TLS para o IBM QRadar
- Configurando regras do IBM QRadar

Extensão de origem de log do IBM QRadar para analisar logs de auditoria

do IBM Cloud Private

É possível mapear eventos de auditoria do IBM Cloud Private para o modelo de evento do IBM QRadar usando o editor DSM (Módulo de Suporte de Dispositivo).

O tipo de origem de log é usado para analisar logs de auditoria. É possível incluir propriedades customizadas para analisar campos customizados. Para obter uma lista de propriedades customizadas, consulte Propriedades customizadas para analisar registros de auditoria do IBM Cloud Private.

Nota: se você já configurou um tipo de origem de log, não será necessário concluir as tarefas que estão nas seções a seguir. É possível continuar com Configurando o IBM QRadar para receber logs de auditoria do IBM Cloud Private sobre o TLS.

Criar tipo de origem de log

1. Navegue para o Editor DSM **Admin** > **Origens de Dados** > **Editor DSM**.
2. Clique em **Criar novo** para criar um novo tipo de origem de log.
3. Insira um nome para o novo tipo de origem de log e salve-o.

Inclua propriedades customizadas no tipo de origem de log

1. Use o registro de auditoria do IBM Cloud Private de amostra a seguir para extrair campos e incluir propriedades customizadas. IBM QRadar


```
{"typeURI":"http://schemas.dmtf.org/cloud/audit/1.0/event","eventType":"activity","id":"icp:db4217b0-f274-11e8-a8f9-51a9a7260dca","action":"create","requestPath":"/identity/api/v1/directory/ldap/ddd46230-e77a-11e8-92af-2773a9077558/importUserGroups","initiator":{"typeURI":"service/security/account/user","name":"admin","credential":{"type":"token"},"host":{"user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0 Safari/605.1.15","address":"icp-management-ingress:8443"}},"target":{"id":"07035ecfb9a2aeab68826ae643f4352a8e016e0c89c17185b61e78e7d4574235","name":"ddd46230-e77a-11e8-92af-2773a9077558","actions":"cn=testgroup,ou=groups,dc=ibm,dc=com","typeURI":"service/storage/directory"},"observer":{"id":"target"},"severity":"normal","outcome":"success","reason":{"reasonType":"HTTP","reasonCode":200},"eventTime":"2018-11-27T18:47:14.347Z","kubernetes.container_id":"07035ecfb9a2aeab68826ae643f4352a8e016e0c89c17185b61e78e7d4574235","kubernetes.container_name":"platform-identity-management","kubernetes.pod":"auth-idp-zxqbm","kubernetes.namespace":"kube-system","origination":"ui","version":"v1.0"}
```


2. Clique no ícone de edição.

3. Cole o registro de auditoria do IBM Cloud Private de amostra na área de trabalho e salve-o.
4. Clique em **Incluir** para incluir propriedades customizadas.
5. Se uma propriedade não foi criada anteriormente, clique em **Criar nova**.
6. Insira o nome da propriedade e selecione o tipo de campo apropriado. Inclua uma breve descrição e salve-a.
7. Selecione todas as propriedades necessárias e inclua-as no Tipo de Origem de Log.

Configure as propriedades

Os registros de auditoria do IBM Cloud Private estão no formato JSON. As propriedades podem usar o `Tipo de expressão` como `JSON`.

Edite a configuração de propriedade

1. Selecione uma propriedade que precisa ser configurada, por exemplo, Nome do Pod.
2. Selecione o Tipo de Expressão JSON na lista.
3. Especifique a expressão JSON.
4. Mantenha `Ativado` selecionado.
5. Inclua múltiplas expressões, se necessário.

Propriedades customizadas para analisar registros de auditoria

do IBM Cloud Private

Tabela 1. Propriedades customizadas para analisar registros de auditoriaIBM Cloud Private

Propriedades	Tipo de Expressão	Expression
`Nome do Contêiner`	`JSON`	`/"kubernetes.container_name"`
`Nome do Pod`	`JSON`	`/"kubernetes.pod"`
`requestPath`	`JSON`	`/"requestPath"`
`Categoria de Evento`	`JSON`	`/"eventType"`
`ID do Evento`	1. `JSON` 2. `Regex`	1 . `/"action"` 2. `\"outcome\":\"success\"`
`Nome do Host de Identidade`	`JSON`	`/"initiator"/"host"/"address"`
`Horário da Origem de Log`	`JSON`	`/"eventTime", Date Format = yyyy-MM-dd'T'HH:mm:ss'Z'`
`namespace`	`JSON`	`/"kubernetes.namespace"`
`Resultado`	`JSON`	`/"outcome"`
`IP de Origem`	`JSON`	`/"initiator"/"host"/"address"`
`Nome do Destino`	`JSON`	`/"target"/"name"`
`Nome do Usuário`	`JSON`	`/"initiator"/"name"`

Mantenha as propriedades a seguir no estado em que se encontram.

Propriedades	Propriedades	Propriedades
`MAC de Destino`	`Porta de Destino`	`` `Campo ``

Estendido de Identidade |
Nome do Grupo de Identidades	IP de Identidade	IPv6 de Identidade
MAC de Identidade	Nome do BIOS de Rede de Identidade	Destino IPv6
IP de Destino pós-NAT	Porta de Destino pós-NAT	IP de Origem pós-NAT
Porta de Origem Pós-NAT	IP de Destino Pré-NAT	Porta de Destino Pré-NAT
IP de Origem Pré-NAT	Porta de Origem Pré-NAT	Protocolo
MAC de Origem	Porta de Origem	Origem do IPv6
IP de Destino`		

Configurando o IBM QRadar para receber logs de auditoria do

IBM Cloud Private sobre TLS

É possível configurar o IBM QRadar para receber logs de auditoria do IBM Cloud Private sobre o TLS.

Inclua uma origem de log para receber eventos do IBM Cloud Private

1. Navegue para a guia **Administrador**.
2. Clique em **Origens de log**.
3. Clique em **Incluir** para incluir uma nova origem de log.
4. Especifique os valores para os parâmetros a seguir na janela de configuração da origem de log.
5. `Nome da Origem de Log` Um nome exclusivo para a origem de log
6. `Descrição da Origem de Log` Uma breve descrição da origem de log
7. `Tipo de Origem de Log` O tipo de origem de log que foi criado anteriormente para analisar logs de auditoria
8. `Configuração de Protocolo` Selecione `Syslog TLS` na lista suspensa
9. `Identificador da Origem de Log` Um identificador exclusivo para a origem de log
10. `Porta de Recebimento do TLS` Deve ser exclusiva que, por padrão, é 6514 para TLS
11. `Modo de Autenticação` Selecione `TLS` na lista suspensa
12. `Tipo de Certificado` Selecione `Fornecer certificado` na lista suspensa
13. `Caminho do Certificado do Servidor Fornecido` Insira o caminho absoluto para o certificado do servidor. Por exemplo, `/opt/qradar/conf/trusted_certificates/server_cert`
14. `Caminho da Chave Privada Fornecido` Insira o caminho absoluto para a chave privada. Por exemplo, `/opt/qradar/conf/trusted_certificates/private_server_key.der`
15. Na guia **Administrador**, clique em **Avançado** > **Implementar configuração completa**.
16. Clique em **Continuar**.

Gerar certificados autoassinados

IBM QRadar O syslog TLS precisa de chaves públicas e privadas no formato adequado. Um par de chaves privadas customizadas deve estar no formato PKCS8 codificado em DER.

Nota: restrinja o uso de `ca cert`. Ele é usado para a origem de log TLS.

O processo a seguir cria `private_key.der` e `public_key.pem`.

- `public_key.pem` pode ser usado como um certificado do servidor e o Fluentd (cliente) usará como um certificado do cliente para enviar logs sobre TLS (somente para prova de conceito)
- `private_key.der` pode ser usado como uma chave privada
- `openssl genrsa -out /tmp/private_key.pem 2048`
- `openssl pkcs8 -topk8 -inform PEM -outform DER -in /tmp/private_key.pem -out /tmp/private_key.der -nocrypt`
- `openssl req -new -key /tmp/private_key.pem -out /tmp/csr.pem`
- `openssl req -x509 -sha512 -days 365 -in /tmp/csr.pem -key /tmp/private_key.der -keyform DER -out /tmp/public_key.pem`

Nota: o campo de nome comum é importante. Use o nome do host do servidor IBM QRadar.

Configurando o cluster do IBM Cloud Private para enviar logs de auditoria

sobre TLS para o IBM QRadar

É possível configurar seu cluster do IBM Cloud Private para enviar logs de auditoria sobre TLS para o IBM QRadar.

Por padrão, o encaminhamento para logs de auditoria está desativado. Atualize o mapa de configuração do `audit-logging-fluentd-ds-config` para encaminhar logs de auditoria para o IBM QRadar. O mapa de configuração do `audit-logging-fluentd-ds-config` define a configuração do Fluentd.

Inclua o certificado do servidor IBM QRadar no segredo

audit-elk-certs

Nota: todas as origens de log que possuem certificados do servidor IBM QRadar podem enviar logs para o IBM QRadar. Os certificados do servidor do IBM QRadar devem ter acesso restrito.

1. Converta o certificado do servidor IBM QRadar em `base64` executando o comando a seguir:

   ```
   cat public_key.pem | base64 -w 0
   ```

2. No menu de navegação, clique em **Configuração** > **Segredos**.
3. Clique no ícone **Abrir e fechar lista de opções** para o `audit-elk-certs`.
4. Clique em **Editar**.
5. Edite o arquivo JSON secreto `audit-elk-certs`. Inclua `"qradar.crt"` como um parâmetro e inclua o certificado do servidor IBM QRadar como um valor para `base64`.
6. Clique em **Enviar**.

Atualizando os arquivos ConfigMap *audit-logging-fluentd-ds-config* e

audit-logging-fluentd-ds-remote-syslog

Atualizando o arquivo ConfigMap *audit-logging-fluentd-ds-config*

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
3. Clique no ícone **Abrir e fechar lista de opções** para o arquivo `audit-logging-fluentd-ds-config`.
4. Clique em **Editar**.
5. Ative o encaminhamento para a criação de log de auditoria configurando o valor de parâmetro `ENABLE_AUDIT_LOGGING_FORWARDING` como `true`.
6. Remova o comentário `@include /fluentd/etc/remoteSyslog.conf` para encaminhar para o IBM QRadar com o SIEM. Deve-se manter outros plug-ins de saída comentados.
7. Clique em **Enviar**.

Atualizando o arquivo ConfigMap *audit-logging-fluentd-ds-remote-syslog*

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps**.
3. Clique no ícone **Abrir e fechar lista de opções** para o arquivo `audit-logging-fluentd-ds-remote-syslog`.
4. Clique em **Editar**.
5. Inclua valores para os campos a seguir: nome do host, número da porta e identificador de origem de log do servidor QRadar.
6. Clique em **Enviar**.

Removendo pods do *audit-logging-fluentd-ds-** do fluentd

Remova todos os pods fluentd do Daemonset `audit-logging-fluentd-ds` depois de atualizar os ConfigMaps.

- Conclua as etapas a seguir para remover os pods do Daemonset `audit-logging-fluentd-ds` a partir da console de gerenciamento:

  1. Efetue login em seu console de gerenciamento.
  2. A partir do menu de navegação, clique em **Carga de Trabalho** > **DaemonSets**.
  3. Clique no arquivo `audit-logging-fluentd-ds`.
  4. Na seção *Pods*, remova os pods clicando no ícone **Abrir e fechar lista de opções**.
  5. Clique em **Remover**.

- Para remover os pods do Daemonset `audit-logging-fluentd-ds` a partir da CLI do Kubernetes, execute o comando a seguir:

  ```
  kubectl get pod -n kube-system -o wide | grep audit-logging-fluentd-ds- | awk '{print $1}' |
  xargs kubectl delete pod -n kube-system
  ```

Inclua hostAliases no arquivo de implementação do Daemonset *audit-logging-fluentd-ds*

1. Edite a implementação do `audit-logging-fluentd-ds` executando o comando a seguir:

   ```
   kubectl -n kube-system edit daemonset audit-logging-fluentd-ds
   ```

2. Inclua o mapeamento de nome do host e de endereço IP do servidor IBM QRadar no arquivo de implementação. Seu arquivo `audit-logging-fluentd-ds` pode ser semelhante ao conteúdo a seguir:

```
hostAliases:
   - hostnames:
     - <server machine name>.ibm.com
     ip: a.b.c.d
```

3. Salve o arquivo editado.

Os `hostAliases` para o arquivo de implementação do Daemonset `audit-logging-fluentd-ds` são incluídos.

Configurando as regras do IBM QRadar

Crie regras do IBM QRadar para acionar alertas em eventos importantes.

Conclua as etapas a seguir para criar as regras:

1. Selecione e abra o evento de destino.
2. Localize algumas propriedades exclusivas do evento de destino. Por exemplo, QID é exclusivo para um tipo de evento específico.
3. Navegue para as regras: **Ofensas** > **Regras**.
4. Clique em **Ação** > **Nova regra de evento**.
5. Forneça um nome exclusivo para a regra na seção **Aplicar**.
6. Inclua as condições exclusivas apropriadas em uma regra para acionar o evento desejado.
7. Clique em **Avançar**.
8. Aplique a Ação de Regra, a Resposta de Regra e o Limitador de Resposta.
9. Clique em **Avançar** para revisar a regra.
10. Clique em **Concluir**.
11. Localize todas as regras criadas na guia **Regras**. É possível colocar as regras criadas em um grupo diferente.

Integrando o IBM Cloud Private com o Splunk

Se você usar o Splunk Enterprise, será possível integrar seus logs de auditoria do IBM Cloud Private com o Splunk.

Configurando o cluster do IBM Cloud Private (cliente)

Inclua o certificado de CA do Splunk no segredo `audit-elk-certs` para

IBM Cloud Private

1. Converta o certificado de CA do Splunk no formato de agrupamento 0 `base64` usando o comando a seguir:

```
cat splunkCA.pem | base64 -w 0
```

2. No menu de navegação, clique em **Configuração > Segredos**.
3. Selecione `audit-elk-certs`. Clique em **Editar**.
4. O segredo `audit-elk-certs` está no formato JSON. Inclua "`splunkCA.pem`" como uma chave e a versão codificada em base64 do certificado de CA do Splunk como o valor.
5. Clique em **Enviar**.

Atualize os arquivos `ConfigMap` `audit-logging-fluentd-ds-config` e

`audit-logging-fluentd-ds-splunk-hec-config` para IBM Cloud Private

O arquivo `ConfigMap` `audit-logging-fluentd-ds-splunk-hec-config` contém um plug-in de saída que é usado para encaminhar logs de auditoria para o Splunk. O plug-in de saída está incluído no arquivo `ConfigMap` principal, `audit-logging-fluentd-ds-config`.

Atualize o arquivo `ConfigMap`

`audit-logging-fluentd-ds-splunk-hec-config`

1. No menu de navegação, clique em **Configuração > ConfigMap**.
2. Procure pelo arquivo `audit-logging-fluentd-ds-splunk-hec-config`.
3. Clique em **Editar**.
4. Inclua o nome do host, o número da porta e o `SPLUNK_HEC_TOKEN` do servidor Splunk.
5. Clique em **Enviar**.

O exemplo a seguir é um arquivo `ConfigMap` `audit-logging-fluentd-ds-splunk-hec-config` padrão.

```
{
  "apiVersion": "v1",
  "kind": "ConfigMap",
  "metadata": {
    "name": "audit-logging-fluentd-ds-splunk-hec-config",
    "namespace": "kube-system",
    "resourceVersion": "11134128",
    "labels": {
      "app": "audit-logging-fluentd",
      "chart": "audit-logging-3.2.0",
      "component": "fluentd",
      "heritage": "Tiller",
      "release": "audit-logging"
    }
  },
  "data": {
    "splunkHEC.conf": "<match icp-audit kube-audit>\n  @type splunk_hec\n  hec_host
SPLUNK_SERVER_HOSTNAME\n  hec_port SPLUNK_PORT\n  hec_token SPLUNK_HEC_TOKEN\n  ca_file
/fluentd/etc/tls/splunkCA.pem\n\n  source ${tag}\n</match>"
  }
}
```

Nota: o valor da chave `splunkHEC.conf` está no formato de yaml convertido para sequência. `\n` representa uma mudança de linha. Espaços entre `\n` e a próxima palavra devem ser deixados como estão. Mudanças no espaçamento podem resultar em erros.

Por exemplo, `<match icp-audit kube-audit>\n  @type splunk_hec\n  hec_host SPLUNK_SERVER_HOSTNAME\n`. O número de espaços entre o primeiro `\n` e a palavra `@type` é o mesmo que o segundo `\n` e o `hec_host`. Certifique-se de manter o mesmo número de espaços se você incluir um novo campo.

Atualize o arquivo `ConfigMap`

`audit-logging-fluentd-ds-config`

1. No menu de navegação, clique em **Configuração > ConfigMap**.
2. Procure pelo arquivo `audit-logging-fluentd-ds-config`.
3. Clique em **Editar**.
4. Configure a chave `ENABLE_AUDIT_LOGGING_FORWARDING` como `true` para ativar o encaminhamento.
5. Remova o comentário da linha, `@include /fluentd/etc/splunkHEC.conf`. Comente outras linhas de plug-in de saída.
6. Clique em **Enviar**.

O exemplo a seguir é o arquivo `ConfigMap` `audit-logging-fluentd-ds-config` padrão.

```
{
  "apiVersion": "v1",
  "kind": "ConfigMap",
  "metadata": {
    "name": "audit-logging-fluentd-ds-config",
    "namespace": "kube-system",
    "resourceVersion": "11920745",
    "labels": {
      "app": "audit-logging-fluentd",
      "chart": "audit-logging-3.2.0",
      "component": "fluentd",
```

```
      "heritage": "Tiller",
      "release": "audit-logging"
    }
  },
  "data": {
    "ENABLE_AUDIT_LOGGING_FORWARDING": "false",
    "fluent.conf": "# Input plugins\n@include /fluentd/etc/source.conf\n\n# Output plugins\n# Only
use one output plugin conf file at a time. Comment or remove other files \n\n# To forward audit logs
to ELK, uncommnet following line  and restart the
'audit-logging-fluentd-ds-*' pods\n#@include /fluentd/etc/elk.conf\n\n# To forward audit logs to
QRadar,
uncommnet following line, add QRadar server information in the 'audit-logging-fluentd-ds-remote-
syslog-config'
ConfigMap and restart the 'audit-logging-fluentd-ds-*' pods\n#@include
/fluentd/etc/remoteSyslog.conf\n\n#To
forward audit logs to Splunk over HTTPS, uncomment following line, add Splunk server information in
the
'audit-logging-fluentd-ds-splunk-hec-config' ConfigMap and restart the 'audit-logging-fluentd-ds-*'
pods\n#@include /fluentd/etc/splunkHEC.conf"
  }
}
```

Nota: o valor da chave `fluent.conf` está no formato de yaml de conversão para sequência. Mudanças no espaçamento podem resultar em erros. Para comentar qualquer arquivo de configuração, inclua `#` na frente de `@include`. Por exemplo, `\n#@include /fluentd/etc/elk.conf\n\n`, `elk.conf` é comentado.

Remova os pods `audit-logging-fluentd-ds*` fluentd

Use um dos métodos a seguir para remover todos os pods no daemonset `audit-logging-fluentd-ds`. Os pods serão recriados com o roteamento de log de auditoria apropriado.

- Use o console para remover todos os pods do daemonset:
  1. Efetue login no console.
  2. Navegue para **Carga de trabalho > DaemonSets**
  3. Localize e clique em `audit-logging-fluentd-ds daemonset`.
  4. Exclua todos os pods.
- Use a CLI do Kubernetes para remover todos os pods do daemonset:
  1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
  2. Execute o comando a seguir:

     ```
     kubectl get pod -n kube-system -o wide | grep audit-logging-fluentd-ds- | awk '{print
     $1}' | xargs kubectl delete pod -n kube-system
     ```

Inclua hostAliases na especificação de implementação de daemonset

`audit-logging-fluentd-ds`

1. Use o comando a seguir para editar a implementação do daemonset.

   ```
   kubectl -n kube-system edit daemonset audit-logging-fluentd-ds
   ```

2. Inclua o mapeamento de nome do host e de endereço IP do servidor Splunk na `spec` no nível do pod. É possível incluir múltiplos mapeamentos de nomes de host e de endereço IP.

   ```
   hostAliases:
    - hostnames:
      - <Splunk-Server-Host-Name>
      ip: <Splunk-Server-IP-Address>
    - hostnames:
      - <QRadar-Server-Host-Name>
      ip: <QRadar-Server-IP-Address>
   ```

3. Salve o arquivo editado.

Configuração do Fluentd

Usando o plug-in de saída `fluent-plugin-splunk-hec`

```
<match icp-audit kube-audit>
  @type splunk_hec
  hec_host SPLUNK_SERVER_HOSTNAME
  hec_port SPLUNK_PORT
  hec_token SPLUNK_HEC_TOKEN
  ca_file /fluentd/etc/tls/splunkCA.pem

  # Following parameters overwrite HEC default parameters. Optional  parameters are:
  index awesome
  source ${tag}
  sourcetype _json
</match>
```

Para obter mais informações, consulte a Documentação do Splunk.

Splunk

Deve-se configurar o Splunk para obter do fluentd sobre o coletor de eventos HTTP.

HTTP Event Collector (HEC)

1. Ative o HEC.
2. Crie um token HEC e customize-o.
3. Ative o token.

Para obter mais informações, consulte a [Documentação do Splunk]
(https://docs.splunk.com/Documentation/Splunk/7.1.7/Data/UsetheHTTPEventCollector.

Configurações e certificados do Splunk

A seguir estão os diretórios padrão para configurações e certificados.

- Diretório de configuração padrão - `/opt/splunk/etc/system/local`
- Diretório de certificados padrão - `/opt/splunk/etc/auth`

O `/opt/splunk/etc/system/local` inclui os arquivos `input.conf` e `server.conf` que devem ser modificados.

Inclua a seguinte configuração `http` no arquivo `input.conf`

```
# HEC plugin configuration
[http]
port = 8088
disabled = 0
enableSSL = 1
dedicatedIoThreads = 4
maxSockets = 50
maxThreads = 20
serverCert = <server certificates path > # For example -
/opt/splunk/etc/auth/myNewServerCertificate.pem
sslPassword = <certificates password>
```

Inclua o caminho do certificado CA em `[sslConfig]` no arquivo

```
server.conf

[sslConfig]
sslRootCAPath = <ca certificate path> # For example - /opt/splunk/etc/auth/myCACertificate.pem
```

Para obter informações de configuração customizada, consulte a [Documentação do Splunk]
(https://docs.splunk.com/Documentation/Splunk/7.2.6/Admin/Inputsconf.

Para obter informações sobre certificados, consulte a [Documentação do Splunk]
(https://docs.splunk.com/Documentation/Splunk/7.2.6/Security/AboutcreatingcertificatesforSplunk.

Reinicie o serviço `splunkd`

Depois de modificar a configuração, reinicie o serviço usando o comando a seguir:

```
$SPLUNK_HOME/bin/splunk restart splunkd
```

Ativando e desativando o modo FIPS

Depois de instalar o IBM® Cloud Private, é possível ativar ou desativar a conformidade com o Federal Information Processing Standard (FIPS) 140-2 para o ingresso de gerenciamento do IBM Cloud Private (console de gerenciamento), o controlador de ingresso NGINX (serviço de ingresso), o gerenciador de imagem, o registro do Docker e o WebSphere Liberty Application Server (gerenciador de autenticação).

Por padrão, o modo de conformidade FIPS está desativado.

Tipo de usuário ou nível de acesso necessários: administrador de cluster ou administrador de equipe.

Ative ou desative o modo FIPS para o ingresso de gerenciamento do

IBM Cloud Private

1. Efetue logon no console de gerenciamento.
2. Faça download do gráfico `icp-management-ingress-3.1.1.tgz`.
3. Copie o gráfico para um local provisório.
   - Para um cluster do Linux®, execute o comando a seguir:

     ```
     docker run --rm -e LICENSE=accept -i -v /tmp:/tmp ibmcom/icp-inception-amd64:3.2.0-ee cp
     /addon/icp-management-ingress-3.1.1.tgz /tmp
     ```

   - Para um cluster do Linux® on Power® (ppc64le), execute o comando a seguir:

     ```
     docker run --rm -e LICENSE=accept -i -v /tmp:/tmp ibmcom/icp-inception-ppc64le:3.2.0-ee
     cp /addon/icp-management-ingress-3.1.1.tgz /tmp
     ```

4. Configure a interface da linha de comandos (CLI) `helm` como um usuário administrativo. Para obter mais informações sobre como configurar a CLI do Helm, consulte Instalando a CLI do Helm (helm).
5. Para ativar o modo FIPS, execute os comandos a seguir:

   ```
   helm get values --tls icp-management-ingress > /tmp/old-value.yaml
   helm upgrade --set fips_enabled=true icp-management-ingress -f /tmp/old-value.yaml /tmp/icp-
   management-ingress-3.1.1.tgz --tls
   ```

6. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
7. Verifique o log do contêiner `icp-management-ingress` para verificar se o modo FIPS está ativado. No exemplo a seguir, `icp-management-ingress-kj5z6` é o nome do pod.

   ```
   kubectl logs icp-management-ingress-kj5z6 | grep FIPS
   ```

   O seguinte é uma saída de amostra:

   ```
   29/09/2018 09:24:56 [aviso] 20#20: FIPS_mode_set()
   com sucesso (SSL:)
   ```

Para desativar o modo FIPS, execute o comando a seguir:

```
helm upgrade --set fips_enabled=false icp-management-ingress /tmp/icp-management-ingress-3.1.1.tgz -
-tls
```

Ative ou desative o modo FIPS para o controlador de ingresso NGINX

1. Efetue logon no console de gerenciamento.
2. Faça download do gráfico `nginx-ingress-3.1.1.tgz`.
3. Copie o gráfico para um local provisório.
   - Para um cluster do Linux®, execute o comando a seguir:

     ```
     docker run --rm -e LICENSE=accept -it -v /tmp:/tmp ibmcom/icp-inception-amd64:3.2.0-ee cp
     /addon/nginx-ingress-3.1.1.tgz /tmp
     ```

- Para um cluster do Linux® on Power® (ppc64le), execute o comando a seguir:

  ```
  docker run --rm -e LICENSE=accept -it -v /tmp:/tmp ibmcom/icp-inception-ppc64le:3.2.0-ee
  cp /addon/nginx-ingress-3.1.1.tgz /tmp
  ```

4. Configure a interface da linha de comandos (CLI) `helm` como um usuário administrativo. Para obter mais informações sobre como configurar a CLI do Helm, consulte Instalando a CLI do Helm (helm).

5. Para ativar o modo FIPS, execute os comandos a seguir:

   ```
   helm get values --tls nginx-ingress > /tmp/old-value.yaml
   helm upgrade --set fips_enabled=true nginx-ingress -f /tmp/old-value.yaml /tmp/nginx-ingress-
   3.1.1.tgz --tls
   ```

6. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

7. Verifique o log de contêiner do controlador de ingresso NGINX para verificar se o modo FIPS está ativado. No exemplo a seguir, `nginx-ingress-controller-qhczr` é o nome do pod.

   ```
   kubectl logs nginx-ingress-controller-qhczr | grep FIPS
   ```

   O seguinte é uma saída de amostra:

   ```
   29/09/2018 09:24:56 [aviso] 20#20: FIPS_mode_set()
   com sucesso (SSL:)
   ```

Para desativar o modo FIPS, execute o comando a seguir:

```
helm upgrade --set fips_enabled=false nginx-ingress /tmp/nginx-ingress-3.1.1.tgz --tls
```

Ative ou desative o modo FIPS para o gerenciador de imagens do IBM Cloud Private

Para ativar o modo FIPS para o gerenciador de imagens do IBM Cloud Private, execute estes comandos:

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Edite o gerenciador de imagem StatefulSet.

   ```
   kubectl edit StatefulSets image-manager -n kube-system
   ```

3. Mude o valor da variável de ambiente denominada `FIPS_ENABLED` como `true` para o contêiner `image-manager`.
4. Salve o StatefulSet.

Para desativar o modo FIPS para o gerenciador de imagem do IBM Cloud Private, execute estes comandos:

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Edite o gerenciador de imagem StatefulSet.

   ```
   kubectl edit StatefulSets image-manager -n kube-system
   ```

3. Mude o valor da variável de ambiente denominada `FIPS_ENABLED` para `false` para o contêiner `image-manager`.
4. Salve o StatefulSet.

Ative ou desative o modo FIPS para o registro do Docker

Para ativar o modo FIPS para registro do Docker, execute estes comandos:

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Edite o gerenciador de imagem StatefulSet.

   ```
   kubectl edit StatefulSets image-manager -n kube-system
   ```

3. Mude o valor da variável de ambiente denominada `FIPS_ENABLED` como `true` para o contêiner `icp-registry`.
4. Salve o StatefulSet.

Para desativar o modo FIPS para o gerenciador de imagem do IBM Cloud Private, execute estes comandos:

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Edite o gerenciador de imagem StatefulSet.

   ```
   kubectl edit StatefulSets image-manager -n kube-system
   ```

3. Mude o valor da variável de ambiente denominada `FIPS_ENABLED` para `false` para o contêiner `icp-registry`.
4. Salve o StatefulSet.

Ative ou desative o modo FIPS para o gerenciador de autenticação

Para ativar o modo FIPS para o gerenciador de autenticação, execute estes comandos:

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Edite o configmap `platform-auth-idp` .

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

3. Mude o valor da variável denominada `FIPS_ENABLED` como `true`.
4. Salve o mapa de configuração.
5. Reinicie os pods `auth-idp` excluindo-os.

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

6. Aguarde algum tempo para os pods reiniciarem. Verifique o status.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

   Quando o status de todos os pods `auth-idp` mostrar `4/4 Running`, os pods estarão prontos.

Para desativar o modo FIPS para o gerenciador de autenticação, execute estes comandos:

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Edite o configmap `platform-auth-idp` .

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

3. Mude o valor da variável denominada `FIPS_ENABLED` para `false`.
4. Salve o mapa de configuração.
5. Reinicie os pods `auth-idp` excluindo-os.

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

6. Aguarde algum tempo para os pods reiniciarem. Verifique o status.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

   Quando o status de todos os pods `auth-idp` mostrar `4/4 Running`, os pods estarão prontos.

Guia de rede

Revise as informações para gerenciar sua rede do IBM® Cloud Private.

- Rede de contêineres
- Política de rede
- Entendendo nós de alta disponibilidade e do proxy
- Configurações de cluster
- Plug-ins do CNI
- Gerenciador de tráfego local F5 BIG-IP

Rede de contêineres

É possível gerenciar seu cluster do IBM® Cloud Private usando contêineres.

- Container Network Interface
- Modelo de rede do Kubernetes
- Tipos de serviço do Kubernetes
- Descoberta de serviço (kube-dns)
- Recursos de ingresso

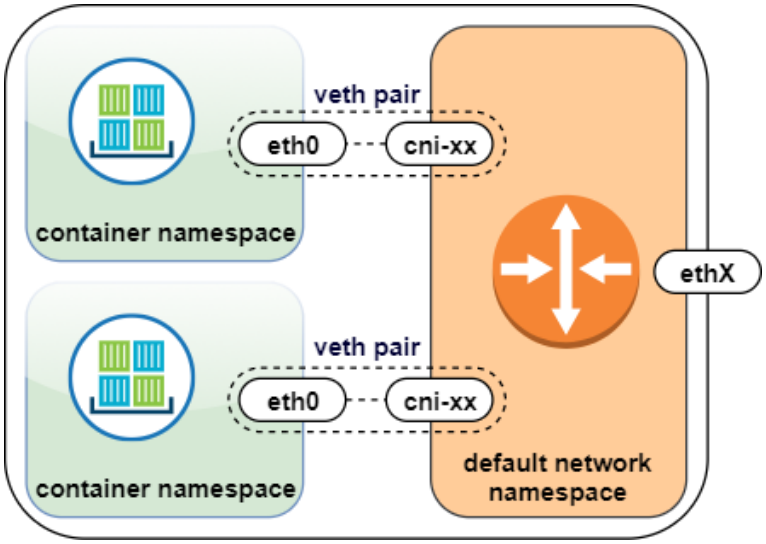
Interface de rede do contêiner

Um contêiner precisa de pelo menos uma interface de rede para se comunicar com outros contêineres ou terminais.

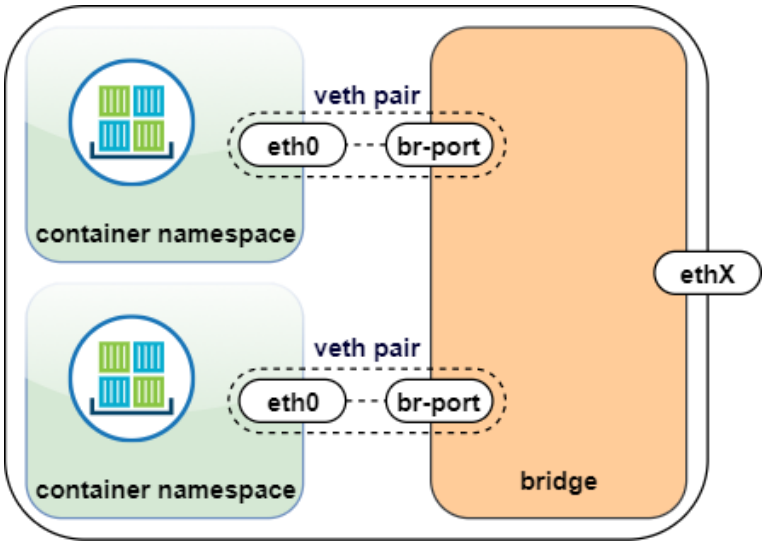
No Kubernetes, a rede de contêiner é estendida pelo cluster e controla como o tráfego de rede ocorre entre os nós no cluster. No ponto de vista do desenvolvedor do aplicativo, todos os contêineres em um cluster do Kubernetes estão em uma sub-rede simples.

Quando um contêiner é iniciado, uma interface de rede lógica é fornecida dentro do contêiner por meio de uma extremidade de um par veth no namespace de rede do contêiner. A outra extremidade do par veth é apresentada no namespace de rede padrão no nó do trabalhador. A extremidade do par veth no nó pode ser apresentada como uma interface lógica ou como uma interface em uma ponte de Camada 2, dependendo da tecnologia de rede subjacente.

Exemplo: eth0 conectado a uma interface de rede lógica em um namespace de rede padrão de um nó



Exemplo: eth0 conectado a uma interface de ponte em um namespace de rede padrão do nó e ponte com um uplink

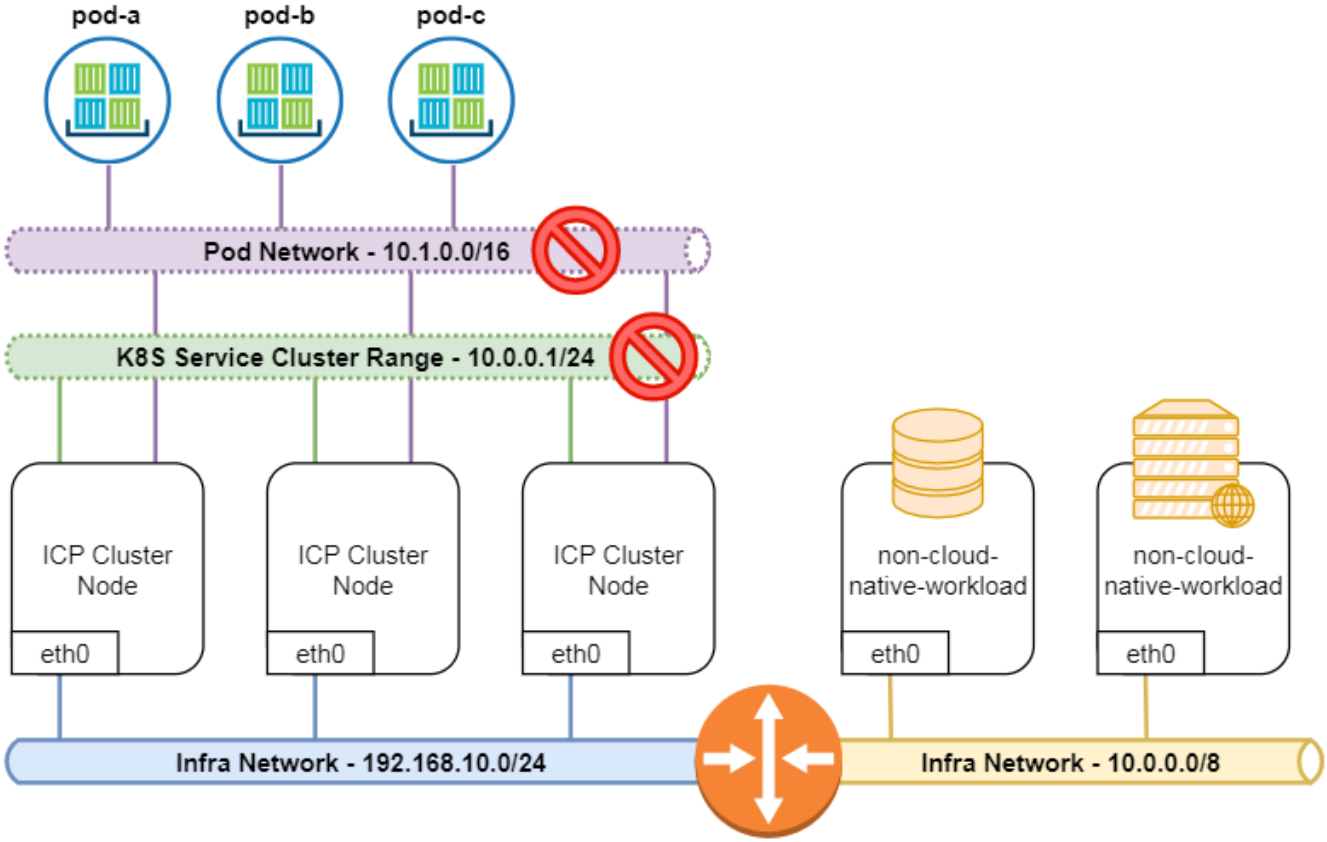


Modelo de rede do Kubernetes

O Container Network Interface (CNI) é responsável por fornecer endereços IP para os pods que estão em execução no Kubernetes e programar os nós do trabalhador para rotear os pacotes de acordo com o Modelo de rede do Kubernetes.

Os IPs do pod são desenhados a partir de um conjunto de IPs que foi criado no momento da instalação e geralmente devem ser selecionados a partir do intervalo de rede privada RFC1918. Esses IPs, também chamados de CIDR de rede de pod, são confinados no cluster do Kubernetes e são especificados no momento da instalação do IBM® Cloud Private usando o parâmetro `network_cidr` em `config.yaml` na notação CIDR. O tamanho `network_cidr` deve ser selecionado com o tamanho apropriado para o número de pods que se espera que sejam executados em todo o cluster.

Nota: a sub-rede selecionada não deve estar em conflito com nenhum recurso de rede fora do cluster com o qual os contêineres podem precisar se comunicar, incluindo uma ou mais sub-redes nas quais os nós do cluster se encontram.**



Conforme mostrado no exemplo, o `network_cidr` (10.1.0.0/16, que é o valor padrão em config.yaml) e o `service_cluster_ip_range` (10.0.0.1/24, que é o valor-padrão em config.yaml) entram em conflito com o CIDR 10.0.0.0/8 de rede de infraestrutura. Isso pode quebrar a comunicação entre os pods e o non-cloud-native-workload anterior. Para evitar tais situações, os atributos `network_cidr` e `service_cluster_ip_range` no cluster.yaml e no config.yaml devem ser configurados para que eles não entrem em conflito com o CIDR de infraestrutura.

Tipos de serviço do Kubernetes

Quando um pod precisa se comunicar com outro pod, ele precisa de uma maneira de saber o endereço IP do outro pod. Os serviços do Kubernetes fornecem um mecanismo para localização de outros pods.

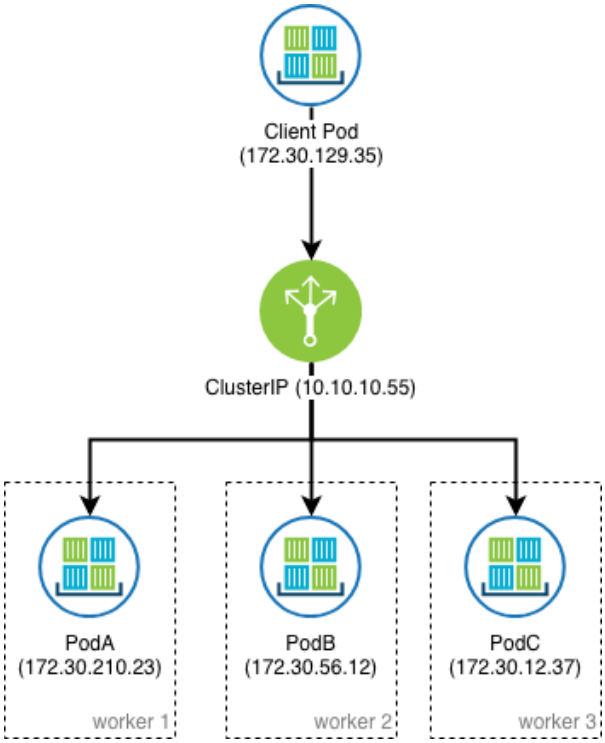
De acordo com o modelo de rede do Kubernetes, os IPs de pod são efêmeros, ou seja, se um pod travar ou for excluído e um novo pod for criado, muito provavelmente ele receberá um novo endereço IP. Serviços do Kubernetes permite que você selecione um mecanismo para localização de outros pods.

ClusterIP

Um IP fixo interno conhecido como um `ClusterIP` pode ser criado na frente de um pod ou de uma réplica conforme necessário. Esse endereço IP fixo é extraído de outro conjunto de IP, que é especificado no tempo de instalação do IBM® Cloud Private usando o parâmetro `service_cluster_ip_range` em `config.yaml`. Ele é selecionado a partir do intervalo de rede privado RFC1918 como com o parâmetro `network_cidr`. O tamanho desta sub-rede deve ser escolhido em consideração ao número de serviços esperados no cluster.

Nota: é importante que a sub-rede selecionada não entre em conflito com nenhum recurso de rede fora do cluster com o qual os contêineres possam precisar se comunicar, incluindo o parâmetro `network_cidr` e uma ou mais sub-redes nos quais os nós do cluster se encontram.

O ClusterIP fornece um endereço IP de carga balanceada. Um ou mais pods que correspondem a um seletor de rótulo pode encaminhar o tráfego para o endereço IP. O serviço `ClusterIP` deve definir uma ou mais portas para atender com as portas de destino para encaminhar o tráfego TCP/UDP para contêineres. O endereço IP que é usado para o `ClusterIP` não é roteável fora do cluster, como o endereço IP do pod é.



Internamente, o Kubernetes resolve o seletor de rótulo para um conjunto de pods e obtém os endereços IP do Pod efêmero e gera recursos `Terminais` para os quais o proxies `ClusterIP` encaminham o tráfego. O `kube-proxy` que está em execução em cada nó é usado para configurar cada nó do trabalhador. Ele encaminha o tráfego que é enviado para o endereço IP `ClusterIP` para um endereço IP efêmero do pod em tempo real que corresponde o seletor de rótulo em algum lugar no cluster. As regras de encaminhamento são atualizadas quando novos serviços são criados ou removidos, quando os pods que correspondem aos seletores de rótulo são iniciados ou removidos ou quando a atividade de pod é mudada.

A atividade de um pod é determinada por uma verificação de funcionamento que é definida no yaml para a implementação. Essa verificação de funcionamento pode ser um HTTP GET que espera um código de status 200, uma abertura de porta TCP ou a execução de um comando de dentro do contêiner que retorna um código de status específico. Essas verificações são especificadas na definição de recurso do pod. As verificações de funcionamento são executadas localmente em cada nó do trabalhador pelo processo `kubelet` e sincronizadas com o plano de controle. Se um limite de falha de verificação de funcionamento for atendido, o `ClusterIP` removerá o contêiner do grupo de destino.

Duas tecnologias principais podem ser usadas para implementar as regras de encaminhamento: iptables ou

ipvs.

- iptables: `kube-proxy` cria e sincroniza as regras iptables locais que interceptam o tráfego para o `ClusterIP` em cada nó e seleciona um pod de backend saudável aleatoriamente para o qual encaminhar o tráfego.
- ipvs: `kube-proxy` cria e sincroniza as regras ipvs que correspondem aos serviços do Kubernetes e pode aplicar um algoritmo de balanceamento de carga ao selecionar um backend para o proxy para o qual o tráfego é roteado.

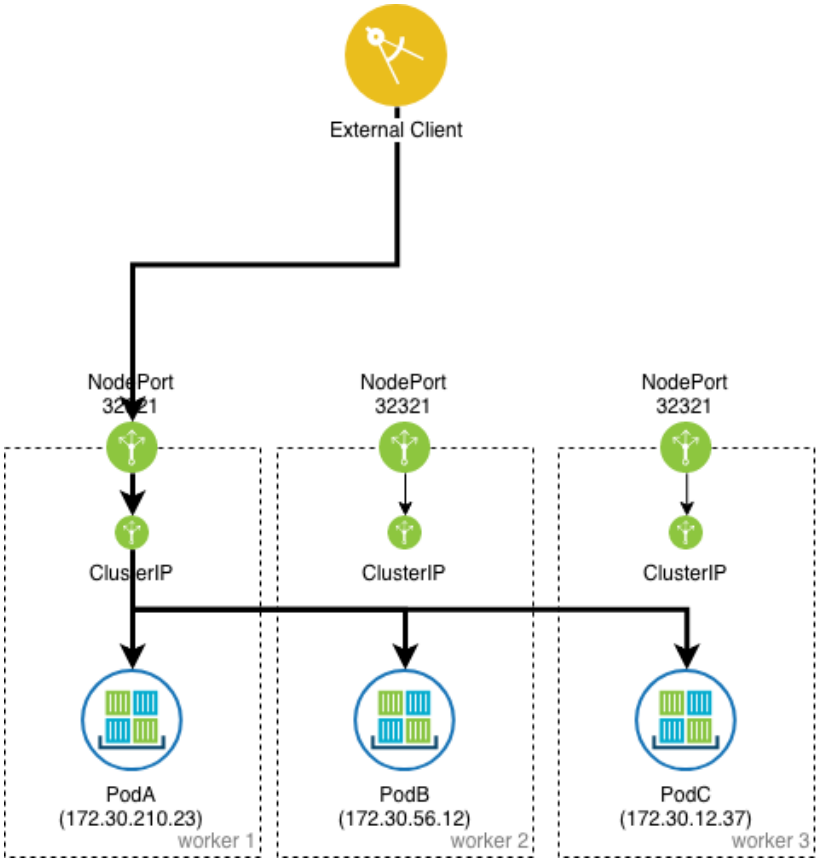
Em comparação com iptables, ipvs tem as seguintes vantagens:

- Alto desempenho para processamento de pacotes e inserção de novas regras
- A capacidade de construir balanceamento de carga de serviço

NodePort

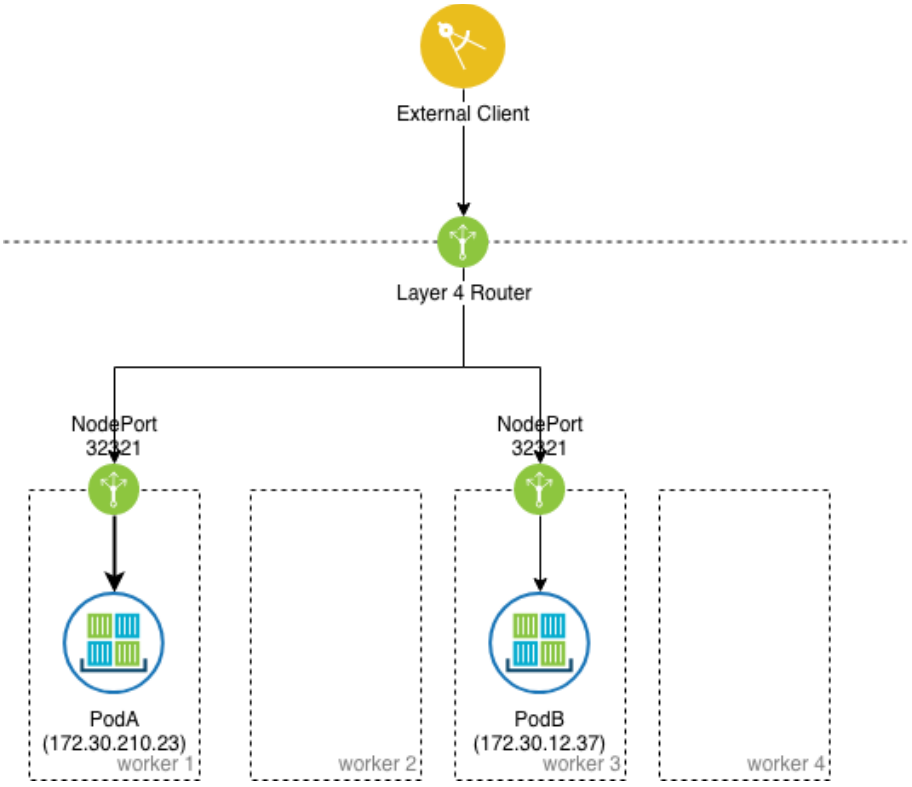
Serviços do tipo `NodePort` são construídos sobre os serviços de tipo `ClusterIP` expondo o serviço ClusterIP fora do cluster em portas altas (padrão 30000-32767). Se nenhum número de porta for especificado, o Kubernetes selecionará automaticamente uma porta livre. O `kube-proxy` local é responsável pelo atendimento na porta no nó e pelo encadeamento do tráfego do cliente no NodePort para o `ClusterIP`.

Por padrão, cada nó no cluster atende nesta porta, incluindo nós em que o pod que corresponde ao seletor de rótulo não é executado. O tráfego nesses nós é submetido a NAT internamente e encaminhado para o pod de destino (política de tráfego externo do `Cluster`).



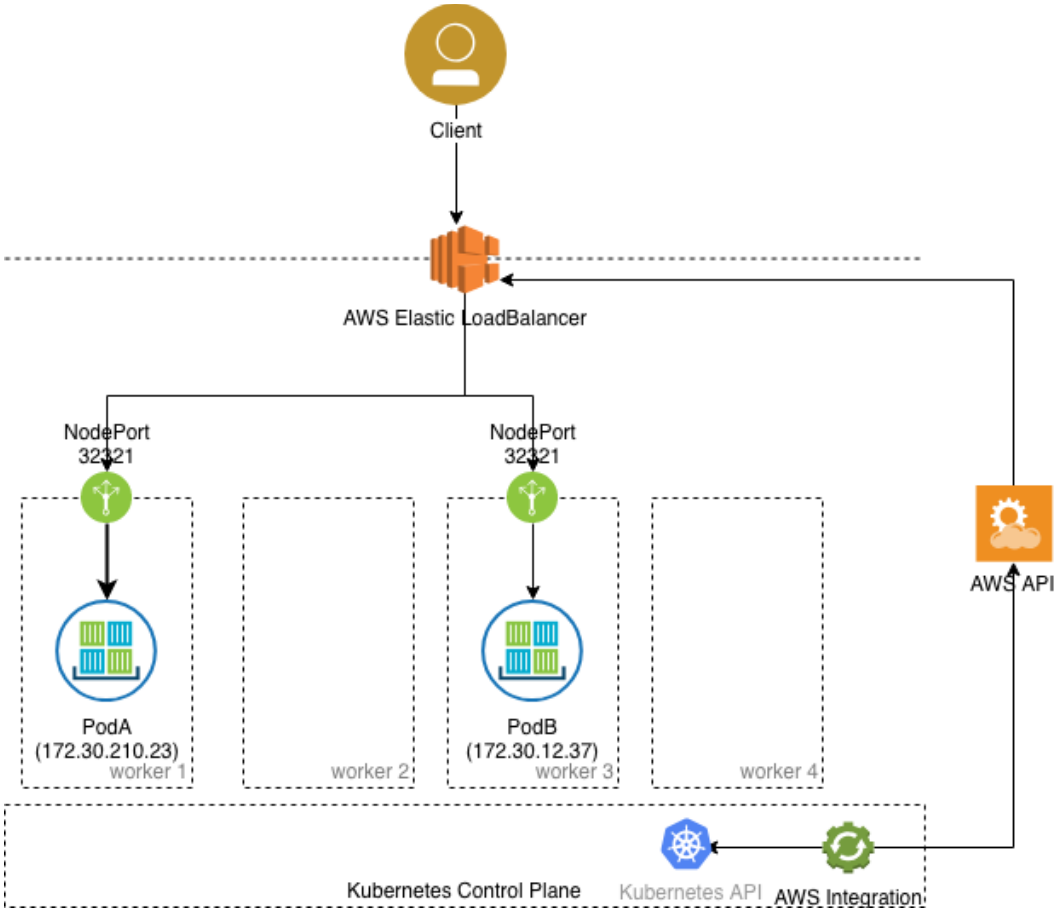
Este comportamento pode ser controlado no manifest do objeto de serviço do Kubernetes, configurando a propriedade `.spec.externalTrafficPolicy` como `Local`, que faz com que somente os nós do trabalhador que executam o pod atendam no NodePort especificado. Dessa forma, um hop extra pode ser evitado e o endereço IP do cliente é preservado quando ele se comunica com o pod.

O NodePort pode ser útil ao configurar manualmente os balanceadores de carga externos para encaminhar o tráfego da camada 4 de clientes fora do cluster para um conjunto específico de pods que estão em execução no cluster do Kubernetes. Nesses casos, o número da porta específico que é usado para NodePort deve ser configurado antecipadamente e o balanceador de carga externo deve ser configurado para encaminhar o tráfego para a porta de recebimento em todos os nós do trabalhador. Uma verificação de funcionamento deve ser configurada no balanceador de carga externo para determinar quais nós do trabalhador estão executando pods saudáveis e quais não são.



LoadBalancer

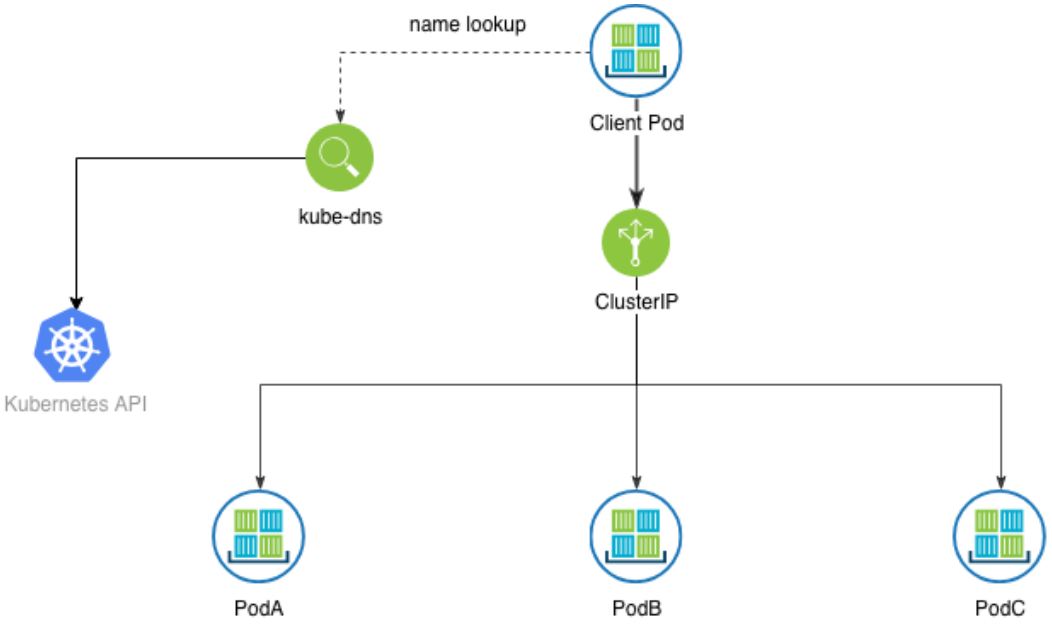
O tipo de serviço `LoadBalancer` é construído sobre os tipos de serviço `NodePort` por meio do fornecimento e configuração de balanceadores de carga externos a partir de provedores de nuvem pública e privada. Ele expõe os serviços que estão em execução no cluster encaminhando o tráfego da camada 4 para os nós do trabalhador. Esta é uma maneira dinâmica de implementar um caso que envolve balanceadores de carga e serviços de tipo `NodePort` externos. No entanto, ele geralmente requer uma integração que é executada dentro do cluster do Kubernetes que executa um relógio na API para serviços do tipo `LoadBalancer`.



Descoberta de serviço (kube-dns)

O Kubernetes espera que um serviço esteja em execução dentro da malha da rede do pod que executa a resolução de nome e age como o servidor de nomes principal dentro do cluster.

No IBM® Cloud Private, isso é implementado usando o CoreDNS que é executado nos nós principais. O CoreDNS resolve nomes para todos os serviços que estão em execução no Kubernetes e encaminha consultas de nomes com relação aos servidores de nomes de envio de dados em nome de contêineres. O serviço DNS em si é executado como um serviço `ClusterIP` que é suportado por um ou mais contêineres para alta disponibilidade.



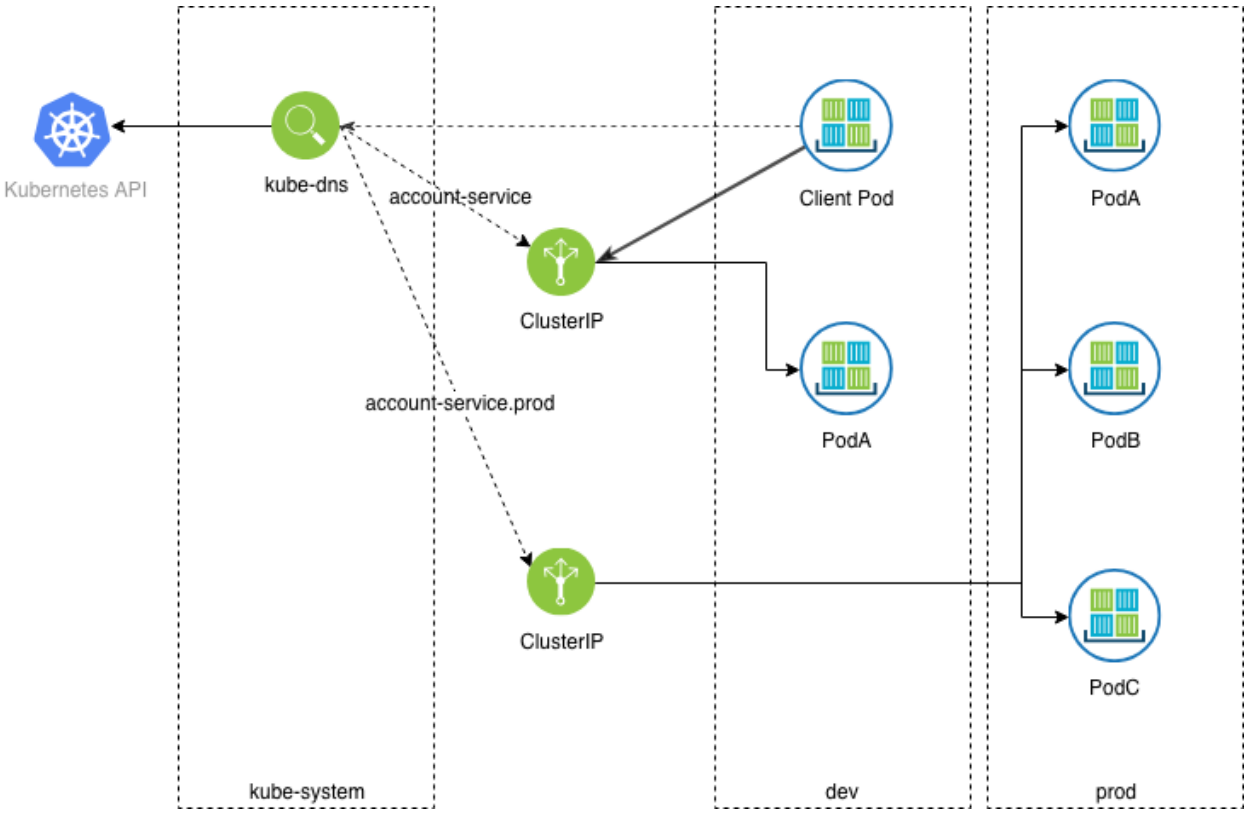
Os nomes de serviço do Kubernetes são resolvidos para `ClusterIPs` que representam um ou mais pods que correspondem a um seletor de rótulo. O cluster é designado a um domínio do cluster que é especificado no momento da instalação usando `cluster_domain` (este é `cluster.local`, por padrão) para distinguir entre nomes locais dos nomes de cluster e externos.

Cada cluster do Kubernetes é logicamente separado em namespaces e cada namespace age como um subdomínio para resolução de nome. Ao examinar um `/etc/resolv.conf` do contêiner, observe que os pontos da linha `nameserver` em um endereço IP interno para o cluster e os sufixos `search` são gerados em uma ordem específica:

```
# cat /etc/resolv.conf
nameserver <kube-dns ClusterIP>
search <namespace>.svc.<cluster_domain> svc.<cluster_domain> <cluster_domain> <additional ...>
options ndots:5
```

O `<additional ...>` é uma lista de sufixos de procura obtidos do arquivo `/etc/resolv.conf` do nó do trabalhador.

Por padrão, um nome abreviado do host como `account-service` tem `<namespace>.svc.<cluster_domain>` anexado a ele, por isso, é selecionado um pod que corresponde ao seletor de rótulo que está em execução no mesmo namespace que o pod em execução. Um pod pode consultar o `ClusterIP` de um pod em um namespace diferente anexando o namespace ao nome do host. Por exemplo, o `account-service.prod` tem como destino o `account-service` que está em execução no namespace `prod`, já que o sufixo de procura `svc.<cluster_domain>` é anexado ao final.



Observe a última linha em `/etc/resolv.conf`, `options ndots:5`, que indica ao resolvedor do sistema do contêiner que quaisquer nomes de host que estão sendo resolvidos e que tenham menos de cinco pontos no nome devem ter os sufixos de domínio de procura anexados a eles. Isso pode afetar o desempenho, já que as consultas de recursos de rede externa com menos de cinco pontos no nome resultam em consultas para cada entrada na linha de procura. Por exemplo, uma consulta de `www.ibm.com` resulta em consultas de `www.ibm.com.<namespace>.svc.<cluster_domain>`, `www.ibm.com.svc.<cluster_domain>`, `www.ibm.com.<cluster_domain>`, etc., antes de tentar `www.ibm.com`. Para resolver esse problema, a inclusão de um `.` adicional no término de nomes completos de domínio que são usados na configuração do aplicativo evita que o resolvedor do sistema faça um ciclo por meio da lista de sufixos nas consultas de nome (por exemplo, `www.ibm.com.`).

Serviços sem interface com o usuário

Em alguns casos, é desejável que um serviço `ClusterIP` não seja criado em todos os casos; isso pode ser feito especificando um tipo de serviço `None`. Isso cria registros A para cada pod que corresponde ao seletor de rótulo no DNS, mas no ClusterIP. Isso é normalmente usado com StatefulSets, em que cada um dos pods precisa ter nomes resolvíveis para comunicação entre todos os pods no conjunto (por exemplo, um banco de dados em cluster, como o MongoDB). Quando um serviço sem interface com o

usuário é criado, um registro de cada pod está no formato `<pod-name>.<service-name>.<namespace>.svc.<cluster-domain>`.

Serviços externos

É possível ter terminais do proxy do Kubernetes fora do cluster. Isso pode ser feito criando um recurso `Service` sem seletor de rótulo e criando recursos `Endpoints` manualmente que contêm os IPs fora do cluster para proxy ou criando um recurso `Service` com o tipo `ExternalName` contendo um nome DNS externo. Essa ação cria um registro CNAME no DNS do cluster. Usando essas funções, o DNS do cluster pode ser usado como descoberta de serviço para serviços dentro e fora do cluster.

Recursos de ingresso

Um ingresso é uma coleção de regras para permitir conexões de entrada com os serviços de cluster do Kubernetes. Ele pode ser configurado para fornecer aos serviços do Kubernetes URLs acessíveis externamente, para finalizar conexões TLS, oferecer hospedagem virtual baseada em nome, etc.

Controlador de ingresso para o tráfego de Camada 7

Os recursos de ingresso do no Kubernetes são usados para o tráfego de Camada 7 do proxy para contêineres no cluster.

Os recursos de ingresso requerem que um componente do controlador de ingresso seja executado como um serviço de proxy de Camada 7 dentro do cluster. No IBM® Cloud Private, um controlador de ingresso baseado em nginx é fornecido por padrão e é implementado em nós de proxy ou principais (caso o principal aja como proxy).

O controlador de ingresso padrão observa objetos de ingresso do Kubernetes em todos os namespaces por meio do Kubernetes API e programa dinamicamente as regras de proxy nginx para serviços de envio de dados com base no recurso de ingresso. Por padrão, o controlador de ingresso é autoinicializado com políticas de balanceamento de carga, como algoritmos de balanceamento de carga, esquema de ponderação de backend, etc.

Mais de um controlador de ingresso também poderá ser implementado se o isolamento entre namespaces for necessário. O próprio controlador de ingresso é uma implementação do contêiner que pode ser escalada. Ele é exposto em uma porta do host nos nós do proxy e pode fazer proxy de toda a malha IP de pod e de serviço que esteja em execução no cluster.

Para negar todo o tráfego de ingresso para aplicativos que estão em seu namespace, consulte Negando o tráfego de ingresso.

Ingresso de serviço único

É possível expor um serviço único por meio do ingresso. No exemplo a seguir, o servidor Node.js foi criado com o nome de serviço `mynode-ibm-nodejs-sample` na porta 3000. Nesse caso, todo o tráfego no endereço do controlador de ingresso e na porta (80 ou 443) é encaminhado para esse serviço.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: mynode1ing
spec:
  backend:
    serviceName: mynode-ibm-nodejs-sample
    servicePort: 3000
```

Fanout simples

Com um fanout simples, é possível definir múltiplos serviços HTTP em caminhos diferentes e fornecer um único proxy que seja roteado para os terminais corretos no backend. Quando você tem um balanceador de carga altamente disponível que está gerenciando seu tráfego, esse tipo de recurso de ingresso é útil na redução do número de balanceadores de carga para um mínimo.

No exemplo a seguir, `/` é o destino de regravação para dois serviços: `customer-api` na porta 4191 e `orders-api` na porta 9090. Ambas as raízes de contexto desses serviços estão em `/`; o ingresso regrava o caminho `/api/customers/*` e `/api/orders/*` para `/` quando ele efetua proxy das solicitações para os backends.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    ingress.kubernetes.io/rewrite-target: /
```

```
  name: api
spec:
  rules:
  - host: api.example.com
    http:
      paths:
      - backend:
          serviceName: customer-api
          servicePort: 4191
        path: /api/customer/*
      - backend:
          serviceName: orders-api
          servicePort: 9090
        path: /api/orders/*
```

Hospedagem virtual baseada em nome

A hospedagem virtual baseada em nomes fornece o recurso para hospedar múltiplos aplicativos que estejam usando o mesmo endereço do controlador de ingresso. Este tipo de ingresso roteia solicitações de HTTP para diferentes serviços com base no cabeçalho `Host`. No exemplo a seguir, dois servidores Node.js são implementados. O console para o primeiro serviço pode ser acessado usando o nome do host `mynode1.example.com` e o segundo em `mynode2.example.com`. No DNS, o `mynode1.example.com` e o `mynode2.example.com` podem ser um registro A para o IP virtual de nó de proxy 10.0.0.1 ou CNAME para o balanceador de carga que encaminha o tráfego para onde o controlador de ingresso está atendendo.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    name: mynode1ing
spec:
  rules:
  - host: mynode1.example.com
    http:
      paths:
      - backend:
          serviceName: mynode1-ibm-nodejs-sample
          servicePort: 3000

apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    name: mynode2ing
spec:
  rules:
  - host: mynode2.example.com
    http:
      paths:
      - backend:
          serviceName: mynode2-ibm-nodejs-sample
          servicePort: 3000
```

Geralmente, é uma boa prática fornecer algum valor para o host, já que o padrão é `*`, que encaminha todas as solicitações para o backend.

TLS

Um serviço de ingresso pode ser protegido usando uma chave privada e um certificado TLS. A chave privada e o certificado TLS devem ser definidos em um segredo com nomes de chaves `tls.key` e `tls.crt`. O ingresso presume que a finalização e o tráfego TLS têm o proxy efetuado somente na porta 443.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    ingress.kubernetes.io/rewrite-target: /
  name: api
spec:
  rules:
  - host: api.example.com
    http:
      paths:
```

```
      - backend:
          serviceName: customer-api
          servicePort: 4191
        path: /api/customer/*
      - backend:
          serviceName: orders-api
          servicePort: 9090
        path: /api/orders/*
  tls:
  - hosts:
    - api.example.com
    secretName: api-tls-secret
```

No exemplo anterior, a finalização do TLS é incluída no recurso de ingresso `api.example.com`. O nome do assunto do certificado ou os nomes alternativos do assunto (SANs) devem corresponder ao valor do `host` no recurso de ingresso, ser válidos (não expirados) e a cadeia de certificados completa (incluindo quaisquer certificados intermediários e raiz) deve ser confiável pelo cliente, caso contrário, o aplicativo mostrará um aviso de segurança durante o handshake TLS. No exemplo, o nome do assunto `tls.crt` contém `api.example.com` ou é um certificado de curinga para `*.example.com`. A entrada DNS para `api.example.com` é um registro A para o endereço IP virtual dos nós do proxy.

O segredo `api-tls-secret` é criado no mesmo namespace que o recurso de ingresso, usando o comando a seguir:

```
kubectl create secret tls api-tls-secret --key=/path/to/tls.key --cert=/path/to/tls.crt
```

O segredo também poderá ser criado no yaml se as cargas úteis de chave e de certificado TLS forem codificadas em base-64.

```
apiVersion: v1
type: Opaque
kind: Secret
metadata:
  name: api-tls-secret
data:
  tls.crt: <base64-encoded cert>
  tls.key: <base64-encoded key>
```

WebSockets

O suporte de WebSockets é fornecido pelo controlador de ingresso nginx pronto para uso. Como o tempo limite padrão é 60 segundos, o tempo limite precisa ser aumentado. Para expor o serviço como um serviço do WebSocket, ele deve ser anotado na definição de recurso de ingresso como `nginx.org/websocket-services: "service1[,service2,...]"`.

Controlador de ingresso compartilhado

No IBM Cloud Private, um controlador de ingresso global é instalado por padrão e é implementado em todos os nós do proxy. Isso fornece a capacidade de definir os recursos de ingresso para seus aplicativos em todos os namespaces. O controlador de ingresso global é executado no namespace `kube-system`. Se uma NetworkPolicy for usada para isolar o tráfego de namespace, outro precisará ser criado para permitir o tráfego do controlador de ingresso com quaisquer serviços de backend com proxy em outros namespaces.

Vantagens:

- Um controlador de ingresso comum reduz recursos de cálculo que são necessários para os aplicativos de host.
- Um controlador de ingresso comum está pronto para uso imediato.

Desvantagens:

- Todo o tráfego do cliente passa por um controlador de ingresso compartilhado. O tráfego do cliente de um serviço pode afetar o outro.
- Capacidade limitada para isolar o tráfego de recursos de ingresso de envio de dados do tráfego de ingresso de recebimento de dados. Uma API pública e um painel de operações que está em execução no mesmo cluster compartilham o mesmo controlador de ingresso.
- Se um invasor obter acesso ao controlador de ingresso, ele poderá observar o tráfego decriptografado de todos os serviços com proxy.
- É necessário manter diferentes documentos do recurso de ingresso para diferentes estágios; ou seja, manter múltiplas cópias do mesmo arquivo yaml do recurso de ingresso com campos de namespace diferentes.
- O controlador de ingresso precisa de acesso para ler os recursos de ingresso, de serviço e de pod em cada namespace no Kubernetes API para implementar as regras de ingresso.

Controladores de ingresso isolados por namespace

Um controlador de ingresso pode ser instalado como um gráfico Helm em um namespace isolado e executar o ingresso para serviços no namespace. Neste tipo de implementação, o controlador de ingresso recebe uma função que pode acessar apenas o ingresso e os recursos no namespace.

Vantagens

- Delineação de recursos de ingresso para vários estágios de desenvolvimento, de produção e assim por diant;
- O desempenho de cada namespace pode ser escalado individualmente.
- O tráfego é isolado; quando combinado com nós do trabalhador isolado em VLANs separadas, o isolamento de Camada 2 pode ser obtido, já que o tráfego de envio de dados não sai de uma VLAN.
- O CI/CD pode usar o mesmo documento de recurso de ingresso a ser implementado (supondo que o namespace de desenvolvimento seja diferente do namespace de produção) em vários estágios.

Desvantagens:

- Controladores de ingresso adicionais devem ser implementados usando recursos adicionais.
- Os controladores de ingresso em namespaces separados podem precisar de um nó dedicado ou de um balanceador de carga externo dedicado.
- Negando o tráfego de ingresso

Negando o tráfego de ingresso

Negar todo tráfego de ingresso para aplicativos que estão sob seu namespace.

Por padrão, o tráfego de ingresso é permitido para aplicativos em seu namespace. É possível desativar o tráfego de ingresso.

1. No menu de navegação, clique em **Infraestrutura** > **Rede**.
2. Selecione **Negar tráfego de rede**.
3. Clique em **Confirmar**.

Política de rede e microssegmentação

No Kubernetes, recurso de política de rede é um conjunto de regras de tráfego de rede que são aplicadas a um grupo de pods em um cluster do Kubernetes.

A política de rede especifica como um pod tem permissão para se comunicar com outros. Os controladores de política de rede (em execução como pods no cluster do Kubernetes) convertem os requisitos e as restrições das políticas de rede que são recuperadas do Kubernetes API na infraestrutura de rede.

A `NetworkPolicy` pode ser aplicada aos pods em execução no Kubernetes. As regras de ingresso e de egresso podem conter um seletor de pod e as sub-redes de Camada 3 em um namespace de cada vez.

A natureza declarativa de `NetworkPolicy` como um documento yaml que pode ser confirmado para o controle de origem o torna uma parte ideal dos pipelines de implementação do DevSecOps. Como o documento de política é abstraído da implementação de rede real, o yaml declara apenas os intentos enquanto permite que o CNI implemente as regras na malha de rede.

Para configurar sua política de rede, consulte Criando uma NetworkPolicy.

As políticas a seguir são padrões comuns para aplicação na prática.

NetworkPolicy comum: negar todo o ingresso

Uma boa prática é definir e aplicar uma `NetworkPolicy` padrão para negar todo o tráfego de ingresso a todos os pods em todos os namespaces do aplicativo, em seguida, colocar em uma lista de desbloqueio pods e sub-redes com base nas necessidades do aplicativo. A `NetworkPolicy` a seguir seleciona todos os pods no namespace `prod` e não contém regras de ingresso, indicando que todo o tráfego de ingresso é descartado.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
```

```
  name: default-deny-all-ingress
  namespace: prod
spec:
  podSelector: {}
  ingress: []
```

NetworkPolicy comum: negar egresso externo

Outra boa prática é definir e aplicar uma `NetworkPolicy` padrão para negar o tráfego de egresso fora do cluster para namespaces do aplicativo, em seguida, incluir em uma lista de desbloqueio quaisquer sub-redes externas para os pods, conforme necessário. A política de rede a seguir seleciona todos os pods no namespace `prod` e permite o egresso dos pods dentro do cluster (todos os namespaces). Todo o outro tráfego é descartado.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-external-egress
  namespace: prod
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress:
  - to:
    - namespaceSelector: {}
```

NetworkPolicy comum: inclua em uma lista de desbloqueio tráfego de ingresso de pod

À medida que os desenvolvedores de aplicativos gravam microsserviços, como parte do ciclo de desenvolvimento, eles podem definir declarativamente os recursos da `NetworkPolicy` em yaml como um recurso de implementação. Esses recursos definem quaisquer regras de ingresso que são necessárias, por exemplo, quaisquer dependências do microsserviço, expressas usando seletores de pod. Por exemplo, a `NetworkPolicy` a seguir permite o tráfego para os pods do cliente a partir dos pods da web na porta 80 no namespace `prod`:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: customer-api-allow-web
  namespace: prod
spec:
  podSelector:
    matchLabels:
      app: customer
  ingress:
    - ports:
      - port: 80
      from:
      - podSelector:
          matchLabels:
            app: web
```

NetworkPolicy comum: permitir todo o tráfego de ingresso externo

Em uma camada da web, controlador de ingresso ou gateway da API, todo o tráfego de fora do cluster é permitido. Por exemplo, crie um `NetworkPolicy` no namespace `prod` que seleciona os pods da `web` e permite o ingresso de todas as origens:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: web-allow-external
  namespace: prod
spec:
  podSelector:
    matchLabels:
      app: web
  ingress:
  - ports:
    - port: 80
    from: []
```

Se um balanceador de carga externo for usado, em vez de permitir o ingresso de todas as origens, o ingresso deverá ser limitado apenas ao balanceador de carga externo.

Rede comum do NetworkPolicy: inclua em uma lista de desbloqueio egresso de pod para

sub-redes externas

Para serviços que requerem egresso para recursos fora do cluster, por exemplo, um banco de dados, uma API externa ou um diretório do usuário, inclua em uma lista de desbloqueio a sub-rede na qual o recurso de rede se encontra. Por exemplo, para permitir que o serviço de pedidos conversem com o banco de dados de pedido no `172.16.32.0/27` na porta 3306, use os comandos a seguir:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: customer-api-allow-web
  namespace: prod
spec:
  podSelector:
    matchLabels:
      app: orders
  policyTypes:
  - Egress
  egress:
  - ports:
    - port: 3306
    to:
    - ipBlock:
        cidr: 172.16.32.0/27
```

Para obter mais exemplos de `NetworkPolicy`, consulte Receitas de política de rede do Kubernetes.

Separação de obrigações

É importante assegurar a separação de obrigações. Use as funções do IBM® Cloud Private fornecidas e o RBAC do Kubernetes para restringir quem tem permissão para implementar recursos em cada namespace. Permissões relaxadas podem facilitar o contorno da política de rede implementando contêineres que correspondem ao seletor de rótulo do pod.

Crie uma conta de serviço para uma ferramenta de implementação de CI/CD com permissões para implementar contêineres e não permita que nenhum outro usuário crie essas implementações. Cumpra controles sobre o que aciona uma implementação, ou seja, quem pode confirmar a origem que aciona uma implementação para o cluster de aplicativo.

O administrador da rede é responsável por aplicar e auditar as políticas de rede em todo o cluster para conformidade. Como o NetworkPolicy é definido declarativamente e revisado antes da implementação, quaisquer políticas de rede existentes devem ser automatizadas e corresponder ao que é implementado no cluster a qualquer momento.

- Criando um NetworkPolicy
- Restringindo o acesso aos serviços de plataforma

Criando um NetworkPolicy

Configure um NetworkPolicy que controla o acesso às redes entre pods.

Para configurar o NetworkPolicy, as redes Calico devem estar ativadas durante a instalação do cluster.

Para obter mais informações sobre a configuração de políticas de rede, consulte Políticas de rede.

1. No menu de navegação, clique em **Plataforma** > **Rede**.
2. Selecione **Criar NetworkPolicy**.
3. Insira os detalhes da política. Para criar um NetworkPolicy, os parâmetros a seguir são necessários:
    - Um nome para a política.
    - `Aplicar a` - Uma lista de pods aos quais a política deve ser aplicada. Se você não especificar uma lista de pods, a política será aplicada a todos os pods. É possível obter uma lista de pods na console de gerenciamento ou na CLI do Kubernetes. Consulte a seção `De` a seguir.

- O número da porta e o protocolo para abrir. Se você não especificar um número da porta, todas as portas serão abertas.

- `From` - Uma lista de pods que têm acesso permitido. Se você não especificar uma lista de pods, todos os pods terão acesso aos pods designados. É possível usar seletores de rótulo de correspondência de namespace e pod para especificar os pods permitidos.

  Se você selecionar o seletor de rótulo de correspondência do pod, todos os pods com o rótulo especificado serão selecionados. O pod deve estar no mesmo namespace que o usuário associado. É possível visualizar uma lista de rótulos de pod na console de gerenciamento ou na CLI do Kubernetes. Por exemplo:

- Seletor de rótulo de correspondência de pod

  1. Obtenha a lista de pods:

     ```
     kubectl get pods
     ```

     A saída se assemelha ao texto a seguir:

     ```
     NAME                           READY     STATUS    RESTARTS   AGE
     dev-nginx-254164163-02gbb      1/1       Running   0          2h
     dev-tomcat-3353689452-k5bpz    1/1       Running   0          2h
     ```

  2. Obtenha os detalhes de um pod. Execute este comando:

     ```
     kubectl describe pods dev-nginx-254164163-02gbb
     ```

     A saída se assemelha ao texto a seguir:

     ```
     Name:        dev-nginx-254164163-02gbb
     Namespace:    dev
     Node:        9.21.62.194/9.21.62.194
     Start Time:    Fri, 10 Mar 2017 03:50:16 -0500
     Labels:        app=dev-nginx
           image=nginx_1_11_7
           pod-template-hash=254164163
     Status:        Running
     IP:        10.1.53.6
     Controllers:    ReplicaSet/dev-nginx-254164163
     Containers:
     ...
     ```

- Seletor de rótulo de correspondência de namespace

  Se você selecionar o seletor de rótulo de correspondência de namespace, todos os pods que estarão associados a um namespace que têm o rótulo de namespace especificado serão selecionados. É possível visualizar o rótulo do namespace na CLI do Kubernetes.

  1. Obtenha a lista de namespaces:

     ```
     kubectl get namespaces
     ```

     A saída se assemelha ao texto a seguir:

     ```
     NAME          STATUS    AGE
     default       Active    1h
     dev           Active    31m
     kube-system   Active    1h
     qa            Active    29m
     ```

  2. Abra o arquivo YAML que descreve um namespace. Execute este comando:

     ```
     kubectl get namespaces dev -o yaml
     ```

     Nesse exemplo, `dev` é o nome do namespace.

     O arquivo YAML é aberto e assemelha-se ao texto a seguir:

     ```
     apiVersion: v1
     kind: Namespace
     metadata:
      creationTimestamp: 2017-03-10T13:17:52Z
      labels:
     ```

```
    team: dev
  name: dev
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: 2017-03-10T13:17:52Z
  labels:
    team: dev
  name: dev
  ...
```

4. Clique em **Criar**.

Entendendo nós de alta disponibilidade e do proxy

Saiba sobre os componentes de plataforma que são necessários para configurar seu cluster.

- Alta Disponibilidade
- Nós do proxy dedicado e o controlador de ingresso compartilhado

Alta disponibilidade

Para a alta disponibilidade, os nós principais e do proxy devem ser implementados de modo redundante. As informações a seguir discutem opções para alta disponibilidade no IBM® Cloud Private.

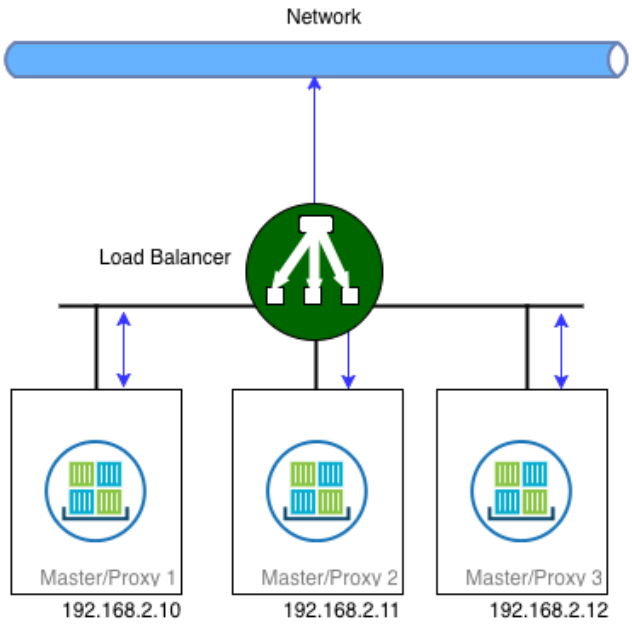
Balanceador de carga externo

Se possível, um balanceador de carga externo altamente disponível, como o F5, pode ser usado para propagar o tráfego entre as instâncias de nó principal ou de proxy separadas no cluster. O balanceador de carga externo pode ser uma URL de DNS ou um endereço IP e especificado usando `cluster_lb_address` em `config.yaml` durante a instalação. O `cluster_CA_domain` e quaisquer certificados TLS devem ser configurados para que sejam um CNAME ou um registro que aponte para o nome do DNS ou endereço IP do balanceador de carga externo. Todos os nós no cluster devem ser capazes de resolver esse CNAME para comunicação interna. Para obter mais informações, consulte Comunicação do Kubelet com o servidor de API.

Quando você está usando um balanceador de carga externo, o balanceador de carga principal monitora a porta 8001 do servidor do Kubernetes API para o funcionamento em todos os nós principais.

- encaminha o tráfego para 8001 (Kubernetes API)
- 8443 (plataforma UI), 9443 (serviço de autenticação)
- 8500 e 8600 (registro privado)

  Quando você estiver usando o balanceador de carga externo, cada nó principal poderá estar em sub-redes diferentes se o tempo de rede de roundtrip entre os nós principais for menor que 33 ms para o etcd. Ainda é possível ter uma separação de obrigações, em que a equipe de rede possui a configuração única e o gerenciamento contínuo de F5 e a integração incorporada do Kubernetes. F5 cria automaticamente a configuração necessária para que o administrador do Kubernetes não precise trabalhar diretamente com o balanceador de carga de F5.



Endereços IP virtuais

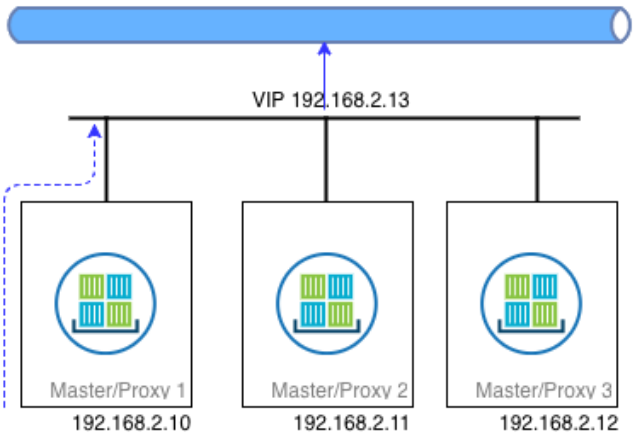
Caso um balanceador de carga esteja indisponível, a alta disponibilidade dos nós principal ou do proxy pode ser obtida usando um endereço IP virtual, que está em uma sub-rede que é compartilhada pelos nós principais e do proxy. O IBM Cloud Private suporta três tipos de soluções de gerenciamento de IP virtual:

1. Etcd (padrão)
2. Ucarp
3. Keep-alive

Essa configuração é feita uma vez como parte da instalação do IBM Cloud Private, ao usar a configuração `vip_manager` em `config.yaml`. Para o `ucarp` e `keepalived`, as propagandas acontecem na interface de gerenciamento e o IP virtual é mantido na interface que é fornecida pelo `cluster_vip_iface` e pelo `proxy_vip_iface`. Em situações em que o IP virtual está aceitando um alto carregamento de tráfego do cliente, a rede de gerenciamento que está executando as propagandas para a eleição principal deve ser separada da rede de dados que está aceitando o tráfego do cliente.

Observe as limitações do uso de um endereço IP virtual:

- A qualquer momento, apenas um nó principal ou do proxy retém o lease para o endereço IP virtual e é representado em uma linha pontilhada na figura. Assim, embora a alta disponibilidade seja obtida usando um IP virtual, a carga do tráfego não é balanceada entre todas as réplicas.
- O uso de um IP virtual requer que todos os nós candidatos usem uma interface `cluster_vip_iface` ou `proxy_vip_iface` na mesma sub-rede.
- Todas as conexões TCP de longa execução ou stateful dos clientes são quebradas durante um failover e devem ser estabelecidas novamente.



Etcd (padrão)

Etcd é um armazenamento de valor da chave distribuído

que é usado internamente pelo IBM Cloud Private para armazenar informações de estado. O Etcd usa um algoritmo de consenso distribuído que é chamado de Raft. O gerenciador de VIP baseado em `etcd` usa o armazenamento de valor da chave distribuído para controlar qual nó principal ou proxy é a instância que retém o endereço IP virtual. O endereço IP virtual é arrendado para o líder, portanto, todo o tráfego é roteado para esse nó principal ou do proxy.

O gerenciador de IP virtual `etcd` é implementado como um cliente etcd que usa um par de valor de chaves. O nó principal ou do proxy atual que está mantendo o endereço IP virtual adquire um lease para esse par de valor de chaves com um TTL de 8 segundos. Os outros nós principais ou do proxy de espera observam o par de valor de chaves do lease. Se o lease expirar sem ser renovado, os nós de espera supõem que o primeiro principal falhou e tentará adquirir seu próprio lease para que a chave seja o novo nó principal. O nó principal que é bem-sucedido e que grava a chave ativa o endereço IP virtual. O algoritmo usa tempo limite de eleição aleatória para reduzir a chance de qualquer condição de corrida, em que mais de um nó tenta se tornar o líder do cluster.

O ARP Gratuitous não é usado com o gerenciador de IP virtual `etcd` quando ele executa failover, portanto, quaisquer conexões do cliente existentes com o endereço IP virtual após o failover falhará até que o cache ARP do cliente expire e que o endereço MAC para o novo portador do IP Virtual seja adquirido. No entanto, o gerenciador de IP virtual `etcd` evita o uso de multicast que o `ucarp` e o `keepalived` requerem.

Ucarp

Ucarp é uma implementação do common address redundancy protocol (CARP) transportada para o Linux®. O Ucarp permite que um nó principal "anuncie" que ele possui um endereço IP específico usando o endereço multicast 224.0.0.18.

Cada nó envia uma mensagem de propaganda em sua interface de rede dizendo que ele pode ter um endereço IP virtual a cada poucos segundos. Essa mensagem é chamada de base de propaganda. Cada nó principal envia um valor de defasagem com essa mensagem do CARP. Isso é semelhante à sua prioridade de manutenção desse IP, que é a defasagem de publicidade (advskew). Se duas ou mais máquinas estiverem efetuando a publicidade em intervalos de 1 segundo (advbase= 1), aquela com o advskew menor 'vencerá'. Quaisquer vínculos são quebrados pelo nó que possui o endereço IP inferior. Para alta disponibilidade, a movimentação de um endereço entre vários nós desta maneira permite que você sobreviva à indisponibilidade de um host, mas isso permite apenas maior disponibilidade e que mais nenhum ajuste de escala seja feito.

Um nó principal se torna principal quando as condições a seguir são atendidas:

1. Ninguém mais anuncia por três vezes seu próprio intervalo de propaganda (advbase).
2. Você especificou `--preempt` e ouve um principal com um intervalo (de propaganda) mais longo (ou com o mesmo advbase, mas com um advskew maior).

O nó principal existente se torna um backup quando as condições a seguir são atendidas:

1. Outro principal anuncia em um intervalo mais curto (ou com o mesmo advbase, mas com um advskew menor).
2. Outro principal anuncia o mesmo intervalo e tem um endereço IP inferior.

Após o failover, o ucarp enviará uma mensagem ARP gratuita para todos os seus vizinhos, para que eles possam atualizar seus caches do ARP com o novo endereço MAC do principal.

Keep-alive

O Keepalived fornece recursos simples e robustos para balanceamento de carga e alta disponibilidade, originalmente usados para alta disponibilidade de roteadores virtuais. O Keepalived usa o Virtual Router Redundancy Protocol (VRRP) como um protocolo de eleição para determinar qual nó principal ou proxy mantém o IP virtual. O gerenciador de IP virtual keepalived implementa um conjunto de verificadores para manter e gerenciar de maneiras dinâmica e adaptável um conjunto de servidores de carga balanceada de acordo com seu funcionamento. O VRRP é um tijolo fundamental para failover. O gerenciador de IP virtual keepalived implementa um conjunto de ganchos para o estado finito do VRRP que fornece interações de protocolo de baixo nível e de alta velocidade.

Para assegurar a estabilidade, o daemon keepalived é dividido em três processos:

1. Um processo-pai chamado watchdog que é responsável pelo monitoramento do processo filhos bifurcados.
2. Um processo-filho para VRRP.
3. Outro processo-filho para a verificação de funcionamento.

A configuração keepalived que é fornecida com o IBM Cloud Private usa o endereço multicast 224.0.0.18 e o protocolo IP número 112. Isso deve ser permitido no segmento de rede em que as propagandas principais são feitas. O keepalived também gera uma senha para autenticação entre os candidatos principais, que é a soma MD5 do IP virtual.

Por padrão, o Keepalived usa o octeto final do endereço IP virtual como o ID do roteador virtual (VRID). Por exemplo, para um endereço IP virtual de 192.168.10.50, ele usa o VRID 50. Se houver quaisquer outros dispositivos que usam o VRRP no segmento de Camada 2 de gerenciamento que está usando esse VRID, poderá ser necessário mudar o endereço IP virtual para evitar conflitos.

Comunicação do kubelet com o servidor da API

Cada nó no cluster executa um agente do nó do kubelet que gerencia a comunicação a partir do plano de controle do Kubernetes. O kubelet se comunica com o cluster_lb_address ou com o `cluster_vip` na porta 8001 (porta do Kubernetes API), as mesmas portas que os clientes da API. Assim, se esta for uma entrada de DNS, ela deverá ser resolvida para o endereço IP virtual ou do balanceador de carga por cada nó no cluster.

Nós do proxy dedicado e o controlador de ingresso compartilhado

A instalação do IBM Cloud Private define algumas funções do nó que são dedicadas à execução do controlador de ingresso do IBM Cloud Private compartilhado chamado nós do proxy. Esses nós servem como um proxy reverso de camada 7 para as cargas de trabalho que estão em execução no cluster. Em situações em que um balanceador de carga externo pode ser usado, tal como um F5, essa é a configuração recomendada. Pode ser difícil proteger e escalar os nós do proxy e o uso de um balanceador de carga evita o hops de rede adicionais por meio de nós do proxy para os pods que estão executando o aplicativo real.

Se um balanceador de carga externo estiver planejado para ser usado, configure o cluster para rotular os nós principais como nós do proxy usando o arquivo `hosts` antes da instalação. Isso marca os nós principais com o rótulo `proxy` adicional e o controlador de ingresso compartilhado é iniciado nos nós principais. Esse controlador de ingresso geralmente pode ser ignorado para o tráfego de northbound ou usado para aplicativos leves southbound expostos, como consoles administrativos adicionais para alguns aplicativos que estão em execução no cluster.

```
[master]
192.168.30.10
192.168.30.11
192.168.30.12

[proxy]
192.168.30.10
192.168.30.11
192.168.30.12
```

Se um controlador de ingresso e recursos de ingresso forem necessários para agregar vários serviços usando os recursos de ingresso integrados, instale controladores de ingresso isolados usando o gráfico Helm incluído para o namespace. Em seguida, exponha esses recursos individualmente por meio do balanceador de carga externo.

Configurações de cluster

Defina suas configurações de cluster para suas necessidades do aplicativo.

- Estendendo o intervalo NodePort padrão

- Isolando ambientes de rede e de cálculo
- Configurando o refletor de rota Calico após a instalação do IBM Cloud Private

Estendendo o intervalo NodePort padrão

É possível aumentar ou mudar o intervalo padrão de portas que são usadas por NodePorts para permitir que portas específicas sejam abertas para suas necessidades do aplicativo.

Por padrão, o intervalo de IP de serviço é de 31000 a 32000. Esse intervalo contém 1.000 portas, o que significa que é possível criar apenas 1.000 recursos de serviço. Se você precisar de mais serviços ou precisar expor portas específicas que não estão nesse intervalo para determinados aplicativos, então é necessário mudar o intervalo do padrão.

Conclua as etapas a seguir em cada nó principal em seu cluster:

1. Faça backup do arquivo `master.json`.

   ```
   cp /etc/cfc/pods/master.json <back_up_location>
   ```

2. No local de backup, edite o arquivo `master.json` incluindo uma linha após o parâmetro `--servicecluster-ip-range` que contém `--service-node-port-range=<start-port>-<endport>` para adequar às necessidades de seu cluster.

3. Inclua uma vírgula no término da linha `--service-cluster-ip-range`.

4. Para ambientes de alta disponibilidade, deve-se atualizar o arquivo `master.json` em cada nó principal um por um. Os serviços do Kubernetes não são interrompidos durante o processo de atualização.

Por exemplo, para mudar o intervalo de portas de 19000 a 22000, faça as atualizações a seguir:

1. Mude `--service-cluster-ip-range=10.0.0.1/24` para incluir uma vírgula: `--service-cluster-iprange=10.0.0.1/24,`

2. Inclua uma linha após o `--service-cluster-ip-range` conforme a seguir: `--service-node-portrange=19000-22000`

3. Copie o arquivo modificado para atualizar o arquivo manifest de pod estático:

   ```
   cp / < back_up_location> /master.json /etc/cfc/pods/
   ```

Quando o arquivo manifest é atualizado, o Kubelet reinicia todos os pods estáticos, que incluem o `kube-apiserver`, o `kube-controller-manager` e o `kube-scheduler`.

Isolando namespaces e proxies após a instalação do IBM Cloud Private

É possível configurar o namespace e o isolamento do proxy depois de instalar o cluster do IBM Cloud Private.

- Ativando o namespace e o isolamento de proxy
- Modificando o isolamento de namespace
- Modificando o isolamento de proxy
- Excluindo o isolamento de namespace
- Excluindo o isolamento do grupo de proxy

Ativando o namespace e o isolamento de proxy

1. Inclua um grupo de hosts. Para obter mais informações, consulte Incluindo um grupo de hosts.
2. Atualize o arquivo `config.yaml` com a configuração para ativar o isolamento do namespace e do proxy. Para obter informações adicionais, consulte Isolando namespaces e proxies durante a instalação do IBM Cloud Private.
3. Execute o comando a seguir para isolar namespaces e proxies:

   ```
   sudo docker run --net=host -t -e LICENSE=accept \ -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 env-isolation
   ```

Modificando o isolamento de namespace

É possível mudar ou incluir grupos de namespaces ou de hosts que são configurados para isolamento de namespace.

1. Atualize o parâmetro `isolated_namespaces:` no arquivo `<installation_directory>/cluster/``config.yaml`. Por exemplo, mude a configuração de `isolated_namespaces: [{ namespace: devops, hostgroup: worker-dev }]` para `isolated_namespaces: [{ namespace: devops, hostgroup: worker-dev-modified }]`.

2. Execute o comando a seguir para implementar as mudanças:

   ```
   sudo docker run --net=host -t -e LICENSE=accept \ -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 env-isolation
   ```

   Depois de executar o comando, o grupo de hosts para o namespace `devops` é modificado para o `worker-dev-modified` no exemplo.

Modificando o isolamento de proxy

É possível mudar ou incluir grupos de namespaces ou de hosts de proxy que são configurados para isolamento de proxy. Conclua estas etapas:

1. Exclua a liberação do Helm `nginx-ingress-<hostgroup-name>`. Para obter mais informações sobre a exclusão de uma liberação de Helm, consulte Gerenciando liberações de Helm.
2. Atualize o parâmetro `isolated_proxies:` no arquivo `<installation_directory>/cluster/``config.yaml`.
3. Execute o comando a seguir para implementar as mudanças:

   ```
   sudo docker run --net=host -t -e LICENSE=accept \ -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 env-isolation
   ```

Excluindo isolamento de namespace

É possível excluir um isolamento de namespace. O namespace não é excluído. Apenas o isolamento do namespace é excluído.

1. Remova o namespace do arquivo `config.yaml` que está na pasta `<installation_directory>/cluster`. Por exemplo, mude a configuração de `isolated_namespaces: [{namespace: production, hostgroup: worker-prod }]` para `isolated_namespaces: [{namespace: test, hostgroup: worker-test}]`.

2. Execute o comando a seguir para remover o isolamento de namespace:

   ```
   sudo docker run --net=host -t -e LICENSE=accept \ -v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 env-isolation
   ```

   Depois de executar o comando, o isolamento do namespace `production` é excluído e o isolamento do namespace `test` é incluído.

Excluindo o isolamento do grupo de proxy

Para excluir o isolamento de um grupo de proxies, conclua estas etapas:

1. Remova a entrada do grupo de proxies que você deseja excluir do parâmetro `isolated_proxies` no arquivo `config.yaml`. Por exemplo, para remover o isolamento de proxy para o grupo de hosts `proxyprod`, mude a configuração de `isolated_proxies: [{namespace: devops, hostgroup: proxydev, lb_address: 172.68.20.11}, {namespace: production, hostgroup: proxyprod}]` para `isolated_proxies: [{namespace: devops, hostgroup: proxydev, lb_address: 172.68.20.11}]` no arquivo `config.yaml`.

2. Exclua a liberação do Helm `nginx-ingress-<hostgroup-name>`. Para obter mais informações sobre a exclusão de uma liberação de Helm, consulte Gerenciando liberações de Helm.

O grupo de hosts de proxy não é excluído. Somente o isolamento de grupo de proxies é excluído.

Configurando o refletor de rota Calico após a instalação do IBM Cloud Private

Em um ambiente com várias zonas, quando não quiser conectividade da Camada 3 nessas zonas, você poderá configurar um refletor de rota Calico.

Para obter mais informações, consulte Implementando o IBM Cloud Private em segmentos isolados da Camada 3.

- Ativando o refletor de rota do Calico
- Modificando a configuração do refletor de rota do Calico
- Desativando o refletor de rota do Calico

Ativando o refletor de rota do Calico

1. Se você não deseja ativar o refletor de rota do Calico no nó principal, é possível incluir um grupo de hosts para o refletor de rota do Calico. Para obter mais informações sobre como incluir grupos de hosts, consulte Incluindo um grupo de hosts.

2. Atualize o arquivo `config.yaml` com a configuração para ativar o refletor de rota do Calico. Para obter mais informações, consulte config.yaml.

3. Execute o comando a seguir para ativar o refletor de rota do Calico:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 calico-rr
```

Modificando a configuração do refletor de rota do Calico

É possível atualizar as informações do grupo de hosts da configuração do refletor de rota do Calico ou é possível incluir grupos de hosts na configuração.

1. Se você estiver incluindo um grupo de hosts, primeiro crie o grupo de hosts. Para obter mais informações sobre como incluir grupos de hosts, consulte Incluindo um grupo de hosts.

2. Atualize o arquivo `config.yaml` com as informações de grupo de hosts novas ou atualizadas. Para obter mais informações, consulte config.yaml.

3. Execute o comando a seguir para atualizar a configuração do refletor de rota do Calico:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 calico-rr
```

Desativando o refletor de rota do Calico

1. Desative o refletor de rota do Calico no arquivo `config.yaml`. Para obter mais informações sobre a configuração do refletor de rota do Calico, consulte config.yaml.

```
management_services:
 calico-route-reflector: "disabled"
```

2. Execute o comando a seguir para desativar o refletor de rota do Calico:

```
sudo docker run --net=host -t -e LICENSE=accept \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:3.2.0 calico-rr
```

3. (Opcional) Remova as informações do grupo de hosts do refletor de rota do Calico. Para obter mais informações, consulte Removendo um nó do cluster do IBM Cloud Private.

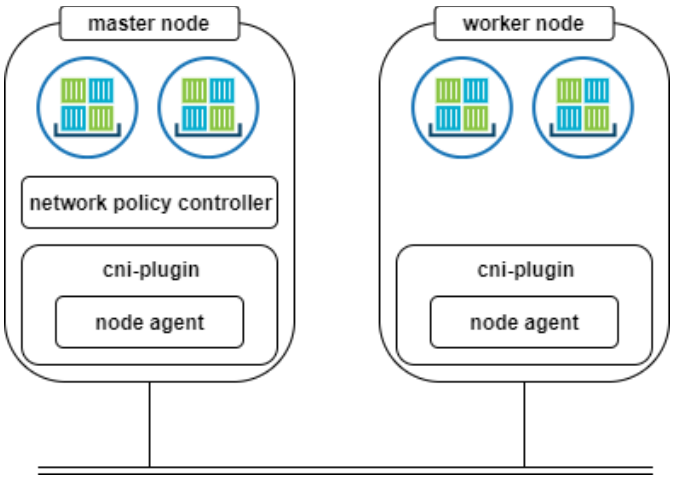
Plug-ins do Container Network Interface (CNI)

O plug-in do Container Network Interface (CNI) é responsável por fornecer uma malha de rede para que os contêineres se comuniquem entre si no cluster.

O sistema de orquestração do contêiner executa o plug-in fornecido pelos provedores CNI para integrar ou implementar diferentes tecnologias de infraestrutura de rede.

Plug-ins do Kubernetes e CNI

Cada plug-in CNI deve ser implementado como um plug-in executável que é iniciado pelo Kubernetes. Os plug-ins executáveis são colocados em `/opt/cni/bin` e a configuração em `/etc/cni/*.conflist`.



Provedores do CNI do Kubernetes

- Calico
- NSX-T

Calico

O Calico é um projeto da comunidade de software livre que fornece rede para contêineres e máquinas virtuais.

O Calico é construído na terceira camada, também conhecida como camada 3 ou a camada de rede, do modelo Open System Interconnection (OSI). Ele usa o Protocolo de Roteamento de Borda (BGP) para construir tabelas de roteamento que facilitam a comunicação entre os nós do agente. Por meio do uso desse protocolo, as redes Calico oferecem melhor desempenho e isolamento de rede.

O Calico implementa o Container Network Interface (CNI) do Kubernetes como um plug-in e fornece agentes para Kubernetes para fornecer rede para contêineres e pods.

O Calico cria uma rede de Camada 3 simples e designa um endereço IP totalmente roteável para cada pod. Ele divide um CIDR de rede grande em blocos menores de endereços IP e designa um ou mais desses blocos menores para os nós no cluster. A divisão é feita durante a instalação do IBM Cloud Private ao usar o parâmetro `network_cidr` em `config.yaml` na notação CIDR.

O Calico, por padrão, cria uma malha BGP entre todos os nós do cluster e transmite as rotas para redes de contêiner para todos os nós do trabalhador. Cada nó é configurado para agir como um gateway de Camada 3 para a sub-rede. A sub-rede é designada ao nó do trabalhador e entrega a conectividade às sub-redes de pod que são hospedadas no host. Todos os nós participam na malha do BGP, que divulga todas as rotas locais que os nós do trabalhador possuem com todos os outros nós. Os peers BGP que são externos ao cluster podem participar, mas o tamanho do cluster afeta a quantidade de propagandas do BGP que esses peers externos recebem. Os refletores de roteamento podem ser necessários quando as escalas de cluster passam um determinado tamanho.

Para obter mais informações, consulte Configurando peers BGP.

Ao rotear o tráfego de pod, o Calico usa os recursos do sistema, como as tabelas de rotas e iptables locais do nó. Todo o tráfego de pod atravessa as regras iptables antes de serem roteadas para seu destino.

O Calico mantém seu estado usando um armazenamento de chave/valor etcd. Por padrão, no IBM Cloud Private, o Calico usa o mesmo armazenamento de valor da chave etcd como Kubernetes para armazenar a política e o estado de configuração de rede.

O Calico pode ser configurado para permitir que os pods se comuniquem entre si com ou sem o tunelamento IP-in-IP. O IP-in-IP inclui um cabeçalho adicional para todos os pacotes como parte do encapsulamento, mas permite que os contêineres se comuniquem em sua rede de sobreposição quase em qualquer rede subjacente que não seja de sobreposição.

Em alguns ambientes em que o espaço de endereço de sub-rede subjacente está restringido e não há acesso para incluir conjuntos de IP adicionais, como em algumas nuvens públicas, o Calico pode ser uma boa opção. No entanto, em ambientes que não requerem sobreposição, o tunelamento IP-in-IP deve ser desativado para remover a sobrecarga do encapsulamento do pacote e permitir que qualquer infraestrutura de roteamento físico faça inspeção de pacote para conformidade e auditoria. Nesses cenários, a rede subjacente pode ser informada sobre as sub-redes de pod adicionais, incluindo os roteadores de rede subjacentes na malha

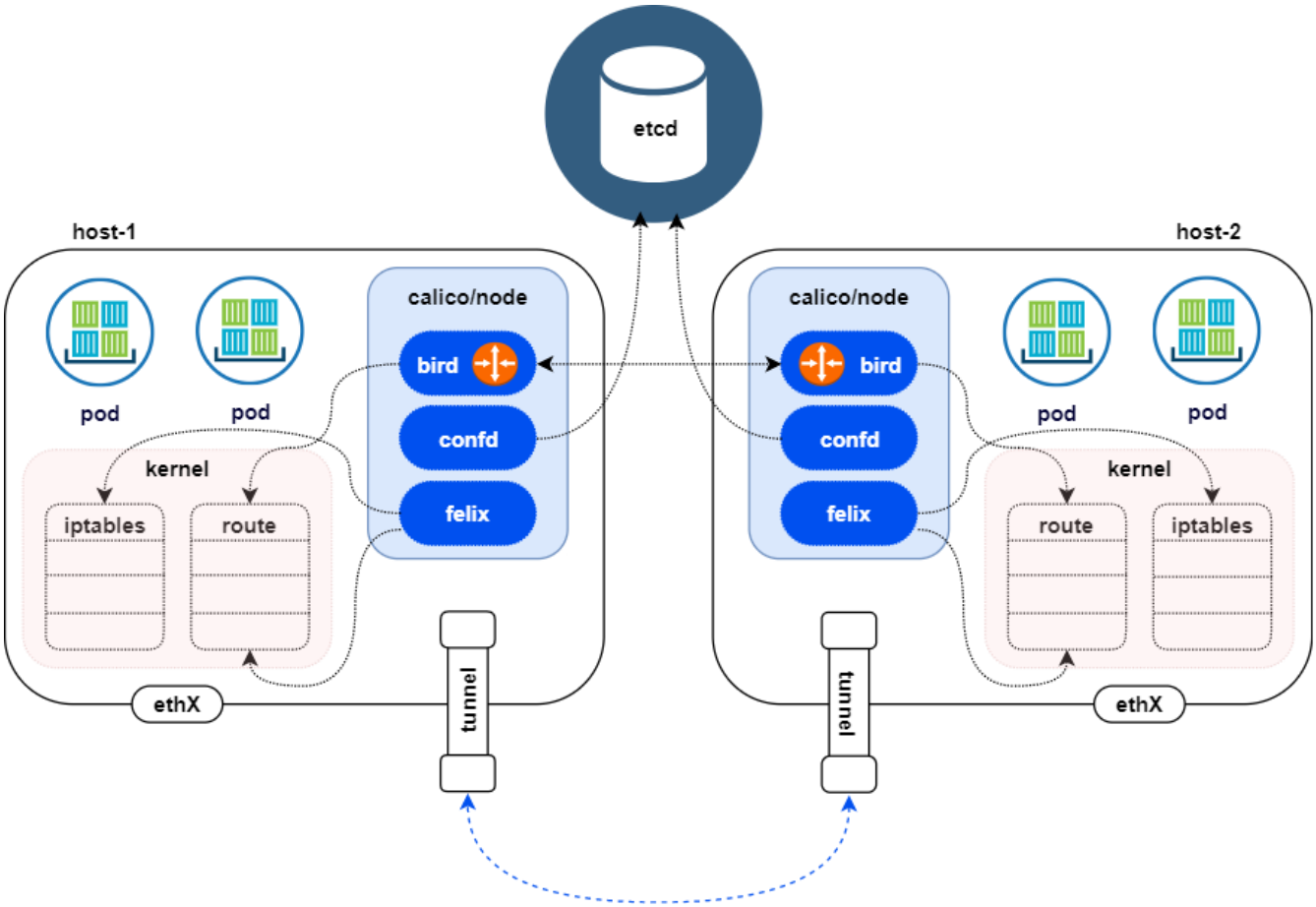
BGP. Para obter mais informações sobre uma rede do Calico quando os nós estão em diferentes segmentos de rede, consulte Rede do Calico em diferentes segmentos de rede.

Componentes do Calico

O Calico possui os componentes a seguir:

1. agente calico/node
2. calico/cni
3. calico/kube-controller

Para assegurar-se de que os nós atendam aos requisitos do sistema Calico, revise as informações em Preparando os nós.



Agente calico/nó

Essa entidade consiste em três componentes - felix, bird e confd.

- A principal responsabilidade do `felix` é programar as IPtables e as rotas do host para fornecer a conectividade que você deseja para e a partir dos pods nesse host.
- `bird` é um agente BGP de software livre para Linux® que é usado para trocar informações de roteamento entre os hosts. As rotas que são programadas pelo `felix` são selecionadas pelo `bird` e distribuídas entre os hosts do cluster.
- O `confd` monitora o armazenamento de dados `etcd` para mudanças na configuração do BGP, como informações do IPAM, número do AS. Ele também muda os arquivos de configuração `bird` e aciona o `bird` para recarregar esses arquivos em cada host. O agente calico/nó cria pares veth para conectar o namespace de rede do pod com o namespace de rede padrão do host.

calico/cni

O plug-in CNI fornece as funções de gerenciamento de endereço IP (IPAM) fornecendo endereços IP para os pods que são hospedados nos nós.

calico/kube-controller

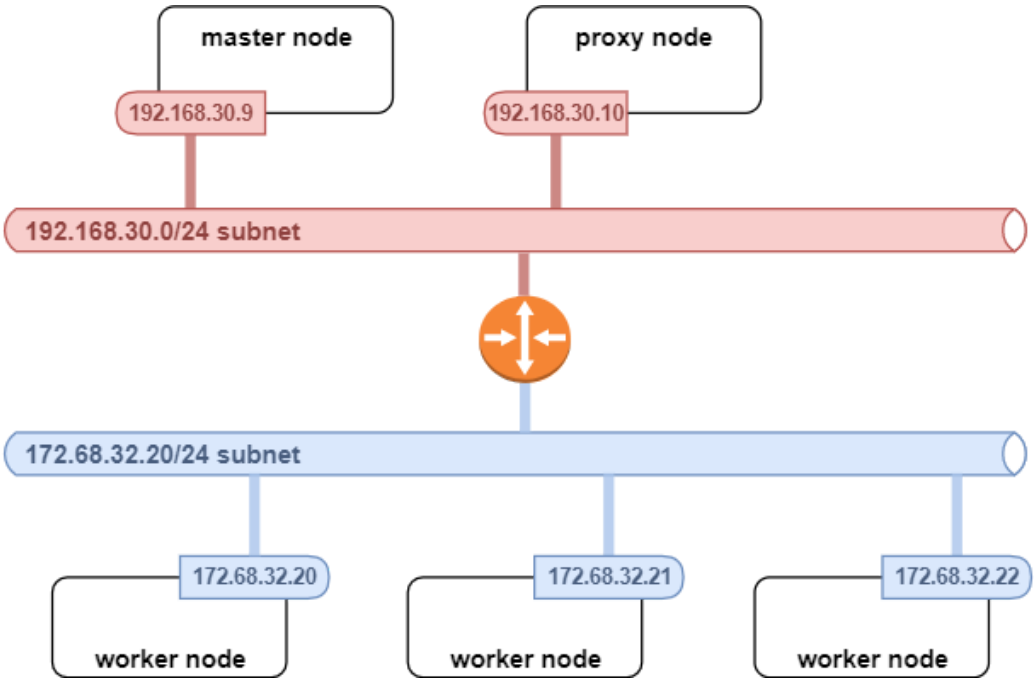
O `calico/kube-controller` observa os objetos NetworkPolicy do Kubernetes e mantém o armazenamento de dados do Calico em sincronia com os objetos do Kubernetes. O `calico/node` que está em execução em cada nó usa as informações no armazenamento de dados `etcd` do Calico para programar os iptables locais.

calicoctl

O `calicoctl` é uma ferramenta de linha de comandos que pode ser usada para gerenciar as políticas de segurança e de rede do Calico e outras configurações do Calico. Ele se comunica diretamente com o `etcd` para manipular o armazenamento de dados. Ele fornece uma série de comandos de gerenciamento de recursos e pode ser usado para solucionar problemas de rede do Calico. Para configurar sua CLI do Calico, consulte Instalando a CLI do Calico (calicoctl).

Rede do Calico através de diferentes segmentos de rede

Quando os nós estão em segmentos de rede diferentes, eles são conectados por um roteador na rede subjacente e de infraestrutura. O tráfego entre dois nós em diferentes sub-redes atravessa o roteador, que é o gateway para as duas sub-redes. Se o roteador não reconhecer a sub-rede do pod, ele não será capaz de encaminhar os pacotes entre os hosts.



Há duas maneiras de manipular este cenário:

1. O Calico pode ser configurado para criar terminais de túnel IP-in-IP em cada nó para cada sub-rede que é hospedada no nó. Qualquer pacote originado pelo pod e que está saindo do nó é encapsulado com o cabeçalho IP-in-IP e o endereço IP do nó é usado como a origem. Dessa forma, o roteador de infraestrutura não vê os endereços IP do pod.

   O tunelamento IP-in-IP promove um rendimento e uma latência de rede extras devido ao processamento de pacote adicional em cada terminal para encapsular e desencapsular os pacotes. No metal bare, a sobrecarga não é significativa, uma vez que determinadas operações de rede podem ser transferidas para as placas da interface de rede. No entanto, em máquinas virtuais, a sobrecarga pode ser significativa e também afetada pelos números de núcleos de CPU e de tecnologias de E/S de rede que são configurados e usados pelos hypervisors. A sobrecarga de encapsulamento de pacote adicional também pode ser significativa quando tamanhos menores de unidade máxima de transmissão (MTU) são usados, já que isso pode apresentar fragmentação de pacote. Os quadros de Jumbo devem ser ativados sempre que possível.

2. A segunda opção é tornar o roteador de infraestrutura ciente da rede do pod. É possível fazer isso ativando o BGP no roteador e incluindo os nós no cluster como os peers BGP. Essas etapas permitem que o roteador e os hosts troquem as informações de rota entre si. O tamanho do cluster nesse cenário pode entrar em jogo como na malha BGP. Cada nó no cluster é um peer do roteador após a ativação do BGP no roteador.

Opções de configuração do IBM Cloud Private para uso da rede de contêiner do

Calico

config.yaml:

```
network_type: calico
network_cidr: 10.1.0.0/16
calico_ipip_mode: Always
calico_tunnel_mtu: 1430
calico_ip_autodetection_method: can-reach={{ groups['master'][0] }}
```

- `network_type`: escolha `calico` para implementar o Calico como a rede do contêiner (o padrão é `calico`)
- `network_cidr`: `network_cidr`: Escolha uma sub-rede IP privada suficientemente grande, que deve ser consumida por todas as cargas de trabalho que estão planejadas para serem hospedadas no cluster. Esse intervalo de IPs não deve entrar em conflito com a rede do host existente ou com o `service_cluster_ip_range`. O número de IPs varia de versão para versão, com base no número de componentes que são incluídos no IBM Cloud Private e no número de componentes e recursos que são instalados. É possível obter a contagem de IPs total atual, listando todos os pods que estão instalados.

  Se você tiver `N` nós do trabalhador e estiver destinado para o número `P` de pods, o intervalo de IP seguro será `(N x P) + N`. Esse intervalo pode tolerar a indisponibilidade do nó `N-1`. O valor padrão é `10.1.0.0/16`, que é suficiente para hospedar um cluster grande.

- `calico_ipip_mode`: esta opção pode ser usada para ativar o tunelamento IP-in-IP. Ele deve ser ativado em ambientes que requerem uma rede de sobreposição, por exemplo, em ambientes nos quais os pacotes de egresso passam por uma verificação rígida para o IP de origem com relação ao IP do host, como o OpenStack. Essa opção também precisará ser ativada se os roteadores de infraestrutura não puderem ser configurados para permitir que as rotas do BGP sejam trocadas.

  Quando esse parâmetro é configurado como `Always`, é necessário permitir o `ip protocol 4` por meio dos firewalls de infraestrutura em todos os nós, por exemplo, Grupos de Segurança no OpenStack.

- `calico_tunnel_mtu`: esse parâmetro configura a MTU apropriada no dispositivo de túnel e na interface pod. Com esse parâmetro, é possível ajustar a MTU de acordo com a MTU do caminho da rede de infraestrutura (PMTU) para evitar a fragmentação de pacotes. Além disso, em determinados casos, os firewalls de infraestrutura não são programados para permitir que as mensagens de erro de PMTU do ICMP sejam propagadas. Como resultado, a sessão L3 não aprende a PMTU, que pode levar à perda de pacote e de conexão.

  O valor padrão é 1430, que é adequado para a maioria dos cenários e redes IaaS, como as redes do OpenStack Neutron VxLAN. Esse valor deve ter pelo menos 20 bytes a menos que a MTU ou a PMTU da interface para acomodar o cabeçalho IP-in-IP.

- `calico_ip_autodetection_method`: esse parâmetro é usado para selecionar a interface no nó e para a comunicação de dados entre pods nos nós. Três métodos estão disponíveis para ajudar a selecionar o caminho de dados para os pods.

  - `first-found`

    Esse método usa o primeiro endereço IP válido em uma interface válida que é localizada primeiro no nó.

  - `interface=<interface name list>`

    Esse parâmetro aceita uma lista separada por vírgulas de nomes de expressões regulares como valor. Ele usa o primeiro endereço IP que está localizado na interface especificada.

    Exemplos:

    ```
    calico_ip_autodetection_method: interface=eth0
    calico_ip_autodetection_method: interface=eth.*
    calico_ip_autodetection_method: interface=eth.*,ens.*
    ```

    Nota: os nomes da interface de rede não podem conter as seguintes sequências: `docker.*`, `cbr.*`, `dummy.*`, `virbr.*`, `lxcbr.*`, `veth.*`, `lo`, `cali.*`, `tunl.*` ou `flannel.*`.

  - `can-reach=<remote IP address or host name>`

    O método `can-reach` usa a tabela de rota local de um nó. Ele também usa a interface que é usada para se comunicar com um endereço de destino especificado para determinar a interface de rede do pod. Esse parâmetro aceita um endereço IP remoto ou o nome de domínio como valor.

    Alguns IPs não são reconhecidos pelo Calico. Assegure-se de que suas interfaces não tenham IPs nos intervalos a seguir:

    - `10.0.2.15/24`: este intervalo de IP é o intervalo de endereços da interface vagrant/virtualbox padrão do NAT.

- `192.168.122.*`: esse intervalo de IPs é o intervalo de endereços da interface da MV libvirt padrão.

Para obter informações sobre as topologias de implementação do IBM Cloud Private que usam o Calico, consulte Topologias de implementação do IBM Cloud Private.

Monitoramento do Calico no Prometheus/Grafana

Por padrão, o componente do Prometheus que está em execução nos nós de gerenciamento extrai o agente do nó Calico que está em execução no IBM Cloud Private para métricas. O IBM Cloud Private inclui um painel Grafana que exibe as métricas de rede do cluster que são recuperadas do Calico na representação gráfica.

Além disso, os alertas podem ser configurados usando essas métricas que são coletadas do Felix.

Instalando a CLI do Calico (calicoctl)

É possível usar a interface da linha de comandos (CLI) do Calico, calicoctl, para gerenciar redes e políticas de segurança do Calico.

À medida que você instala a CLI do Calico, certifique-se de que ela esteja instalada em seu cluster do IBM® Cloud Private em um nó principal, do trabalhador ou proxy.

Também é possível configurar o `calicoctl` a partir de uma estação de trabalho remota que esteja fora do ambiente do IBM Cloud Private.

Para configurar a linha de comandos Calico, conclua as etapas a seguir:

1. Na página *Introdução* da console de gerenciamento do IBM Cloud Private, clique em **Instalar ferramentas da CLI**.
2. Expanda **Instalar a CLI do Calico**. Leia o texto e, em seguida, faça download do instalador usando o comando `curl`.

   Escolha o comando curl para o sistema operacional aplicável. Por exemplo, é possível executar o comando a seguir para macOS, em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal:

   ```
   curl -kLo <install_file> https://<Cluster Master Host>:<Cluster Master API
   Port>/api/cli/calicoctl-darwin-amd64
   ```

   Lembre-se de que o comando curl para seu cluster está localizado na console de gerenciamento.
3. Depois de executar o comando curl para seu sistema operacional, é possível instalar a CLI do Calico. Para configurar a CLI do Calico, execute os comandos a seguir que correspondem à sua arquitetura de nó, em que `<path_to_installer>` é o caminho para o diretório no qual você transferiu por download o arquivo CLI e `<install_file>` é o nome do arquivo transferido por download.
   - Por exemplo, para Linux® e macOS, execute os comandos a seguir para mudar e mover o arquivo.

     ```
     chmod 755 <path_to_installer>/<install_file>
     ```

     ```
     sudo mv < path_to_installer > / < install_file> /usr/local/bin/calicoctl
     ```
   - Para o Windows™, renomeie o arquivo transferido por download para `calicoctl` e inclua o arquivo em sua variável de ambiente PATH.
4. Confirme se a CLI do Calico está instalada.

   ```
   calicoctl -- help
   ```
5. Se você estiver configurando o `calicoctl` em uma estação de trabalho remota, copie os arquivos a seguir do nó principal para sua estação de trabalho:
   - `/etc/cfc/conf/etcd/ca.pem`
   - `/etc/cfc/conf/etcd/client-key.pem`
   - `/etc/cfc/conf/etcd/client.pem`
6. Configure `calicoctl` para usar o armazenamento de dados etcdv3. Use o mesmo `cluster_name` que está no arquivo `config.yaml` no nó de inicialização.
   - Exporte o arquivo de certificado com o comando a seguir:

     ```
     export ETCD_CERT_FILE=/etc/cfc/conf/etcd/client.pem
     ```

- Exporte o arquivo de certificado de CA:

  ```
  export ETCD_CA_CERT_FILE=/etc/cfc/conf/etcd/ca.pem
  ```

- Exporte o arquivo-chave:

  ```
  export ETCD_KEY_FILE=/etc/cfc/conf/etcd/client-key.pem
  ```

- Exporte o domínio de CA com o comando a seguir, em que `<Cluster Master Host>` está definido no Terminal principal:

  ```
  export ETCD_ENDPOINTS=https://<Cluster Master Host>:4001
  ```

Nota: para reter os valores da variável de ambiente entre sessões, é possível incluí-los em um script, tal como `.bashrc`. Consulte o exemplo a seguir. Deve-se copiar o script para todos os nós nos quais você deseja executar os comandos da CLI do Calico:

```
#!/bin/sh
   export ETCD_CERT_FILE=/etc/cfc/conf/etcd/client.pem
   export ETCD_CA_CERT_FILE=/etc/cfc/conf/etcd/ca.pem
   export ETCD_KEY_FILE=/etc/cfc/conf/etcd/client-key.pem
   export ETCD_ENDPOINTS=https://<Cluster Master Host>:4001
```

Para obter mais informações sobre como configurar o calicoctl com o armazenamento de dados etcdv3, consulte Configurando o calicoctl para se conectar a um armazenamento de dados etcd.

7. Use a linha de comandos do Calico. Para iniciar com a linha de comandos do Calico, consulte Referência de comando.

Preparando os nós

Assegure-se de que os nós atendam aos requisitos do sistema Calico.

Para minimizar os problemas de rede do Calico, assegure-se de que os nós no cluster do IBM® Cloud Private atendam aos requisitos a seguir:

- Para redes IPv4 , instale uma versão do kernel Linux® que seja `3.10` ou superior.
- Verifique as configurações de rede. Consulte Configurações de rede para obter detalhes.
- Assegure a conectividade de rede entre todos os nós no cluster. Verifique a conectividade do nó principal com os nós do trabalhador, dos nós do trabalhador com o nó principal e entre todos os nós do trabalhador.
- Assegure-se de que as portas estejam abertas e que nenhuma regra de firewall esteja bloqueando essas portas. Para obter mais informações sobre as portas padrão do IBM Cloud Private, consulte Portas padrão.
- Assegure-se de que o encaminhamento de IP esteja ativado. Para verificar a configuração de redirecionamento de IP, execute o comando a seguir:

  ```
  sysctl net.ipv4.ip_forward
  ```

  Se o encaminhamento de IP estiver ativado, `net.ipv4.ip_forward = 1` será exibido.

  Para ativar o encaminhamento de IP entre reinicializações da sessão e do sistema, inclua `net.ipv4.ip_forward = 1` no arquivo `/etc/sysctl.conf`.

  Para ativar a mudança no arquivo `/etc/sysctl.conf`, execute o comando a seguir:

  ```
  sysctl -p /etc/sysctl.conf
  ```

Topologias de implementação do IBM Cloud Private que usam o Calico

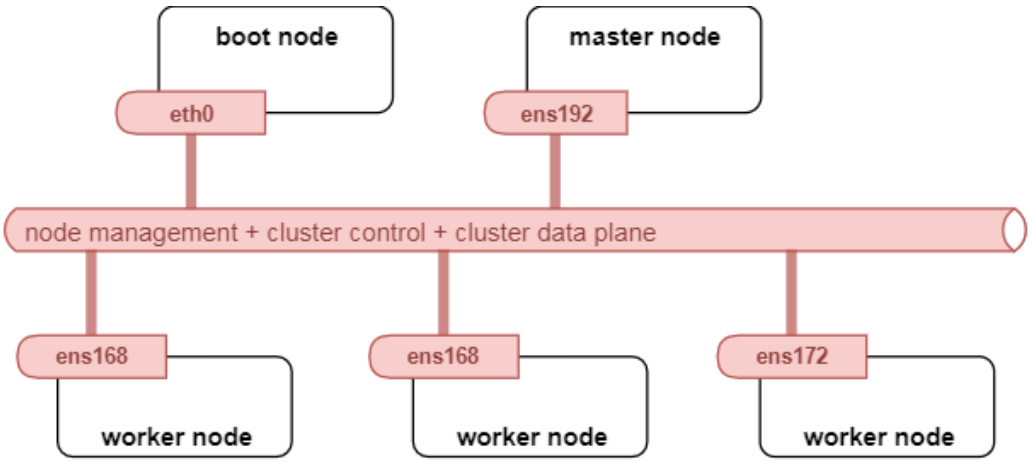
Dependendo de seu ambiente, é possível usar o Calico para diferentes topologias de implementação do IBM Cloud Private.

Rede do Calico por meio de NICs especificados

Por causa da nomenclatura de dispositivo de rede consistente, os controladores de interface de rede (NICs) podem ser nomeados de forma diferente para cada nó do cluster, mas estão fisicamente conectados à mesma rede. Por exemplo:

- master - `ens192`

- worker1 - `ens168`
- worker2 - `ens168`
- worker3 - `ens172`



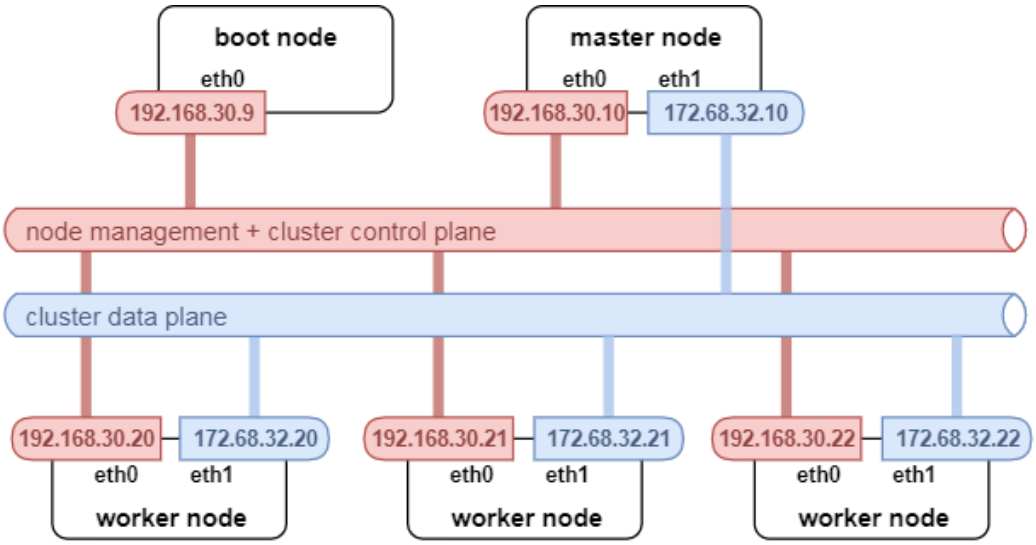
Neste cenário, instale o IBM Cloud Private usando a configuração `config.yaml` a seguir:

```
calico_ip_autodetection_method: interface=ens192,ens168,ens172
```

Nota: se um nó tiver múltiplas correspondências na lista, a primeira será considerada para criar a malha de rede do Calico. Isso pode, no entanto, quebrar a malha de rede do Calico se as interfaces não desejadas forem usadas. Esse método é adequado somente quando uma interface de rede corresponde apenas a um valor na lista.

Rede separada para comunicação entre pods e plano de controle usando vários NICs

Em ambientes em que várias redes estão disponíveis, o IBM Cloud Private pode ser configurado para usar redes físicas separadas para planos de controle e de dados para que o tráfego de gerenciamento do Kubernetes esteja em um segmento de rede de Camada 2 ou 3 separado. Neste cenário, duas redes são configuradas, `192.168.30.0/24` no `eth0` para o tráfego de gerenciamento e `172.68.32.0/24` para o tráfego de aplicativo.



Neste cenário, instale o IBM Cloud Private usando a configuração a seguir:

`cluster/hosts`:

A rede em que cada `kubelet` se comunica com o `kube-apiserver` usa o endereço IP da sub-rede de gerenciamento em `192.168.30.0/24`:

```
[master] 192.168.30.10

[proxy] 192.168.30.10

[worker] 192.168.30.20 192.168.30.21 192.168.30.22
```

`cluster/config.yaml`:

Configure a malha de IP do Calico para usar a interface de rede de dados `eth1` ou para especificar a interface que roteia a rede de dados `172.68.32.0/24`.

```
calico_ip_autodetection_method: interface=eth1
```

Ou

```
calico_ip_autodetection_method: can-reach=<Gateway IP for subnet 172.68.32.0/24>
```

Isolamento de carga de trabalho usando diferentes segmentos de rede de Camada 2

Em cenários com múltiplos locatários ou em cenários em que a conformidade requer que algumas cargas de trabalho precisam ser isoladas umas das outras, implemente cargas de trabalho em segmentos Camada 2 separados. Use um controle de roteador ou de firewall e inspecione o fluxo do tráfego de rede entre eles.

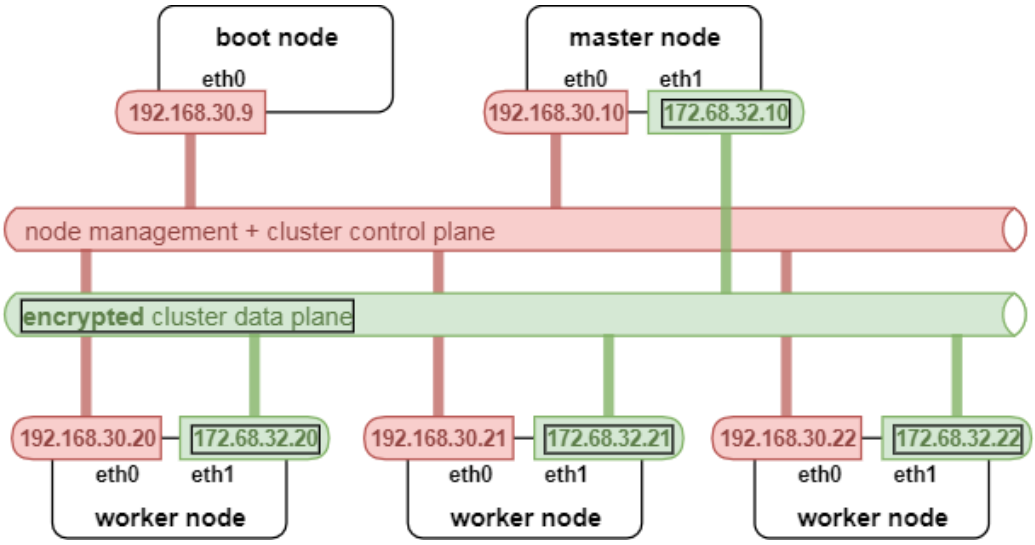
Se os nós do trabalhador e os nós do proxy já estiverem instalados em segmentos de rede de Camada 2 separados, o IBM Cloud Private poderá usar o controlador de admissão `podNodeSelector` para implementar pods em um destino de namespace específico ou em um grupo específico de nós do trabalhador. Nos casos em que a inspeção de pacote stateful é necessária para o tráfego que flui entre grupos de nós do trabalhador, é necessário que o encapsulamento IP-in-IP seja desativado e que o roteador se una à malha BGP.

Para obter mais informações sobre como configurar o IBM Cloud Private durante a instalação para isolamento, consulte Isolando ambientes de rede e de cálculo.

Criptografia de comunicação entre pods com o IPsec

O tráfego de rede pode ser protegido quando todos os aplicativos que estão em execução dentro dos pods implementam a criptografia SSL ou TLS. Esse cenário nem sempre é possível, por exemplo, quando você está executando gráficos Helm de terceiros que não suportam conexões criptografadas por TLS. Para proteger o tráfego entre pods, o IBM Cloud Private pode ser configurado para criptografar a rede de malha de pod usando o IPsec no modo de transporte para que todo o tráfego de pod seja criptografado de forma transparente. Esta configuração pode ser a definição desejada em redes não confiáveis compartilhadas, como uma rede pública.

A ativação da malha de IPsec requer pelo menos duas interfaces de rede; uma para o tráfego de gerenciamento e outra para o plano de dados. O IPsec também requer que o modo IP-in-IP seja ativado e que inclua sobrecarga de CPU adicional para executar a criptografia e a decriptografia do tráfego de pod. Para obter mais informações, consulte Criptografando o tráfego de rede de dados de cluster com o IPsec.



`cluster/hosts`

```
[master] 192.168.30.10

[proxy] 192.168.30.10

[worker] 192.168.30.20 192.168.30.21 192.168.30.22
```

cluster/config.yaml

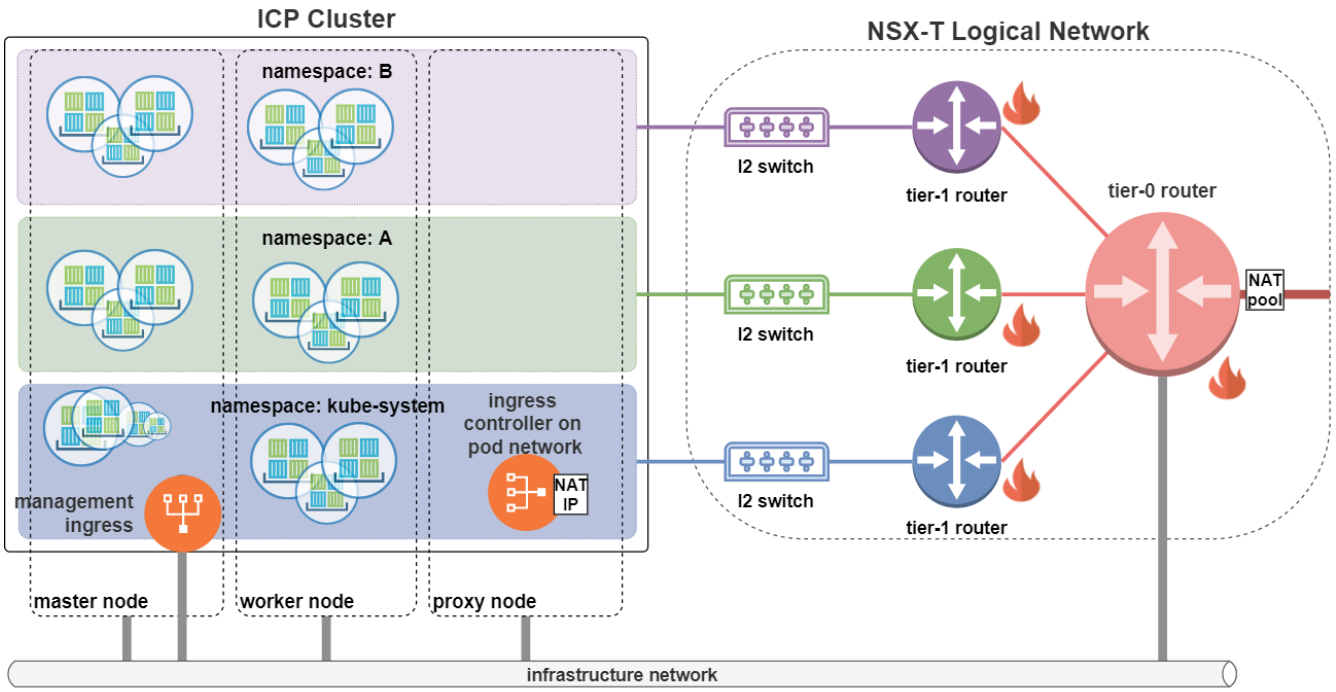
```
calico_ipip_mode: Sempre calico_tunnel_mtu: 1390 calico_ip_autodetection_method: interface=eth1

ipsec_mesh:
  enable: true
  interface: eth1
  subnets: [172.68.32.0/24]
  exclude_ips: [172.68.32.1, 172.68.32.2, 172.68.32.3]
  cipher_suite: aes128gcm16!
```

- O `calico_ipip_mode` deve ser configurado como `Always` para ativar o IPsec.
- `calico_tunnel_mtu` deve ser ajustado de forma que possa acomodar o IP do Calico IP-in-IP (20 bytes) e o cabeçalho IPsec adicional (40 bytes) = 60 bytes. Ele deve ter pelo menos 60 bytes a menos em comparação ao tamanho da MTU da interface de rede de dados fornecida. Para minimizar problemas de desempenho, quadros jumbo (MTU grande) podem ser ativados na interface de rede de dados e no uplink para reduzir a probabilidade de fragmentação de pacote entre os nós.
- `calico_ip_autodetection_method` deve sempre especificar a interface na qual o IPsec está ativado, isto é, a rede de dados.
- `ipsec_mesh.enable` deve ser configurado como `true`.
- `ipsec_mesh.interface` deve ser configurado como o mesmo valor que `calico_ip_autodetection_method: interface =...`. Após a 3.1.1, esse valor é descontinuado e o `calico_ip_autodetection_method: interface =...` é usado para identificar a interface na qual o IPsec está ativado.
- `ipsec_mesh.subnets` é a lista de sub-redes para as quais o IPsec deve ser ativado. É possível fornecer valores separados por vírgulas como `[172.68.32.20/30, 172.68.32.48/29, 172.68.32.100/31]`
- `ipsec_mesh.exclude_ips` é a lista de IPs a serem excluídos da sub-rede IPsec e os pacotes para esses endereços IP não são criptografados. Por exemplo, é possível incluir os terminais de gerenciamento de rede, como o DHCP, o DNS e o Gateway IP, na lista de exclusão para não criptografar pacotes que são destinados a esses terminais. Deixe a lista vazia, caso deseje que todo o tráfego seja criptografado.
- `ipsec_mesh.cipher_suite` especifica os algoritmos de criptografia/autenticação ESP a serem usados durante a criptografia. O padrão para Ubuntu é `aes128gcm16!` e o RHEL é `aes_gcm_c128`. Assegure-se de que os módulos de kernel de criptografia apropriados sejam carregados.

NSX-T

NSX-T é uma plataforma de virtualização e segurança de rede que automatiza a implementação de políticas de rede, objetos de rede, isolamento de rede e microssegmentação.



Virtualização de rede NSX-T para Kubernetes

Segregação L2&L3

O NSX-T cria um comutador L2 separado, o Virtual Distributed Switch (VDS), e o L3, um roteador lógico distribuído (DLR) para cada namespace. O roteador de nível de namespace é chamado de roteador T1. Todos os roteadores T1 estão conectados ao roteador T0, que age como o gateway de borda para o cluster do IBM® Cloud Private e também como o firewall de borda e como o balanceador de carga. Devido ao comutador L2 separado, todo o tráfego de transmissão é confinado ao namespace e, assim como um roteador L3 separado, cada namespace pode hospedar sua própria sub-rede de IP do pod.

Microssegmentação

O NSX-T fornece firewall distribuído (DFW) para gerenciamento do tráfego leste/oeste. As políticas de rede do Kubernetes são convertidas em regras de DFW do NSX-T. Com a segmentação L2 e com sub-redes L3 dedicadas para as políticas de rede de namespace e k8s, é possível atingir a microssegmentação dentro e entre namespaces.

Conjuntos do NAT

O dispositivo Edge é um componente importante do cluster de gerenciamento do NSX-T. Ele oferece roteamento, firewall, balanceamento de carga e conversão de endereço de rede, entre outros recursos. Ao criar os pods na rede de pod do NSX-T (e não depender da rede do host), todo o tráfego pode ser feito para atravessar o dispositivo de borda usando seus recursos de firewall, de balanceamento de carga e de conversão de endereço de rede. O dispositivo de borda designa IPs SNAT para tráfego de saída e IPs DNAT para tráfego de entrada a partir do conjunto NAT (criado como parte da implementação do NSX-T). Dependendo da conversão de endereço de rede, os IPs de nó do cluster não são expostos no tráfego de saída.

Referências para considerações de rede com o NSX-T

Para obter mais informações sobre como integrar o NSX-T ao IBM Cloud Private, consulte Integrando o VMware NSX-T 2.4 ao IBM Cloud Private.

F5 BIG-IP

O F5 BIG-IP Controller fornece uma integração de plataforma nativa de dispositivos BIG-IP com Kubernetes. O BIG-IP Controller for Kubernetes (k8s-bigip-ctlr) configura objetos BIG-IP para aplicativos no cluster do IBM® Cloud Private, servindo o tráfego norte-sul.

O BIG-IP Controller suporta os recursos a seguir:

- Criar, gerenciar e destruir dinamicamente os objetos BIG-IP.
- Encaminhar o tráfego do dispositivo BIG-IP para clusters do Kubernetes por meio de PodIP, NodePort ou ClusterIP.
- Suportar F5 iApps.
- Gerenciar objetos do servidor virtual específicos do F5 que são criados em Kubernetes.
- Gerenciar objetos de ingresso do Kubernetes padrão usando extensões específicas do F5.

Para obter mais informações sobre o F5 BIG-IP Controller for Kubernetes, consulte F5 BIG-IP Controller for Kubernetes.

Topologia

Considere a topologia a seguir para integração:

- Um cluster do IBM Cloud Private com um principal e dois nós do trabalhador. O nome do cluster é `cluster1`.
- Cada um dos nós tem duas placas da interface de rede (NICs): uma na rede de gerenciamento e a outra na rede interna.

O F5 BigIP é geralmente instalado em quatro redes diferentes:

- Rede de HA - o BigIP usa a rede HA para sincronizar o estado entre todos os membros do cluster para manter a Alta Disponibilidade
- Rede de Gerenciamento - o plano de controle; o BigIP usa essa rede para aceitar o tráfego de gerenciamento; o console e a API de REST atendem nesta rede
- Rede Interna - o plano de dados; o BigIP encaminha os tráficos para backends da carga de trabalho que estão em execução nessa rede, por exemplo, contêineres que estão em execução na plataforma IBM Cloud Private
- Rede externa - o IP externo do BigIP aceita conexões nesta rede.

A topologia de rede é semelhante ao diagrama a seguir:

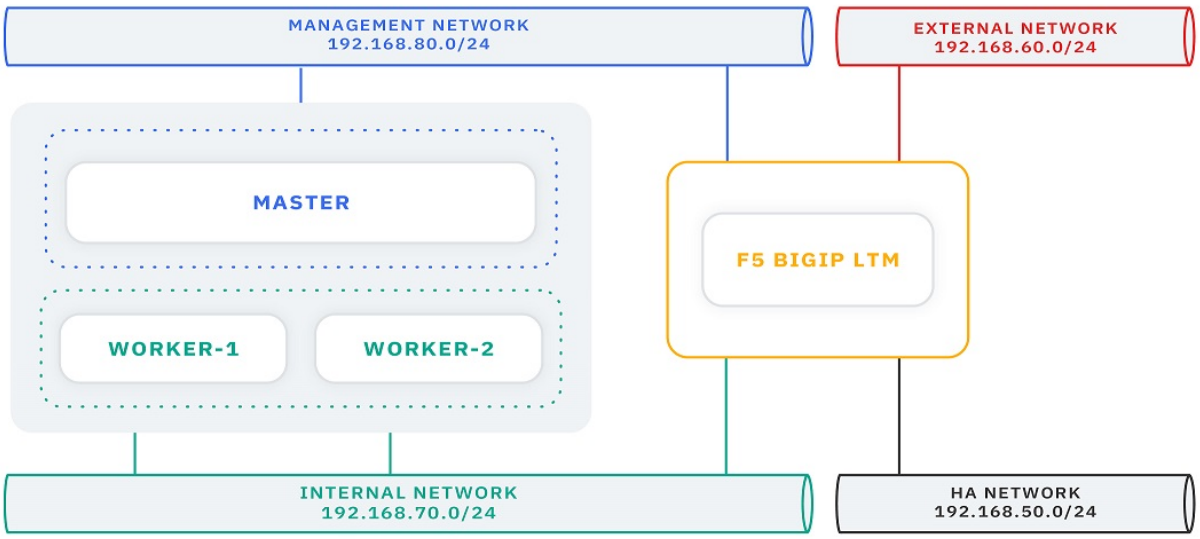


Tabela 1. Topologia de dispositivo de rede IBM Cloud Private e F5 BIG-IP

Tipo	CIDR
Rede Interna	192.168.70.0/ 24
Rede Externa	192.168.60.0/ 24
Rede HA	192.168.50.0/ 24
Rede de Gerenciamento	192.168.80.0/ 24

Tabela 2. IBM Cloud Private informações do nó do cluster

Nó	Endereço de IP
Principal	O endereço IP da rede interna é 192.168.70.225 e o endereço IP da rede de gerenciamento é 192.168.80.225

Nó	Endereço de IP
Nó do trabalhador 1	192.168.70.226
Nó do trabalhador 2	192.168.70.227

Tabela 3. F5 BIG-Informações sobre o dispositivo IP

Tipo	Endereço de IP
Gerenciamento	192.168.80.254
HA	192.168.50.254
Rede externa	192.168.60.254
Rede interna	192.168.70.254

Gerenciador de tráfego local F5 BIG IP

O Gerenciador de tráfego local do F5 BIG IP é uma plataforma de gerenciamento de tráfego que pode servir como um balanceador de carga externo para aplicativos que estão em execução no IBM® Cloud Private.

Ele pode encaminhar o tráfego de Camada 4 para um serviço que esteja em execução IBM Cloud Private ou ser usado como um controlador de ingresso de Camada 7 para o Recursos de ingresso em vez dos nós do proxy. O BigIP está disponível como um dispositivo de hardware ou como uma MV e há também imagens em nuvem para uso em nuvem pública.

Integração do LTM do F5 BIG-IP

O F5 Container Connector pode exportar IPs de pod associados ao serviço do Kubernetes no dispositivo F5 LTM. O F5 Container Connector pode ser implementado como um gráfico do Helm no IBM Cloud Private. O F5 Container Connector observa recursos do Kubernetes que usam a API de dentro do cluster e, em seguida, chama a API de REST do iControl na rede de gerenciamento para criar servidores virtuais no dispositivo F5 LTM.

Nota: a partição padrão do `Common` no dispositivo F5 BigIP não pode ser gerenciada pela integração; deve-se criar uma partição separada.

Para integrar o dispositivo F5 BIG-IP ao seu cluster do IBM Cloud Private, consulte Integrando o IBM Cloud Private ao F5 BIG-IP Controller for Kubernetes.

Quando você estiver usando um dispositivo F5 BigIP, os nós do proxy não precisarão ser implementados. O controlador de ingresso padrão que é implementado com o IBM Cloud Private pode ser ignorado, pois o TLS pode ser finalizado no dispositivo BigIP. Consulte mais informações sobre essa implementação na seção Nós do proxy dedicados e controlador de ingresso compartilhado.

Tipo de rede do conjunto do F5

Quando a integração cria backends para o servidor virtual, ele pode encaminhar o tráfego diretamente para os pods (tipo de rede do conjunto do `cluster`) ou usar o kube-proxy e encaminhar o tráfego para os NodePorts no nó do trabalhador na rede interna (tipo de rede do conjunto `NodePort`). Para o NodePort, uma vez que o S-NAT interno pode ocorrer em nós do trabalhador que não estão executando o pod, o `externalTrafficPolicy` pode ser configurado como `Local` para que o endereço IP do cliente seja preservado. Para obter mais informações, consulte o Tipo de serviço NodePort.

Para encaminhar o tráfego diretamente para os pods na rede interna, é necessário que o BGP aja como um peer do cluster com o roteador ou com o dispositivo F5 BigIP diretamente, para que haja rotas do dispositivo para os pods. Se o Calico for usado, o gráfico `ibm-calico-bgp-peer` poderá ser usado para incluir o dispositivo F5 na malha BGP para que as rotas para os pods sejam preenchidas no dispositivo F5. Para obter mais informações sobre como configurá-lo, consulte Integrando o dispositivo F5 BIG-IP com o IBM Cloud Private.

Expondo serviços do Kubernetes

Para encaminhar o tráfego de Camada 4 do dispositivo para os pods, um recurso ConfigMap que representa um Recurso do F5 é criado que o Controlador BigIP monitora. O controlador do F5 BigIP observa os ConfigMaps com rótulos que correspondem ao `f5type=virtual-server` em todos os namespaces para os quais ele estiver configurado e cria servidores virtuais no dispositivo com base no conteúdo.

Por exemplo, o código a seguir mostra o app de amostra Node.js que está em execução em 3000 na porta 80:

```
kind: ConfigMap
apiVersion: v1
metadata:
 name: nodejs-vs
 namespace: default
 labels:
   f5type: virtual-server
data:
 schema: "f5schemadb://bigip-virtual-server_v0.1.1.json"
 data: |
   {
     "virtualServer": {
       "frontend": {
         "balance": "round-robin",
         "mode": "http",
         "partition": "ICP",
         "virtualAddress": {
           "bindAddr": "172.16.252.180",
           "port": 80
         }
       },
       "backend": {
         "serviceName": "nodejs-test-ibm-nodejs-s",
         "servicePort": 3000
       }
     }
   }
```

Esse tipo de recurso é ideal quando o backend não é um backend HTTP ou para suportar a passagem do TLS para outro proxy, como o gráfico do controlador de ingresso nginx.

Expondo recursos de ingresso

Para encaminhar o tráfego de Camada 7 do dispositivo para os pods, um Recurso de ingresso pode ser usado para incluir vários backends na mesma instância do servidor virtual no dispositivo. Esses recursos podem ter nomes de host ou caminhos diferentes, dependendo das regras que estiverem definidas no recurso de ingresso.

Para expor o recurso de ingresso no dispositivo F5, algumas anotações específicas do F5 indicam ao controlador como programar o F5. Para obter uma lista completa de anotações suportadas, consulte Anexar um servidor virtual a um Ingresso do Kubernetes. Por exemplo, o recurso de ingresso a seguir expõe o app de amostra Node.js no caminho de recurso `/` no servidor virtual padrão na partição do F5 BIG-IP ICP. Um IP específico também pode ser configurado ou, nesse caso, o `controller-default` foi especificado como `172.16.252.180` quando o gráfico do controlador F5 foi instalado.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nodejs
  namespace: default
  annotations:
    virtual-server.f5.com/ip: "controller-default"
    virtual-server.f5.com/partition: "ICP"
spec:
  rules:
  - host: nodejs.csplab.cloudns.cx
    http:
      paths:
      - backend:
          serviceName: nodejs-test-ibm-nodejs-s
          servicePort: 3000
        path: /
```

O uso de recursos de ingresso permite que o mesmo ingresso yaml seja móvel entre as plataformas, por exemplo, o mesmo recurso de ingresso yaml pode ser usado no local, em um dispositivo F5 ou em uma nuvem pública com o controlador de ingresso baseado em nginx padrão.

Além disso, o uso de um recurso de ingresso permite a finalização de TLS no dispositivo F5. A mesma especificação que é definida no recurso de ingresso do Kubernetes padrão é usada. O certificado e a chave TLS são incluídos como um segredo para o Kubernetes. O nome secreto é especificado ou um perfil SSL é armazenado no dispositivo BigIP, se o perfil SSL for especificado como `/<partition>/<profileName>`.

Para evitar que o controlador de ingresso padrão, que é incluído com o IBM Cloud Private, também exponha os mesmos recursos de ingresso que devem ser expostos no F5, inclua a anotação adicional, que faz com que o controlador de ingresso padrão ignore esses recursos.

```
...
metadata:
  annotations:
    kubernetes.io/ingress.class: "f5"
...
```

Integrando o dispositivo F5 BIG-IP ao IBM Cloud Private

Conclua estas etapas para integrar o dispositivo F5 BIG-IP ao seu cluster do IBM® Cloud Private.

Pré-requisitos

- O cluster do IBM Cloud Private deve estar pronto.
- Deve-se ter um dispositivo BIG-IP que seja licenciado e provisionado para seus requisitos.
- Deve-se provisionar e executar o Calico em seu cluster do IBM Cloud Private.

Etapas

1. Configure o dispositivo F5 BIG-IP como um peer para seu cluster do IBM Cloud Private, instalando o gráfico `ibm-calico-bgp-peer` Versão 1.1.0. Ao configurar o dispositivo LTM como um peer para seu cluster do IBM Cloud Private que executa o Calico, o dispositivo LTM se comunica diretamente com os pods.
2. Instale o gráfico `ibm-f5bigip-controller` Versão 1.1.0. O gráfico cria um controlador que observa os objetos do Kubernetes que são criados e comunica os interessados para o dispositivo F5 BIG-IP. Como resultado, o dispositivo F5 BIG-IP cria servidores virtuais apropriados e outros objetos LTM correspondentes.

Configurar o dispositivo F5 BIG-IP como um peer para seu cluster do IBM Cloud Private

Conclua estas etapas para incluir o dispositivo F5 BIG-IP como um peer BGP na malha do Calico em seu cluster do IBM Cloud Private:

1. Efetue login no console de gerenciamento.
2. Clique em **Catálogo** e procure o gráfico `ibm-calico-bgp-peer`.
3. Forneça o endereço IP interno do dispositivo F5 BIG-IP e o número do sistema autônomo (AS). Em um cluster do IBM Cloud Private Calico, o número do AS é configurado para 64512. Se for necessário aplicar o dispositivo como um peer somente a um nó do cluster específico, forneça o endereço IP interno do nó. Caso contrário, deixe o campo em branco para tornar cada nó no cluster ciente desse peer.

4. Configure o contexto para `calicoctl`. Deve-se fornecer a URL de terminal etcd e as credenciais de etcd.

    - `Terminal etcd`: O terminal etcd deve ter o formato `https://<master-node-internal-IP-address>:4001`. Use o comando a seguir para obter as informações sobre o terminal etcd:

      ```
      kubectl get cm etcd-config -ojsonpath={.data.etcd_endpoints} -n kube-system
      ```

      A seguir está uma saída de amostra do comando:

      ```
      https://192.168.70.225:4001
      ```

    - `etcd Secret`: o objeto secreto do Kubernetes com o certificado de CA (etcd-ca), o certificado de cliente (etcd-cert) e a chave privada (etcd-key) que fornecem acesso ao etcd. **Nota:** deve-se criar o objeto secreto do Kubernetes no namespace kube-system.

      ```
      kind: Secret
      metadata:
       name: etcd-secret
       namespace: kube-system
      type: Opaque
      data:
       etcd-ca: LS0......
         .......
         .......
         ..tLQ==
      ```

```
    etcd-cert: MS0......
      .......
      .......
      ..tLQ==
    etcd-key: NS0......
      .......
      .......
      ..tsde2
```

No IBM Cloud Private, é possível usar o comando a seguir para obter as informações secretas de etcd existentes:

```
kubectl get secret etcd-secret -oyaml -n kube-system
```

A seguir está uma saída de amostra do comando:

```
apiVersion: v1
data:
  etcd-ca:
LS0tLS1CRU....LQ==
  etcd-cert:
Q2VydGlm....LS0=
    etcd-key:
LS0tLS1....LS0=
kind: Secret
metadata:
  name: etcd-secret
  namespace: kube-system
type: Opaque
~#
```

5. Forneça os detalhes da imagem `calico-ctl` . Esses detalhes são necessários conforme o comando calicoctl é usado para configurar o peer BGP.
   - `Repository` é o local da imagem.
   - `Tag` é a tag de imagem. No IBM Cloud Private Versão 3.2.0, a tag deve ser `v3.5.2`.
   - `Pullpolicy` é a política de pull de imagem. O valor padrão é `IfNotPresent`.
6. Clique em **Instalar** para instalar o gráfico.
7. Efetue login no dispositivo F5 BIG-IP e execute estes comandos para concluir a configuração do peer BGP:

```
ssh root@<F5-BIG-IP-device-IP-address>
imish
enable
configure terminal
router bgp <AS number>
neighbor <cluster-name> peer-group
neighbor <cluster-name> remote-as <AS number>
neighbor <master node IP address> peer-group <cluster-name>
neighbor <worker node 1 IP address> peer-group <cluster-name>
neighbor <worker node 2 IP address> peer-group <cluster-name>
write
exit
exit
```

Assegure-se de que todos os nós do cluster que você deseja incluir como peers tenham conectividade com o dispositivo F5 BIG-IP. Inclua todos esses nós como peers.

A seguir está uma configuração de exemplo com base na topologia de exemplo:

```
ssh root@192.168.80.254
imish
enable
configure terminal
router bgp 64512
neighbor cluster1 peer-group
neighbor cluster1 remote-as 64512
neighbor 192.168.70.225 peer-group cluster1  # ICP master node
neighbor 192.168.70.226 peer-group cluster1  # ICP worker node 1
neighbor 192.168.70.227 peer-group cluster1  # ICP worker node 2
write
exit
exit
```

8. Verifique se o dispositivo F5 BIG-IP é incluído como um peer BGP na malha do Calico. Verifique o status do nó em qualquer um dos nós do cluster usando o utilitário `calicoctl`.

   ```
   status do nó calicoctl
   ```

   A seguir está uma saída de amostra quando o comando é executado no nó principal:

   ```
   Calico process is running.
   IPv4 BGP status
   +---------------+-------------------+-------+------------+-------------+
   |  PEER ADDRESS  |     PEER TYPE     | STATE |   SINCE    |    INFO     |
   +---------------+-------------------+-------+------------+-------------+
   | 192.168.70.226 | node-to-node mesh | up    | 2018-04-03 | Established |
   | 192.168.70.227 | node-to-node mesh | up    | 2018-04-03 | Established |
   | 192.168.70.254 | global            | up    | 09:46:47   | Established |
   +---------------+-------------------+-------+------------+-------------+
   ```

9. Verifique se a tabela de rotas no dispositivo F5 BIG-IP é atualizada com as rotas para todos os nós em seu cluster do IBM Cloud Private. Agora é possível atingir os IPs do pod diretamente do novo peer BGP.

   ```
   route -n
   ```

   O seguinte é uma saída de amostra:

   ```
   Kernel IP routing table
   Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
   0.0.0.0         192.168.80.1    0.0.0.0         UG    9      0        0 mgmt
   192.168.80.0    0.0.0.0         255.255.255.0   U     0      0        0 mgmt
   10.1.85.192     192.168.70.227  255.255.255.192 UG    0      0        0 internal_vlan
   10.1.145.64     192.168.70.225  255.255.255.192 UG    0      0        0 internal_vlan
   10.1.148.64     192.168.70.226  255.255.255.192 UG    0      0        0 internal_vlan
   192.168.60.0    0.0.0.0         255.255.255.0   U     0      0        0 external_vlan
   192.168.70.0    0.0.0.0         255.255.255.0   U     0      0        0 internal_vlan
   ```

A configuração do peer está concluída. Agora é possível criar o gráfico Helm para integrar o F5 BIG-IP Controller ao IBM Cloud Private.

Instale o gráfico ibm-f5bigip-controller

Instale o gráfico `ibm-f5bigip-controller` para se integrar ao controlador F5 BIG-IP.

1. Efetue login no console de gerenciamento.
2. Navegue para o Catálogo e procure o gráfico `ibm-f5bigip-controller`.
3. Forneça os atributos necessários para o gráfico.
   - `URL` é o endereço IP de gerenciamento do dispositivo F5 BIG-IP.
   - `Partition Name` é a partição do BIG-IP na qual você configura objetos. Certifique-se de que uma partição seja gerenciada por meio de somente um controlador do F5 BIG-IP Kubernetes. O gerenciamento da mesma partição por meio de vários controladores do F5 BIG-IP Kubernetes pode levar a um comportamento inesperado.
   - `Username` é o nome do usuário REST do BIG-IP iControl.
   - `Password` é a senha REST do BIG-IP iControl.
   - `Pool Member Type` é o tipo de membro do conjunto BIG-IP que você deseja criar. Os valores válidos são `cluster` ou `nodeport`. Use `cluster` para criar membros do conjunto para cada um dos terminais para o serviço, que é `pod`. Use `nodeport` para criar membros do conjunto para cada nó no cluster do Kubernetes e para confiar no `kube-proxy` no nível do nó para solicitações de balanceamento de carga para os pods.
   - `Default Ingress IP` é o endereço IP usado pelo controlador para configurar um servidor virtual para todos os ingressos com a anotação: virtual-server.f5.com/ip: `controller-default`.
   - `Namespace(s)` é a lista de namespaces do Kubernetes a serem observados. Por exemplo, `[ "ns1", "ns2" ]`.
   - `Node Label Selector` é o rótulo do nó. O controlador do Kubernetes BIG IP observa somente os nós com o rótulo especificado.
   - `Extra Arguments` são outras opções do controlador do Kubernetes BIG IP. Forneça um mapa na forma de `{"key":"value", …}`.
4. Forneça os valores de parâmetros `NodeSelector` e `Tolerations`.
   - `NodeSelector` é o nó no qual o controlador deve ser executado. Por exemplo, se você desejar que o pod do controlador seja colocado no nó principal, forneça `{"role": "master"}`.
   - `Tolerations` é usado para planejar o controlador para um nó com contaminações correspondentes. Por exemplo, `[{"key":"dedicated","operator":"Exists","effect":"NoSchedule"}, {"key":"CriticalAddonsOnly","operator":"Exists"}]`.
5. Clique em **Instalar** para instalar o gráfico.

6. Verifique se o pod do controlador do F5 BIG IP está sendo executado com êxito e observando os recursos nos namespaces necessários.

   ```
   kubectl get po -n <release-namespace>
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME                                     READY     STATUS    RESTARTS   AGE
   c1-f5bigipctlr-ctlr-7d8759cd7f-z8zm4   1/1       Running   0          52m
   ```

A integração do dispositivo F5 BIG-IP com o IBM Cloud Private está concluída. Agora é possível criar servidores virtuais e balancear a carga dos pods diretamente de seu dispositivo F5 BIG-IP.

Configuração de exemplo de criação de servidores virtuais do IBM Cloud Private

1. Crie um aplicativo com o nome `myapp` no namespace que o controlador do Kubernetes BIG IP está observando.

   ```
   kubectl get po -n <release-namespace> -owide | grep myapp
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME          READY   STATUS    RESTARTS   AGE   IP            NODE
   myapp-kggt8   1/1     Running   0          1h    10.1.148.66   192.168.70.226
   myapp-z5grr   1/1     Running   0          1h    10.1.85.226   192.168.70.227
   ```

2. Crie um serviço para expor seu aplicativo.

   ```
   kubectl get svc -n <release-namespace>
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
   myapp     ClusterIP   10.0.0.20    <none>        8080/TCP   56s
   ```

3. Crie um configmap para criar um servidor virtual de front-end específico do serviço e, se necessário, os conjuntos no sistema BIG-IP.

   ```
   cat f5_configmap.yaml
   kind: ConfigMap
   apiVersion: v1
   metadata:
    name: myapp-vs
    labels:
      f5type: virtual-server
   data:
    schema: "f5schemadb://bigip-virtual-server_v0.1.1.json"
    data: |
      {
        "virtualServer": {
          "frontend": {
            "balance": "round-robin",
            "mode": "http",
            "partition": "icp",
            "virtualAddress": {
              "bindAddr": "192.168.60.10",
              "port": 80
            }
          },
          "backend": {
            "serviceName": "myapp",
            "servicePort": 8080
          }
        }
      }
   ```

   Verifique se o configmap foi criado:

   ```
   kubectl get cm -n <release-namespace>
   ```

   A seguir está uma saída de exemplo:

   ```
   NAME                         DATA      AGE
   myapp-vs                     2         2s
   ```

4. Depois de criar o configmap, é possível ver os servidores virtuais e conjuntos no dispositivo F5 BIG-IP. Abra a UI do dispositivo F5 BIG-IP e verifique os locais a seguir:
    - Para servidores virtuais, consulte **F5 BIG-IP UI** > **<Partition Name>** > **Tráfego local** > **Servidores virtuais**.
    - Para conjuntos, consulte **F5 BIG-IP UI** > **<Partition Name>** > **Tráfego local** > **Conjuntos**.
5. Verifique se as solicitações para o serviço têm carga balanceada.

    Primeira solicitação para o serviço:

    ```
    curl 192.168.60.10
    ```

    A seguir está uma saída de exemplo:

    ```
    Hostname: myapp-kggt8
    Pod Information:
            node name:      192.168.70.226
            pod name:       myapp-kggt8
            pod namespace:  f5
            pod IP: 10.1.148.66
    ```

    Segundo pedido para o serviço:

    ```
    curl 192.168.60.10
    ```

    A seguir está uma saída de exemplo:

    ```
    Hostname: myapp-z5grr
    Pod Information:
            node name:      192.168.70.227
            pod name:       myapp-z5grr
            pod namespace:  f5
            pod IP: 10.1.85.226
    ```

Guia de armazenamento

Saiba como configurar e gerenciar o armazenamento em seu cluster.

- Armazenamento da plataforma
- Armazenamento do aplicativo

Armazenamento da plataforma

Quando o seu cluster possui múltiplos nós do orientador principal ou de vulnerabilidade, você configura uma pasta compartilhada. A pasta compartilhada requer armazenamento da plataforma.

Os nós principais usam a pasta compartilhada para imagens e arquivos de log de auditoria. As imagens do Docker que são enviadas por push para um principal são disponibilizadas para os outros nós principais por meio da pasta compartilhada.

Armazenamento de aplicativo

Provisione armazenamento persistente para as cargas de trabalho que são hospedadas em seu cluster do IBM® Cloud Private.

Um dos principais recursos dentro de qualquer cluster é armazenamento ou volume persistente. Para poder implementar a maioria dos aplicativos baseados em Kubernetes, deve-se primeiramente criar o armazenamento a partir da infraestrutura subjacente. Esse armazenamento provisionado permite a persistência de dados de um aplicativo. No entanto, para que um aplicativo use esse armazenamento, deve-se também criar um volume ou uma solicitação de volume persistente. Solicitações de volume persistente são usadas para fazer solicitação em nome de um aplicativo para o armazenamento provisionado.

- Entendendo o armazenamento do Kubernetes
- Planejando uma solução de armazenamento
- Planejando o armazenamento persistente
- Opções de armazenamento no IBM Cloud Private

Entendendo o armazenamento do Kubernetes

Com a abstração do volume do Kubernetes, é possível criar armazenamento persistente para pods, bem como compartilhar arquivos entre contêineres dentro de um pod.

Para obter mais informações, consulte Entendendo os conceitos básicos de armazenamento do Kubernetes.

O volume que você especifica no campo `.spec.volumes` de uma definição de pod é usado para o pod. O volume é montado no contêiner no local que você especifica no campo `.spec.containers.volumeMounts`.

Volume persistente e solicitação de volume persistente

O Kubernetes fornece as Interfaces de Programação de Aplicativo (APIs) `PersistentVolume` e `PersistentVolumeClaim` para obter detalhes abstratos de como o armazenamento é fornecido e consumido em uma carga de trabalho do aplicativo.

Um **`PersistentVolume`** (PV) é uma parte do armazenamento no cluster que captura os detalhes da implementação do armazenamento. Um PV é provisionado por um administrador de cluster.

Um **`PersistentVolumeClaim`** (PVC) é uma solicitação de armazenamento. É possível usar um PVC para consumir recursos de armazenamento abstratos especificando o tamanho e o modo de acesso.

Para obter mais informações, consulte Volumes persistentes.

Se você tiver um PV provisionado em seu cluster com uma capacidade de 500 Gi, um PVC que tenha uma solicitação de 700 Gi permanecerá desvinculado em seu cluster até que um PV seja provisionado que atenda a essa demanda. Quando um PV está disponível que atenda a esta demanda, o PVC se liga, então, a esse PV disponível e estará pronto para uso.

Para obter mais informações sobre como usar PVs no IBM Cloud Private, consulte PersistentVolume.
Para obter mais informações sobre como usar PVCs no IBM Cloud Private, consulte PersistentVolumeClaims.

Fornecimento estático

Um administrador de cluster pode usar o fornecimento estático para tornar os dispositivos de armazenamento existentes disponíveis para um cluster. O administrador de cluster cria uma série de PVs que estão disponíveis para consumo. O administrador do cluster deve saber os detalhes do dispositivo de armazenamento, suas configurações suportadas e as opções de montagem.

Fornecimento dinâmico

É possível usar fornecimento dinâmico para solicitar armazenamento on demand. Com o fornecimento dinâmico, o administrador de cluster não precisa pré-provisionar o armazenamento. O fornecimento dinâmico usa a classe de armazenamento.

Para obter mais informações sobre o fornecimento dinâmico no IBM Cloud Private, consulte Fornecimento de armazenamento dinâmico.

Classe de armazenamento

Uma classe de armazenamento é usada para provisionar volumes dinamicamente quando há uma solicitação para um volume. Ele resume a plataforma de armazenamento subjacente para que você não tenha que saber os detalhes da plataforma de armazenamento. O administrador de cluster fornece classes de armazenamento predefinidas para cada tipo de armazenamento que é suportado em seu cluster. Os administradores também podem especificar uma classe de armazenamento padrão para os PVCs que não solicitam nenhuma classe de armazenamento específica.

Para obter mais informações, consulte Classes de armazenamento.

Política de recuperação

Uma política de recuperação libera um volume para reutilização. Quando você não precisa mais de um volume, é possível excluir os objetos PVC, que permitem que o volume seja recuperado. Uma política de recuperação de um PV informa ao cluster o que fazer com o volume depois que ele é liberado de sua solicitação. Atualmente, os volumes podem ser retidos, reciclados ou excluídos.

Modo de acesso

O Kubernetes suporta três tipos de modos de acesso para PVs: `ReadWriteOnce`, `ReadOnlyMany` e `ReadWriteMany`.

`ReadWriteOnce (RWO)` - o volume pode ser montado como leitura/gravação por um único nó

`ReadOnlyMany (ROX)` - o volume pode ser montado como de leitura por vários nós

`ReadWriteMany (RWX)` - o volume pode ser montado como leitura/gravação por vários nós

Container Storage Interface

O Container Storage Interface (CSI) define uma interface padrão para o Kubernetes para incluir qualquer armazenamento arbitrário para a carga de trabalho de contêiner. O CSI está disponível como Beta e é ativado por padrão em seu cluster do IBM Cloud Private.

Para obter mais informações sobre o CSI, consulte Plug-ins de volumes fora da árvore.

Para obter mais informações sobre como usar o CSI no IBM Cloud Private, consulte Container Storage Interface (CSI).

- PersistentVolume
- PersistentVolumeClaims
- Fornecimento de armazenamento dinâmico
- Container Storage Interface (CSI)

PersistentVolume

Saiba como gerenciar PersistentVolumes.

- Criando um PersistentVolume
- Excluindo um PersistentVolume

Criando um PersistentVolume

Configure o armazenamento que está disponível para todos os contêineres no cluster.

Para visualizar uma lista de PersistentVolume, no menu de navegação, clique em **Plataforma** > **Armazenamento**.

Dois formatos estão disponíveis para você criar um PersistentVolume por meio da console de gerenciamento.

É possível criar PersistentVolumes inserindo os valores de parâmetro na caixa de diálogo Criar PersistentVolume ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Criando PersistentVolumes usando a caixa de diálogo Criar PersistentVolume

1. No menu de navegação, clique em **Plataforma** > **Armazenamento**.
2. Clique em **Criar PersistentVolume**.
3. Insira os detalhes de armazenamento.

   Para criar o armazenamento, os parâmetros a seguir são necessários:

   - Nome
   - Capacidade
   - Modo de acesso
   - Política de recuperação
   - Tipo de armazenamento - se você estiver usando a caixa de diálogo Criar PersistentVolume para criar um novo armazenamento, apenas NFS, GlusterFS, hostPath ou vSphere poderá ser selecionado. Para usar outros tipos de armazenamento, use a janela "Criar recurso".
   - Parâmetros para armazenamento - esse parâmetro depende do tipo de armazenamento selecionado. Por exemplo, se você escolher o armazenamento NFS, será necessário especificar o servidor e o caminho para o armazenamento.
4. Clique em **Criar**.

Criando PersistentVolumes usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar um volume persistente, consulte a seção *Volumes persistentes* na página Conceitos do Kubernetes.
3. Clique em **Criar**.

Após a conclusão da implementação, um novo PersistentVolume será exibido na lista. Revise o status do PersistentVolume. O volume deve estar em estado `Available`.

- Criando um PersistentVolume NFS
- Criando um PersistentVolume do Glusterfs
- Criando um PersistentVolume do hostPath

Criando um PersistentVolume NFS

Crie um PersistentVolume do sistema de arquivos de rede (NFS).

Pré-requisito

- Um servidor NFS deve estar configurado e disponível. Para obter mais informações sobre as versões suportadas do NFS no IBM® Cloud Private, consulte Sistemas de arquivos e armazenamento suportados.
- O pacote do cliente NFS deve ser instalado em todos os nós em seu cluster do IBM Cloud Private.
  - No Ubuntu, execute os comandos a seguir:

    ```
    sudo apt-get update

    sudo apt-get install nfs-common
    ```

  - No Red Hat Enterprise Linux (RHEL), execute o seguinte comando:

    ```
    yum install nfs-utils
    ```

Criando um PersistentVolume NFS usando a caixa de diálogo Criar PersistentVolume

É possível criar um PersistentVolume NFS em seu cluster, então os contêineres em seu aplicativo poderão usá-lo para persistência de dados.

Para visualizar uma lista de PersistentVolume, no menu de navegação, clique em **Plataforma** > **Armazenamento**.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Criando um PersistentVolume NFS usando a janela Criar PersistentVolume

Para configurar um PersistentVolume NFS:

1. No menu de navegação, clique em **Plataforma** > **Armazenamento**.
2. Clique em **Criar PersistentVolume**.
3. Insira os detalhes do PersistentVolume.

   Para criar um PersistentVolume NFS, os parâmetros a seguir são necessários:

Guia Geral

- Nome - forneça um nome para o PersistentVolume.
- Capacidade
- Modo de acesso
- Política de recuperação
- Tipo de armazenamento - selecionar NFS

Guia Parâmetros

- Parâmetros para armazenamento. Os parâmetros são fornecidos como pares de chaves e valores. Para NFS, deve-se especificar:
  - Um servidor
    - Chave: - servidor
    - Valor: - o nome do host do servidor NFS ou IP.
  - Um caminho
    - Chave: - caminho
    - Valor: - o local do diretório no servidor NFS que está montado como um diretório compartilhado.

4. Clique em **Criar**.

Criando um PersistentVolume NFS usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar uma cota usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistent-volumes.
   Um arquivo YAML simples pode ser semelhante ao seguinte texto:

```
apiVersion: v1
kind: PersistentVolume
metadata:
   name: nfsvol01
spec:
capacity:
   storage: 40Gi
accessModes:
   - ReadWriteOnce
persistentVolumeReclaimPolicy: Recycle
nfs:
   path: <insert/path/to/share>
   server: <insert_nfs_server_ip_or_name>
```

3. Click **Create**.

Após a conclusão da implementação, um novo PersistentVolume do NFS é exibido na lista de PersistentVolume. Revise o status do PersistentVolume. O PersistentVolume deve ter um status `Disponível`.

Criando um PersistentVolume do GlusterFS

Os contêineres no aplicativo podem usar o PersistentVolume do GlusterFS para a persistência de dados.

É possível fornecer dinamicamente um volume GlusterFS utilizando uma classe de armazenamento. Para obter mais informações, consulte Criando uma classe de armazenamento para GlusterFS.

Ou pode-se criar manualmente um PersistentVolume do GlusterFS no cluster, concluindo as etapas das seções a seguir.

Para visualizar uma lista de PersistentVolumes no cluster, no menu de navegação, clique em **Plataforma** > **Armazenamento**.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Criando um terminal e um serviço

Crie um terminal e um serviço somente se você estiver usando um servidor GlusterFS configurado fora do ambiente do IBM Cloud Private.

Os itens a seguir são necessários antes de poder concluir esta tarefa:

- O cluster de servidores GlusterFS com dois ou mais servidores GlusterFS deve estar disponível para acesso por meio de seu cluster do IBM Cloud Private.
- Um armazenamento GlusterFS deve ser criado.
- O cliente GlusterFS deve ser instalado em todos os nós do IBM Cloud Private:

- No Ubuntu, execute os comandos a seguir:

  ```
  sudo apt-get update

  sudo apt-get install glusterfs-client
  ```

- No Red Hat Enterprise Linux (RHEL), execute o seguinte comando:

  ```
  sudo yum install glusterfs-client
  ```

- No RHEL Linux® on Power® (ppc64le), primeiro crie um repositório, em seguida, instale o cliente GlusterFS:

  - Crie um repositório:

    ```
    [centos-gluster40]
    name=CentOS-$releasever - Gluster 4.0
    baseurl=https://buildlogs.centos.org/centos/7/storage/$basearch/gluster-4.0/
    gpgcheck=0 enabled=1
    ```

  - Instale o cliente do GlusterFS:

    ```
    sudo yum install glusterfs-client
    ```

- Crie um terminal para o cluster de servidores GlusterFS.

  YAML de amostra

  ```
   ---
  kind: Endpoints
  apiVersion: v1
  metadata:
   name: glusterfs-cluster
  subsets:
  - addresses:
   - ip: 9.111.249.161
   ports:
   - port: 1729
  - addresses:
   - ip: 9.111.249.162
   ports:
   - port: 1729
  ```

  O endereço `IP` que você especifica deve ser o endereço de um nó no cluster de servidores do GlusterFS.

- Crie o terminal no Kubernetes:

  ```
  kubectl create -f glusterfs-endpoints.yaml
  ```

- Verifique se os terminais foram criados com êxito:

  ```
  kubectl get endpoints
  ```

  Resultado da amostra:

  ```
  NAME ENDPOINTS glusterfs- glusterfs-cluster 9.111.249.161:1729,9.111.249.162:1729 1h
  ```

- Crie um serviço para o terminal. A criação de um serviço para o terminal permite que o terminal seja persistido. Incluir um serviço sem um seletor informa ao Kubernetes que os terminais são incluídos manualmente.

  YAML de amostra

  ```
  ---
  kind: Service
  apiVersion: v1
  metadata:
   name: glusterfs-cluster
  spec:
   ports:
   - port: 1729
  ```

- Crie o serviço no Kubernetes.

  ```
  kubectl create -f glusterfs-service.yaml
  ```

Criando um PersistentVolume do GlusterFS usando a caixa de diálogo Criar PersistentVolume

1. Efetue login no console de gerenciamento IBM Cloud Private.
2. No menu de navegação, clique em **Plataforma** > **Armazenamento**.
3. Clique em **Criar PersistentVolume**.
4. Insira os detalhes do PersistentVolume. Os detalhes podem ser fornecidos em um formato JSON ou preenchendo os campos na caixa de diálogo Criar PersistentVolume.

   Para criar um PersistentVolume, os parâmetros a seguir são necessários:

Guia Geral

- Nome - forneça um nome para o PersistentVolume.
- Capacidade
- Modo de acesso
- Política de recuperação
- Tipo de armazenamento - selecione Glusterfs

Guia Parâmetros

- Parâmetros para armazenamento. Os parâmetros são fornecidos como pares de chaves e valores. É necessário especificar:
  - Um terminal
    - Chave: - terminais
    - Valor: - o nome do terminal que foi criado na etapa 1.
  - Um caminho
    - Chave: - caminho
    - Valor: - o nome do armazenamento GlusterFS.

5. Clique em **Criar**.

Um novo PersistentVolume do GlusterFS é exibido na lista PersistentVolume. Revise o status do PersistentVolume. O PersistentVolume deve ter um status `Available`.

Criando um PersistentVolume do GlusterFS usando a janela "Criar recurso"

1. Efetue login no console de gerenciamento IBM Cloud Private.
2. No painel, clique em **Criar recurso**.
3. Copie e cole um arquivo YAML ou da JSON na janela "Criar recurso". Para obter mais informações sobre como criar um PersistentVolume usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistent-volumes.
4. Clique em **Criar**.

Após a conclusão da implementação, um novo PersistentVolume será exibido na lista. Revise o status do PersistentVolume. O volume deve estar em estado `Available`.

Criando um PersistentVolume hostPath

Crie um PersistentVolume hostPath.

Um volume hostPath monta um arquivo ou diretório no sistema de arquivos do nó de host em seu pod. Para obter mais informações sobre o volume hostPath, consulte Tipos de volumes.

Um PersistentVolume hostPath deve ser usado apenas em um cluster de nó único. O Kubernetes não suporta hostPath em um cluster com diversos nós atualmente.

É possível criar um PersistentVolume hostPath usando a console de gerenciamento.

Para visualizar uma lista de volumes persistentes em seu cluster do IBM® Cloud Private, no menu de navegação, clique em **Plataforma** > **Armazenamento**.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Criando um PersistentVolume hostPath usando a caixa de diálogo Criar PersistentVolume

Para criar um PersistentVolume hostPath:

1. No menu de navegação, clique em **Plataforma** > **Armazenamento**.
2. Clique em **Criar PersistentVolume**.
3. Insira os detalhes do PersistentVolume.

   Para criar um PersistentVolume hostPath, especifique os parâmetros a seguir:

Guia Geral

- Nome - (obrigatório) Forneça um nome para o PersistentVolume.
- Capacidade - (obrigatório) Forneça a capacidade de armazenamento.
- Modo de acesso - (opcional) Selecione `ReadWriteOnce`
- Política de recuperação - (opcional) Selecione `Retain` ou `Recycle`.
- Tipo de armazenamento - (obrigatório) Selecione `Host path`.

Label tab

Attach a label to the PersistentVolume. Um rótulo é um par de chave e valor. Por exemplo: `type=local`. Esse parâmetro é opcional.

Guia Parâmetros

- Parâmetros para armazenamento. Os parâmetros são fornecidos como pares de chaves e valores. Para hostPath, deve-se especificar:
  - Um caminho
    - Chave: - caminho
    - Valor: - o local do arquivo ou diretório no nó de seu cluster. Por exemplo: `/tmp/data`

4. Clique em **Criar**.

Criando um PersistentVolume hostPath usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou da JSON na janela "Criar recurso". Para obter mais informações sobre como criar um PersistentVolume usando um arquivo YAML, consulte Criar um PersistentVolume.

   Um YAML simples pode ser semelhante a este exemplo:

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: hostpath2
  labels:
    type: local
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  reclaimPolicy:
    - Recycle
  hostPath:
    path: "/tmp/data1"
```

3. Clique em **Criar**.

Após a conclusão da implementação, um novo PersistentVolume hostPath é exibido na lista de PersistentVolume. Revise o status do PersistentVolume. O PersistentVolume deve ter um status `Available`.

Excluindo um PersistentVolume

Remova um PersistentVolume que não é mais necessário.

Assegure-se de que nenhum aplicativo esteja usando o PersistentVolume que você deseja remover.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Para remover um PersistentVolume:

1. No menu de navegação, clique em **Plataforma** > **Armazenamento**. Uma lista de PersistentVolumes é exibida.
2. Na lista, selecione **Ação** > **Remover** para o PersistentVolume que você deseja excluir.

O PersistentVolume selecionado é removido da lista.

PersistentVolumeClaims

PersistentVolumeClaims podem ser usados para solicitar armazenamento para um aplicativo.

Os PersistentVolumeClaims devem existir no mesmo namespace que o aplicativo que está usando o PersistentVolumeClaim.

Os PersistentVolumeClaims são usados para acessar seu armazenamento associado. Ao criar um aplicativo e anexar um PersistentVolumeClaim, o PersistentVolume que está associado a essa solicitação é, então, montado no host e disponibilizado para o contêiner do pod que fez a solicitação.

- Criando um PersistentVolumeClaim
- Conectando PersistentVolumeClaims a um aplicativo
- Excluindo um PersistentVolumeClaim

Criando um PersistentVolumeClaim

É possível criar PersistentVolumeClaims para alocar armazenamento para seu aplicativo.

Use essa tarefa para criar PersistentVolumeClaims para seu aplicativo. Antes de poder criar um PersistentVolumeClaims, um PersistentVolume deve estar disponível em seu cluster. Para obter mais informações sobre como gerenciar o armazenamento, consulte Armazenamento.

Um PersistentVolume disponível é ligado a um PersistentVolumeClaim e pode ser usado por um aplicativo. Cada PersistentVolume pode ser ligado apenas a um PersistentVolumeClaim.

É possível criar PersistentVolumeClaims inserindo os valores de parâmetro na caixa de diálogo Criar PersistentVolumeClaim ou colando um arquivo YAML na janela "Criar recurso".

Para visualizar uma lista de PersistentVolumeClaim, no menu de navegação, clique em **Plataforma** > **Armazenamento** > **PersistentVolumeClaim**.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando PersistentVolumeClaims usando a caixa de diálogo Criar PersistentVolumeClaim

1. No menu de navegação, clique em **Plataforma** > **Armazenamento**.
2. Selecione **PersistentVolumeClaim**.
3. Clique em **Criar PersistentVolumeClaim**.
4. Insira os detalhes de PersistentVolumeClaim na caixa de diálogo Criar PersistentVolumeClaim.

   Para criar um PersistentVolumeClaim, os parâmetros a seguir são necessários:

   - Nome - fornece um nome para o PersistentVolumeClaim.
   - Solicitações de armazenamento - quantidade de armazenamento necessária.
   - Modo de acesso - Para volumes que suportam vários modos de acesso, deve-se especificar o modo requerido.
5. Clique em **Criar**.

Criando PersistentVolumeClaims usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou da JSON na janela "Criar recurso". Para obter mais informações sobre como criar um PersistentVolumeClaim usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims.
3. Clique em **Criar**.

Se um PersistentVolumeClaim for criado com êxito, um novo PersistentVolumeClaim será exibido na lista PersistentVolumeClaims. Revise o status. O PersistentVolumeClaim deve ter um status igual a `Bound`.

Conectando PersistentVolumeClaims a uma implementação

Forneça armazenamento a uma implementação.

Use essa tarefa para criar novas implementações com PersistentVolume que está ligado a um PersistentVolumeClaim.

PersistentVolumeClaims podem ser conectados a uma implementação somente durante a criação inicial da implementação.

Para visualizar uma lista de PersistentVolumeClaim, no menu de navegação, clique em **Plataforma** > **Armazenamento** > **PersistentVolumeClaim**.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Para anexar um PersistentVolumeClaim a uma implementação:

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Clique em **Criar implementação**.
3. Na caixa de diálogo Criar implementação, selecione a guia **Volumes**.
4. Forneça as informações a seguir para o volume:
   - Um nome - forneça um nome para o volume.
   - Um volume - forneça o nome do PersistentVolumeClaim que você deseja usar. Esse PersistentVolumeClaim deve existir e estar em um estado ligado. Para obter mais informações sobre como criar um PersistentVolumeClaim, consulte Criando um PersistentVolumeClaim.
   - Um caminho de montagem - este caminho de montagem é o local que você deseja usar dentro do contêiner.
5. Insira os detalhes de implementação necessários. Para obter mais informações sobre como implementar uma nova implementação, consulte Criando uma implementação.
6. Clique em **Criar**. É possível verificar os detalhes do pod para verificar se o PersistentVolume está montado.

Excluindo um PersistentVolumeClaim

Remover PersistentVolumeClaims que não são mais necessários.

Use essa tarefa para remover PersistentVolumeClaims de um cluster.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Para remover um PersistentVolumeClaim:

1. No menu de navegação, clique em **Plataforma** > **Armazenamento**.
2. Selecione **PersistentVolumeClaim**.
3. Selecione **Ação** > **Remover** para o PersistentVolumeClaim que você deseja remover.

O PersistentVolumeClaim selecionado é removido da lista.

Fornecimento de armazenamento dinâmico

O fornecimento dinâmico permite que volumes de armazenamento sejam criados on demand. Use classes de armazenamento para provisionar volumes.

- Criando uma classe de armazenamento
- Excluindo uma classe de armazenamento

Criando uma Classe de Armazenamento

Crie uma classe de armazenamento.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Ao criar uma classe de armazenamento, considere essas classes de armazenamento e sua qualidade de serviço (QoS) associada.

Tabela 1. Classes de armazenamento

Nome da classe de armazenamento	QoS ((Input/output operations per second (IOPS)/tamanho)
ibmc-file-bronze	IOPS 2
ibmc-file-silver	IOPS 4
ibmc-file-gold	IOPS 10
ibmc-file-custom	Tamanho da variável/e IOP
ibmc-block-bronze	IOPS 2
ibmc-block-silver	IOPS 4
ibmc-block-gold	IOPS 10
ibmc-block-custom	Tamanho da variável/e IOP

Criando classes de armazenamento durante a instalação do IBM Cloud Private

É possível criar uma classe de armazenamento durante a instalação do IBM Cloud Private. Após a instalação ser concluída, a classe de armazenamento é criada e pode ser usada para fornecimento de armazenamento dinâmico.

Nota: somente classes de armazenamento do Kubernetes são suportadas.

Conclua estas etapas para fornecer um arquivo de definição de classe de armazenamento:

1. Crie um arquivo `YAML` com as definições de classe de armazenamento. Para obter informações sobre definições de classe de armazenamento, consulte a documentação do Kubernetes.
2. Salve o arquivo `YAML` na pasta `/<installation_directory>/cluster/misc/storage_class`.

Nota: não coloque nenhum outro arquivo no local `/misc/storage_class`. O instalador do IBM Cloud Private seleciona apenas os arquivos `YAML` da classe de armazenamento do diretório e ignora qualquer outro arquivo.

Exemplos de classe de armazenamento

Para obter instruções para criar classes de armazenamento GlusterFS e vSphere no IBM® Cloud Private, consulte Criando uma classe de armazenamento para GlusterFS e Criando uma classe de armazenamento para o volume do vSphere.

Para obter uma lista de várias definições de classe de armazenamento, consulte Classes de armazenamento.

Classe de armazenamento básico

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name:
provisioner: kubernetes.io/<plug-in-type>
parameters:
  parameter 1: value
  ...
  parameter N: value
```

Em que:

- `kind: StorageClass` é a API.
- `apiVersion: storage.k8s.io/v1` é a versão da API.
- `metadata: name:` é o nome da classe de armazenamento.
- `provisioner: kubernetes.io/<plug-in-type>` é o nome do fornecedor de armazenamento.
- `parameters` descrevem os volumes que pertencem à classe de armazenamento. Os parâmetros são opcionais e variam com o fornecedor.

GlusterFS

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: slow
provisioner: kubernetes.io/glusterfs
parameters:
  resturl: "http://127.0.0.1:8081"
  clusterid: "630372ccdc720a92c681fb928f27b53f"
  restauthenabled: "true"
  restuser: "admin"
  secretNamespace: "default"
  secretName: "heketi-secret"
  gidMin: "40000"
  gidMax: "50000"
  volumetype: "replicate:3"
```

Em que:

- `resturl` é uma URL de serviço REST de Gluster ou uma URL de serviço Heketi obrigatória que provisiona volumes Gluster on demand. O formato geral é `IP address:Port`. Se o serviço Heketi for exposto como um serviço roteável na configuração do Kubernetes, o `resturl` poderá ter um formato como `http://heketi-storage-project.cloudapps.mystorage.com`, em que o FQDN é uma URL de serviço Heketi resolvível.
- `restauthenabled` é um Booleano de autenticação de serviço REST do Gluster que permite a autenticação no servidor REST. Se esse valor for `true`, será necessário especificar `restuser` e `restuserkey` ou `secretNamespace` + `secretName`. Essa opção está descontinuada; a autenticação é ativada quando algum desses parâmetros é especificado:
  - `restuser`
  - `restuserkey`
  - `secretName`
  - `secretNamespace`
- `restuser` é o usuário do serviço REST de Gluster ou o usuário Heketi que pode criar volumes no Conjunto Confiável do Gluster.
- `restuserkey` é a senha do usuário do serviço REST de Gluster ou o usuário Heketi a ser usada para autenticação no servidor REST. Essa opção é descontinuada em favor de `secretNamespace` + `secretName`.
- `secretNamespace` e `secretName` são opcionais. Eles identificam uma instância secreta que contém a senha a ser usada para se comunicar com o serviço REST de Gluster. Se `secretNamespace` e `secretName` não forem especificados, uma senha vazia será usada. O segredo fornecido deve ter o tipo “kubernetes.io/glusterfs“.
- `clusterid: 630372ccdc720a92c681fb928f27b53f` é opcional. Esse ID é o ID do cluster que Heketi usa para provisionar o volume. Ele também pode ser uma lista de IDs de cluster separados por vírgulas.
- `gidMin` e `gidMax` são opcionais. Eles são os valores mínimo e máximo do intervalo de GID para a classe de armazenamento. Um valor exclusivo (GID) nesse intervalo (gidMin - gidMax) é usado para volumes fornecidos dinamicamente. Se `gidMin` e `gidMax` não forem especificados, o volume será provisionado com um valor que está no intervalo de 2000 a 2147483647, que são padrões para `gidMin` e `gidMax`, respectivamente.
- `volumetype` é o tipo de volume e seus parâmetros. O tipo de volume é opcional; se ele não for especificado, o fornecedor decidirá o tipo de volume. Exemplos: `‘Replica volume’: volumetype: replicate:3`, em que `3` é a contagem de réplicas; `‘Disperse/EC volume’: volumetype: disperse:4:2`, em que `4` são os dados e `2` é a contagem de redundância; `‘Distribute volume’: volumetype: none`.

vSphere

- Crie um PersistentVolume com um formato de disco especificado pelo usuário.

  ```
  kind: StorageClass
  apiVersion: storage.k8s.io/v1
  metadata:
   name: fast
  provisioner: kubernetes.io/vsphere-volume
  parameters:
   diskformat: zeroedthick
  ```

  Em que o valor de `diskformat` é `thin`, `zeroedthick` ou `eagerzeroedthick`. O valor padrão é `thin`.

- Crie um PersistentVolume com um formato de disco em um armazenamento de dados especificado pelo usuário.

  ```
  kind: StorageClass
  apiVersion: storage.k8s.io/v1beta1
  metadata:
  ```

```
 name: fast
provisioner: kubernetes.io/vsphere-volume
parameters:
 diskformat: zeroedthick
 datastore: VSANDatastore
```

    - `diskformat`: `thin`, `zeroedthick` ou `eagerzeroedthick`. Padrão: `thin`.
        - `datastore`: também é possível especificar o armazenamento de dados no `StorageClass`. O volume é criado no armazenamento de dados que é especificado na classe de armazenamento, que nesse caso é `VSANDatastore`. Este campo é opcional. Se ele não for especificado, como na descrição de YAML anterior, o volume será criado no armazenamento de dados que é especificado no arquivo de configuração do vSphere que é usado para inicializar o vSphere Cloud Provider.

- Crie um PersistentVolume com capacidades de armazenamento VSAN especificadas pelo usuário.

  ```
  kind: StorageClass
  apiVersion: storage.k8s.io/v1beta1
  metadata:
    name: vsan-policy-fast
  provisioner: kubernetes.io/vsphere-volume
  parameters:
    diskformat: thin
    hostFailuresToTolerate: "1"
    diskStripes: "2"
    cacheReservation: "20"
    datastore: VSANDatastore
  ```

  É possível especificar recursos de armazenamento VSAN para fornecimento de volume dinâmico dentro do Kubernetes.

  As políticas de armazenamento capturam requisitos de armazenamento, como desempenho e disponibilidade, para PersistentVolumes. Essas políticas determinam como os objetos de armazenamento do volume de contêiner são fornecidos e alocados no armazenamento de dados para assegurar a qualidade de serviço requerida. As políticas de armazenamento são compostas por recursos de armazenamento, que normalmente são representados por um par chave-valor. A chave é uma propriedade específica que o armazenamento de dados pode oferecer. O valor é uma métrica ou um intervalo que o armazenamento de dados assegura para um objeto provisionado, tal como um volume de contêiner que é apoiado por um disco virtual.

  O VSAN expõe diversos recursos de armazenamento. Esta tabela lista os recursos de armazenamento do VSAN que o vSphere Cloud Provider suporta:

Tabela 1. Capacidade de armazenamento VSAN

Nome da capacidade de armazenamento	Descrição
cacheReservation	Reserva do cache de leitura flash
diskStripes	Número de faixas do disco por objeto
forceProvisioning	Forçar fornecimento
hostFailuresToTolerate	Número de falhas para tolerar
iopsLimit	Limite de IOPS para o objeto
objectSpaceReservation	Reserva de espaço objeto

Enquanto o administrador de infraestrutura vSphere cria uma classe de armazenamento dentro do Kubernetes, ele pode especificar requisitos de armazenamento para aplicativos em termos de recursos de armazenamento. Quando o administrador cria um StorageClass, ele deve especificar os nomes de capacidade de armazenamento que são usados na tabela 1 porque esses nomes podem ser diferentes daqueles que o VSAN usa. Por exemplo, *Número de faixas de disco por objeto* é referido como *stripeWidth* na documentação do VSAN. No entanto, vSphere Cloud Provider usa o nome *diskStripes*.

- Criando uma classe de armazenamento para GlusterFS
- Criando uma classe de armazenamento para o volume do vSphere

Criando uma classe de armazenamento para o volume do vSphere

Crie uma classe de armazenamento para fornecer PersistentVolume em um armazenamento de dados vSphere.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Pré-requisito

o IBM® Cloud Private deve ser configurado com um vSphere Cloud Provider. Consulte Configurando um vSphere Cloud Provider.

Crie uma classe de armazenamento

Para criar uma classe de armazenamento para um volume do vSphere, especifique os valores de campo a seguir na definição de classe de armazenamento:

- metadados:
  - `name`: nome do objeto da classe de armazenamento.
- parâmetros:
  - `diskformat`: `thin`, `zeroedthick` ou `eagerzeroedthick`.
  - `datastore`: nome do armazenamento de dados. O volume é criado no armazenamento de dados que é especificado na classe de armazenamento.
  - `storagePolicyName`: nome da política de armazenamento que você criou em sua configuração do vSphere. Um volume persistente (PV) é provisionado dinamicamente com base na política de armazenamento. Para obter mais informações sobre o gerenciamento baseado em política de armazenamento (SPBM), consulte Gerenciamento baseado em política de armazenamento para o fornecimento dinâmico de volumes.

A seguir há um exemplo de como criar uma classe de armazenamento:

1. Crie um arquivo `YAML` com as definições de classe de armazenamento:

   ```
   vim vsphere-volume-storage-class-1.yaml
   ```

   ```
   kind: StorageClass
   apiVersion: storage.k8s.io/v1
   metadata:
    name: storage-class-1
   provisioner: kubernetes.io/vsphere-volume
   parameters:
    diskformat: thin
    datastore: datastore-1
    storagePolicyName: vsanStoragePolicy
   ```

2. Crie a classe de armazenamento:

   ```
   kubectl create -f vsphere-volume-storage-class-1.yaml
   ```

   A saída se assemelha ao código a seguir:

   ```
   storageclass "storage-class-1" created
   ```

3. Verifique se a classe de armazenamento está criada:

   ```
   kubectl describe sc storage-class-1
   ```

   A saída se assemelha ao código a seguir:

   ```
   Name:           storage-class-1
   IsDefaultClass: No
   Annotations:    <none>
   Provisioner:    kubernetes.io/vsphere-volume
   Parameters:     datastore=datastore-1,diskformat=thin,storagePolicyName=vsanStoragePolicy
   Events:         <none>
   ```

Criando uma classe de armazenamento para GlusterFS

Crie uma classe de armazenamento para provisionar armazenamento do GlusterFS.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Para criar uma classe de armazenamento para GlusterFS, especifique os valores de campo a seguir na definição de classe de armazenamento:

- metadados:
  - `name` é o nome da classe de armazenamento.
    Nota: O nome deve consistir em caracteres alfanuméricos minúsculos e deve começar e terminar com um caractere alfanumérico. É possível usar apenas esses caracteres especiais no nome: `-` e `.`.

- parâmetros:
  - `resturl` é a URL REST do Heketi que provisiona volumes.

    `https://<Heketi_service_cluster_IP>:<Heketi_service_port>`

    - Para obter o serviço Heketi, execute o comando a seguir:

      ```
      kubectl -n kube-system get service  -l glusterfs=heketi-service
      ```

    - Para obter o IP do cluster de serviço Heketi, execute o comando a seguir:

      ```
      kubectl -n kube-system get service <heketi-service-name> -
      o=jsonpath='{.spec.clusterIP}'
      ```

    - Para obter o número da porta de serviço Heketi, execute o comando a seguir:

      ```
      kubectl -n kube-system get service <heketi-service-name> -
      o=jsonpath='{.spec.ports[0].port}'
      ```

  - O `volumetype` é o parâmetro opcional para o tipo de volume. Os valores de parâmetro válidos são `none`, `replicate:<replicate_count>` e `disperse:<data>:<redundancy_count>`. Se você não especificar um tipo de volume, o fornecedor configurará o tipo de volume para `replicate:3`.
  - O `volumenameprefix` é um prefixo para o nome do volume. Por padrão, volumes provisionados dinamicamente possuem o esquema de nomenclatura do formato vol_ . Com o parâmetro `volumenameprefix` na classe de armazenamento, é possível prefixar o nome do volume desejado.

Para criar uma classe de armazenamento para GlusterFS, deve-se concluir as etapas que são mostradas no exemplo a seguir:

1. Crie um arquivo YAML que seja denominado `glusterfs.yaml` e que contenha as definições de classe de armazenamento a seguir:

   ```
   apiVersion: storage.k8s.io/v1
   kind: StorageClass
   metadata:
     name: glusterfs
   provisioner: kubernetes.io/glusterfs
   parameters:
     resturl: "https://10.0.0.185:8080"
     restuser: "admin"
     secretName: "heketi-secret"
     secretNamespace: "kube-system"
     volumetype: replicate:3
     volumenameprefix: "icp"
   ```

   Nota: se o seu cluster GlusterFS estiver em um ambiente do IBM® Cloud Private, será necessário usar os valores de parâmetros a seguir:

   - **restuser** deve ser "admin"
   - **secretName** é o mesmo segredo que você usou para autenticação Heketi durante a instalação do GlusterFS
   - **secretNamespace** deve ser "kube-system"

   Para tornar isso a sua classe de armazenamento padrão, inclua a anotação `storageclass.kubernetes.io/is-default-class` e configure-a como `true`.

2. Crie a classe de armazenamento:

   ```
   kubectl criar -f glusterfs.yaml
   ```

   A saída se assemelha ao código a seguir:

   ```
   storageclass "glusterfs" criado
   ```

3. Verifique se a classe de armazenamento está criada:

   ```
   o kubectl descreve sc glusterfs
   ```

Excluindo uma Classe de Armazenamento

Excluir uma classe de armazenamento que não é mais necessária.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Use o comando da linha de comandos Kubernetes a seguir (kubectl) para excluir uma classe de armazenamento. Para obter informações adicionais sobre como instalar e configurar o `kubectl`, consulte Acessando o cluster a partir da CLI do Kubernetes (kubectl).

```
kubectl delete storageclass <storage class name>
```

A seguir está um comando de exemplo e a saída:

```
$ kubectl delete storageclass glusterfs-distributed
storageclass "glusterfs-distributed" deleted
```

Interface de Armazenamento de Contêiner (CSI)

O Container Storage Interface (CSI) está agora disponível como Beta no Kubernetes v1.10. Com a promoção para Beta, o CSI agora é ativado por padrão. Para obter mais informações sobre a implementação Beta do CSI no Kubernetes, consulte Container Storage Interface (CSI) para Kubernetes em Beta.

Configurando o CSI

A maioria dos plug-ins do CSI requer que a sinalização `--allow-privileged=true` seja configurada no binário do servidor API e binários do kubelet. Essa sinalização é ativada por padrão no IBM Cloud Private. Para um ambiente do IBM Cloud Private, é possível usar o Volume do CSI em um pod do Kubernetes sem precisar definir qualquer configuração extra.

Exemplo: implementando um driver NFS do CSI no IBM Cloud Private

Os arquivos de amostra usados nesta implementação de amostra estão disponíveis no repositório GitHub do Kubernetes. Veja kubernetes-csi/drivers.

1. Instale NFS. Para obter mais informações sobre como configurar o NFS, consulte a seção *Pré-requisitos* no tópico Criando um PersistentVolume do NFS.
2. Clone o repositório kubernetes-csi/drivers.
3. Alterne para o diretório `nfs`.

   ```
   Cd drivers / pkg/nfs
   ```

4. Crie a implementação.

   ```
   Kubectl create -f ./deploy/kubernetes/
   ```

   A saída se assemelha ao código a seguir:

   ```
   service "csi-attacher-nfsplugin" created
   statefulset.apps "csi-attacher-nfsplugin" created
   serviceaccount "csi-attacher" created
   clusterrole.rbac.authorization.k8s.io "external-attacher-runner" created
   clusterrolebinding.rbac.authorization.k8s.io "csi-attacher-role" created
   daemonset.apps "csi-nodeplugin-nfsplugin" created
   serviceaccount "csi-nodeplugin" created
   clusterrole.rbac.authorization.k8s.io "csi-nodeplugin" created
   clusterrolebinding.rbac.authorization.k8s.io "csi-nodeplugin" created
   ```

5. Implemente um pod usando o volume CSI criado.

   1. Localize o servidor NFS.

      ```
      Showmount -e
      ```

      A saída se assemelha ao código a seguir:

      ```
      Export list for nfsserver:
      /nfs *
      ```

2. Edite o `examples / kubernetes/nginx.yaml` arquivo. Atualize os parâmetros `server` e `share` par corresponder às suas configurações do sistema.

   ```
   cat examples/kubernetes/nginx.yaml | grep csi -A 5
   ```

   A saída se assemelha ao código a seguir:

   ```
     csi:
       driver: csi-nfsplugin
       volumeHandle: data-id
       volumeAttributes:
         server: 10.10.25.7
         share: /nfs
   ---
   ```

3. Crie um pod usando o driver NFS.

   ```
   Kubectl create -f exemplos / kubernetes/nginx.yaml
   ```

   A saída se assemelha ao código a seguir:

   ```
   persistentvolume "data-nfsplugin" created
   persistentvolumeclaim "data-nfsplugin" created
   pod "nginx" created
   ```

6. Verifique o ponto de montagem.

   1. Acesse o contêiner nginx.

      ```
      Kubectl exec -it sh nginx
      ```

   2. Mude para o diretório `/var/www`.

      ```
      Cd /var/www
      ```

   3. Crie um arquivo.

      ```
      Teste de toque
      ```

   4. Liste o arquivo no ponto de montagem.

      ```
      ls
      ```

      A saída se assemelha ao código a seguir:

      ```
      Teste
      ```

7. Verifique o servidor NFS.

   1. Localize o servidor.

      ```
      Nome do host
      ```

      A saída se assemelha ao código a seguir:

      ```
      nfsserver
      ```

   2. Liste os arquivos no servidor.

      ```
      /nfs ls
      ```

      A saída se assemelha ao código a seguir:

      ```
      Teste
      ```

Planejando uma solução de armazenamento

Planeje uma solução de armazenamento para os aplicativos em seu cluster. Escolha um provedor de armazenamento que atenda aos seus requisitos.

Tipos de armazenamento

Os tipos de armazenamento a seguir que são baseados no modo de volume estão disponíveis no IBM® Cloud Private.

- Armazenamento de bloco
- Armazenamento de arquivos
- Armazenamento de objetos

Provedores de Armazenamento

Os provedores de armazenamento a seguir estão disponíveis no IBM Cloud Private.

- GlusterFS
- Rook Ceph
- Minio
- vSphere
- hostpath
- Network file system (NFS)
- IBM Spectrum Scale ™

Retenção de armazenamento

As políticas de recuperação de armazenamento a seguir são suportadas no IBM Cloud Private.

- Retenção: recuperação manual
- Reciclar: limpeza básica (rm -rf /thevolume/*)
- Excluir: ativos de armazenamento associados, como um volume do AWS EBS, do GCE PD, do Azure Disk ou do OpenStack Cinder, são excluídos

Modo de acesso ao armazenamento

Nota: o modo de acesso que você seleciona tem um impacto sobre as decisões que estão relacionadas em múltiplas réplicas e no failover de nó.

Os modos de acesso a seguir podem ser configurados:

- ReadWriteOnce - o volume pode ser montado como leitura/gravação por um único nó
- ReadOnlyMany - o volume pode ser montado somente leitura por múltiplos nós
- ReadWriteMany - o volume pode ser montado como leitura/gravação por múltiplos nós

Tabela 1. Modos de acesso configuráveis por plug-in de volume

Plug-in de volume	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
GlusterFS	√	√	√
Ceph	√	√	√
IBM Spectrum Scale	√	√	√
hostPath	√	-	-
vSphere	√	-	- (funciona quando pods são localizados juntamente)
NFS	√	√	√

Fornecimento de armazenamento

Entenda como um volume persistente (PV) é criado, provisionado e acessado pela equipe de infraestrutura para executar backup e restauração. Além disso, entenda como um PV pode ser replicado para um site secundário.

Fornecimento estático

O administrador da infraestrutura primeiramente cria volumes de aplicativo (pontos de montagem) e, em seguida, assegura que eles estejam montados. Um administrador do Kubernetes expõe esses pontos de montagem do aplicativo por meio de PersistentVolumes do Kubernetes.

Fornecimento dinâmico

Um administrador do Kubernetes cria um conjunto de classes de armazenamento para pods a serem chamados. Cada classe de armazenamento tem um conjunto predefinido de características de armazenamento. São necessários PVs com vários parâmetros,

como desempenho, para atender aos diferentes requisitos de armazenamento. Esses parâmetros podem ser definidos usando recursos de classe de armazenamento. Para obter mais informações, consulte Configurar um Pod para usar um PersistentVolume para armazenamento.

Redimensionamento do volume persistente

No Kubernetes v1.11, o recurso de expansão de volume persistente é promovido para Beta e é possível redimensionar facilmente um volume existente editando o objeto PVC. Não é mais necessário interagir manualmente com o backend de armazenamento ou excluir e recriar objetos PV e PVC para aumentar o tamanho de um volume. Volumes persistentes reduzidos não são suportados.

Se o provedor em nuvem que você escolher não suportar esse recurso, um volume precisará ser redimensionado manualmente, se necessário.

Para permitir a expansão do PVC, ative o recurso `ExpandPersistentVolumes`. Para obter mais informações, consulte Portas de recurso.

Se você deseja evitar o redimensionamento de todas as solicitações, exceto quando ativá-lo em uma classe de armazenamento, ative `PersistentVolumeClaimResize`. Para obter mais informações, consulte Usando controladores de admissão.

Ao ativar o redimensionamento em uma classe de armazenamento, apenas os PVCs que usam a classe de armazenamento podem expandir volumes. Para obter mais informações, consulte Expandindo solicitações de volumes persistentes.

Outras considerações

Considere esses fatores adicionais enquanto você decide uma solução de armazenamento.

Desempenho

Entender suas necessidades de armazenamento do aplicativo e o requisito de desempenho pode ajudar a decidir qual provedor de armazenamento escolher e planejar para ativação de operação e de qualificação.

Backup e restauração

Entenda como volumes persistentes são criados, como um nome de volume é mapeado para o nome do volume do cluster interno, onde o volume está localizado e como é possível fazer backup e restaurar os volumes.

Segurança para dados em repouso e em movimento

Considere o segredo que é usado para a classe de armazenamento, a criptografia do disco e o sistema de arquivos.

Resiliência

Para obter informações sobre o armazenamento persistente de alta disponibilidade, consulte Planejando o armazenamento persistente altamente disponível. No IBM Cloud Private, o armazenamento corporativo é usado.

Comparação de nós de armazenamento hospedados no IBM Cloud Private versus nós

de armazenamento externo

Tabela 2. Comparação de gerenciamento de nós de armazenamento

Tópico	Nó de armazenamento do IBM Cloud Private	Nó de armazenamento externo
Suporte	Suporte IBM usando a assinatura de suporte TSS	Você gerencia o armazenamento
Atualizar e retroceder	Usando o gráfico Helm	Você gerencia a atualização e o retrocesso
Disponibilidade	Depende da disponibilidade do cluster	Você fornece o armazenamento

Planejando armazenamento persistente

Depois de revisar as soluções e os provedores de armazenamento disponíveis, planeje e solicite o armazenamento que é necessário a partir da equipe de infraestrutura.

Escolhendo uma solução de armazenamento

Antes de decidir qual tipo de armazenamento é a solução correta para você, é preciso entender seus requisitos de aplicativo, o tipo de dados que você deseja armazenar e com que frequência deseja acessar esses dados.

1. Decida se seus dados devem ser armazenados permanentemente ou se seus dados podem ser removidos em qualquer ponto no tempo.

    - **Armazenamento persistente**: seus dados ainda devem estar disponíveis, mesmo que o contêiner, o nó do trabalhador ou o cluster seja removido. Use armazenamento persistente nos cenários a seguir:

        - Apps stateful
        - Dados de negócios principais
        - Dados que devem estar disponíveis devido a requisitos legais, como um período de retenção definido
        - Auditoria
        - Dados que devem ser acessados e compartilhados entre as instâncias do aplicativo

    - **Armazenamento não persistente**: seus dados podem ser removidos quando o contêiner, o nó do trabalhador ou o cluster for removido. O armazenamento não persistente é geralmente usado para informações de criação de log, como logs do sistema ou logs do contêiner, teste de desenvolvimento ou quando você deseja acessar dados do sistema de arquivos do host.

2. Se tiver que persistir seus dados, analise se seu app requer um tipo de armazenamento específico. Ao usar um aplicativo existente, o aplicativo pode ser projetado para armazenar dados de uma das maneiras a seguir:

    - **Em um sistema de arquivos**: os dados podem ser armazenados como um arquivo em um diretório. Por exemplo, é possível armazenar esse arquivo em seu disco rígido local. Alguns apps requerem que os dados sejam armazenados em um sistema de arquivos específico, como `nfs` ou `ext4`, para otimizar o armazenamento de dados e atingir os objetivos de desempenho.
    - **Em um banco de dados**: os dados devem ser armazenados em um banco de dados que segue um esquema específico. Alguns apps vêm com uma interface de banco de dados que pode ser usada para armazenar seus dados. Por exemplo, o WordPress é otimizado para armazenar dados em um banco de dados MySQL. Nesses casos, o tipo de armazenamento é selecionado para você.

3. Se o seu app não tiver uma limitação no tipo de armazenamento que deve ser usado, determine o tipo de dados que você deseja armazenar.

    - **Dados estruturados**: dados que podem ser armazenados em um banco de dados relacional no qual você tem uma tabela com colunas e linhas. Os dados em tabelas podem ser conectados usando chaves e geralmente são fáceis de acessar devido ao modelo de dados predefinido. Exemplos são números de telefone, números de conta, números de Seguridade Social ou CEPs.
    - **Dados semi-estruturados**: dados que não se ajustam em um banco de dados relacional, mas que vem com algumas propriedades organizacionais que podem ser usadas para ler e analisar esses dados mais facilmente. Exemplos são arquivos de linguagem de marcações, como CSV, XML ou JSON.
    - **Dados não estruturados**: dados que não seguem um padrão organizacional e que são tão complexos que não podem ser armazenados em um banco de dados relacional com modelos de dados predefinidos. Para acessar esses dados, ferramentas e software avançados são necessários. Exemplos são e-mails, vídeos, fotos, arquivos de áudio, apresentações, dados de mídia social ou páginas da web.

    Se seus dados forem tanto estruturados quanto não estruturados, tente armazenar cada tipo de dados separadamente em uma solução de armazenamento que tenha sido projetada para esse tipo de dados. O uso de uma solução de armazenamento apropriada para seu tipo de dados facilita o acesso aos seus dados e fornece os benefícios de desempenho, escalabilidade, durabilidade e consistência.

4. Analise como você deseja acessar seus dados. As soluções de armazenamento são geralmente projetadas e otimizadas para suportar operações de leitura ou gravação.

    - **Somente leitura**: seus dados são somente leitura. Você não deseja gravar ou mudar seus dados.
    - **Leitura e gravação**: você deseja ler, gravar e mudar seus dados. Para dados que são lidos e gravados, é importante entender se as operações são de leitura pesada, de gravação pesada ou balanceada.

5. Determine a frequência na qual seus dados são acessados. O entendimento da frequência do acesso a dados pode ajudá-lo a entender o desempenho que você requer para seu armazenamento. Por exemplo, os dados que são acessados com frequência geralmente residem no armazenamento rápido.

    - **Dados quentes**: dados que são acessados com frequência. Casos de uso comuns são apps da web ou móveis.
    - **Dados frios ou quentes**: dados que são acessados poucas vezes, como uma vez por mês ou menos. Os casos de uso comuns são archives, retenção de dados de curto prazo ou recuperação de desastre.

- **Dados frios**: dados que são raramente acessados ou nem são acessados. Casos de uso comuns são archives, backups de longo prazo, dados históricos.
- **Dados congelados**: dados que não são acessados e que precisam ser mantidos devido a razões legais.

Se você não conseguir prever a frequência ou se a frequência não seguir um padrão estrito, determine se suas cargas de trabalho são de leitura ou gravação pesada ou balanceada. Em seguida, consulte a opção de armazenamento que se ajuste à sua carga de trabalho e investigue qual camada de armazenamento fornece a flexibilidade de que você precisa.

6. Investigue se seus dados devem ser compartilhados entre múltiplas instâncias do app.
   Ao usar volumes persistentes do Kubernetes para acessar seu armazenamento, é possível determinar o número de pods que podem montar o volume ao mesmo tempo. Algumas soluções de armazenamento, como armazenamento de bloco, podem ser acessadas somente por um pod por vez. Com outras soluções de armazenamento, é possível compartilhar o volume entre múltiplos pods.

7. Entenda outras características de armazenamento que impactam sua opção.

   - **Consistência**: a garantia de que uma operação de leitura retorna a versão mais recente de um arquivo. As soluções de armazenamento podem fornecer consistência forte quando você tem a garantia de que sempre receberá a versão mais recente de um arquivo ou uma consistência eventual quando a operação de leitura pode não retornar a versão mais recente. Frequentemente os sistemas distribuídos geograficamente fornecem uma consistência eventual pelo fato de que uma operação de gravação deve ser replicada primeiramente para todas as instâncias.
   - **Desempenho**: o tempo que leva para concluir uma operação de leitura ou de gravação.
   - **Durabilidade**: a garantia de que uma operação de gravação que está confirmada em seu armazenamento continue permanentemente e que não seja corrompida ou perdida, mesmo se gigabytes ou terabytes de dados forem gravados em seu armazenamento ao mesmo tempo.
   - **Resiliência**: a capacidade de se recuperar de uma indisponibilidade e continuar as operações, mesmo se um componente de hardware ou software falhou. Por exemplo, seu armazenamento físico experimenta uma indisponibilidade de energia, uma indisponibilidade de rede ou é destruído durante um desastre natural.
   - **Disponibilidade**: a capacidade de fornecer acesso aos seus dados, mesmo se um data center ou uma região estiver indisponível. A disponibilidade para seus dados é geralmente obtida pela inclusão de redundância e configuração de mecanismos de failover.
   - **Escalabilidade**: a possibilidade de estender a capacidade e customizar o desempenho com base em suas necessidades.
   - **Criptografia**: o mascaramento de dados para evitar visibilidade quando os dados são acessados por um usuário não autorizado.

8. Revise as soluções de armazenamento persistente disponíveis e escolha a solução que melhor se ajusta ao seu app e aos requisitos de dados. Para obter as soluções disponíveis, consulte Guia de armazenamento.

Opções de armazenamento no IBM Cloud Private

Opções de armazenamento que estão disponíveis no IBM® Cloud Private.

Visão geral

O IBM® Cloud Private fornece várias tecnologias de armazenamento para armazenamento da carga de trabalho do aplicativo. O IBM Cloud Private suporta todos os tipos de armazenamento que são suportados por Kubernetes. No entanto, os tipos de armazenamento a seguir são testados e verificados no IBM Cloud Private. Todas as tecnologias de armazenamento de software livre são fornecidas no IBM Cloud Private e apoiadas pela instrução de suporte de software livre do IBM Cloud Private.

Tipos de armazenamento fornecidos

Os tipos de armazenamento a seguir são testados e verificados no IBM Cloud Private.

hostPath

Um volume `hostPath` monta um arquivo ou diretório a partir do sistema de arquivos do nó do host em seu pod.

O volume `hostPath` é geralmente usado para teste de nó único. Se o ajuste de escala de pod acontece horizontalmente nos nós, as mesmas informações compartilhadas não estão disponíveis para o pod. O `hostPath` não suporta fornecimento dinâmico.

Network File System

Um volume do Network File System (NFS) permite que um compartilhamento NFS existente seja montado em seu pod.

Os volumes NFS são persistentes e os dados podem ser distribuídos entre os pods. O NFS pode ser montado simultaneamente por múltiplos gravadores. O armazenamento do NFS é fácil de configurar e é familiar para a maioria dos administradores. No entanto, o desempenho pode ser uma preocupação e nenhum fornecimento dinâmico é suportado.

vSphere

Um volume do vSphere é usado para montar um volume do vSphere Virtual Machine Disk (VMDK) no pod.

Quando o IBM Cloud Private é instalado em uma infraestrutura do VMware vSphere e um vSphere Cloud Provider é configurado, os volumes do vSphere podem ser criados para o pod de carga de trabalho.

Os volumes do vSphere são persistentes, embora ele suporte apenas o modo de acesso ReadWriteOnce. Portanto, ele pode não ser uma boa opção quando um aplicativo está procurando por múltiplas réplicas do pod e para manipular o failover do nó. No entanto, os pods que são colocados no mesmo nó podem usar o mesmo volume.

GlusterFS

GlusterFS é um sistema de arquivos Scale Out Network Attached Storage.

O GlusterFS é um sistema de arquivos distribuído e escalável que fornece volume persistente. O fornecimento de volume é feito dinamicamente por meio do Heketi. O GlusterFS fornecido pelo IBM Cloud Private é uma solução de software livre. O IBM Cloud Private conta com a comunidade de software livre para fornecer correções de erros e melhorias, se necessário.

Rook Ceph

O Rook Ceph é o sistema de armazenamento distribuído do Ceph que é implementado e gerenciado pelo Rook Operator dentro do IBM Cloud Private.

O Ceph é um sistema de armazenamento distribuído com várias apresentações de armazenamento que incluem armazenamento de objetos, armazenamento de bloco e sistema de arquivos compartilhados compatível com POSIX. O Ceph no IBM Cloud Private é implementado e gerenciado por meio do Rook. Como o Rook e o Ceph são projetos de software livre, o IBM Cloud Private conta com a comunidade de software livre para trazer melhorias e fornecer correções de erros, se necessário.

Minio

O Minio é um servidor de armazenamento de objeto leve e distribuído compatível com o Amazon S3.

O Minio no IBM Cloud Private é o armazenamento de objeto de software livre que é fácil de configurar e fornece o armazenamento de bloco suportado do IBM Cloud Private existente. O IBM Cloud Private depende da comunidade de software livre para trazer novas melhorias e fornecer correções de erros, se necessário.

IBM Spectrum Scale

O IBM Spectrum Scale™ é um sistema de arquivos de cluster de classe corporativa que fornece acesso simultâneo a um sistema de arquivos único ou a um conjunto de sistemas de arquivos de múltiplos nós.

Atualmente, o volume do IBM Spectrum Scale pode ser consumido por um pod por meio do hostPath ou montando-o como um volume do NFS. O volume do IBM Spectrum Scale também pode ser consumido criando classes de armazenamento usando o plug-in do IBM Storage Enabler for Containers. O plug-in do IBM Storage Enabler for Containers permite que os sistemas de armazenamento de bloco da IBM e que o IBM Spectrum Scale sejam utilizados como dispositivos de armazenamento para clusters de contêiner do Kubernetes. Para obter mais informações, consulte Instalador para o IBM Storage Enabler for Containers.

Comparação de provedores de armazenamento

| Tecnologias de armazenamento | Hospedado no IBM Cloud Private | Fornecimento de armazenamento | Modo de acesso | Comentários | | ---------- | ---------- | ---------- | ---------- | ----- | | GlusterFS | SIM | Dinâmico | somente leitura/gravação (RWO), leitura/gravação de muitos (RWX), memória de leitura (ROM) | Baseado no gráfico de Helm. Pode ser implementado junto com ou depois da instalação do IBM Cloud Private. | Rook Ceph | SIM | Dinâmico | RWO, ROM| Baseado no gráfico de Helm. Pode ser implementado após a instalação do IBM Cloud Private a partir do catálogo da console de gerenciamento. | | vSphere | Não | Dinâmico | RWO, RWX (Pods colocados)| | | Minio | SIM | - | - | Baseado no gráfico de Helm. Pode ser implementado junto com ou depois da instalação do IBM Cloud Private. | NFS | Não | Estático | RWO, RWX, ROM | | | hostPath | Não |Estático | RWO | | | IBM Spectrum Scale| Não | Estático, Dinâmico | RWO, RWX, ROM| |

Opções de armazenamento hospedadas no IBM Cloud Private

As opções de armazenamento que podem ser configuradas em seus nós de cluster do IBM® Cloud Private.

O armazenamento a seguir pode ser hospedado em seus nós do cluster do IBM Cloud Private:

- GlusterFS
- Armazenamento de bloco do Ceph usando Rook
- Minio

Visão geral

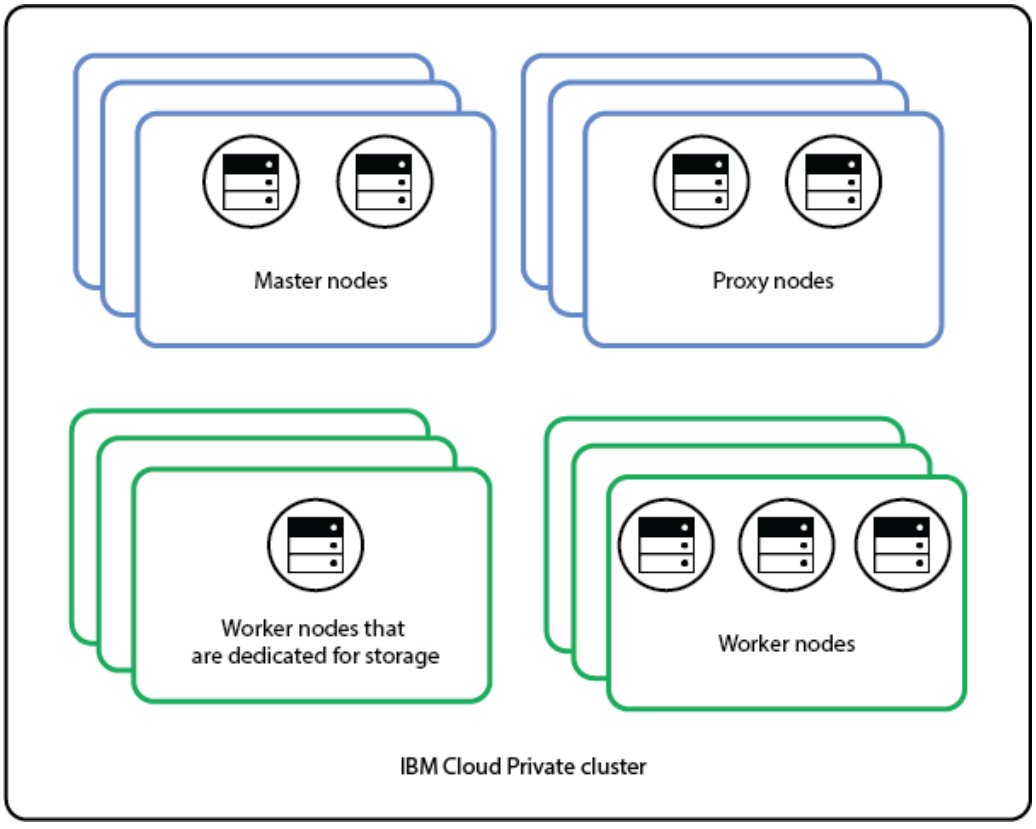
O armazenamento que pode ser configurado em seus nós de cluster do IBM Cloud Private é uma versão em conteinerizada da tecnologia de armazenamento e é executado como um pod em seu nó do cluster. Para todo o armazenamento que está configurado em seus nós do cluster, o gerenciamento, o monitoramento e o upgrade são gerenciados dentro de seu cluster seguindo os recursos do Kubernetes.

Como administrador, é possível escolher hospedar o pod nos nós do trabalhador dedicados. Também é possível, se necessário, usar esses nós para executar as cargas de trabalho do aplicativo. No entanto, o uso de nós do trabalhador dedicados somente para armazenamento é uma opção melhor.

Nó de armazenamento dedicado

No IBM Cloud Private, nós de armazenamento dedicados são nós do trabalhador que são dedicados para armazenamento. Rotule e contamine nós do trabalhador regulares para dedicá-los como nós de armazenamento.

O diagrama a seguir mostra um cluster do IBM Cloud Private típico em que três nós são dedicados para armazenamento.



Como criar nós de armazenamento dedicados

É possível criar grupos de hosts customizados no IBM Cloud Private. Esse grupo de hosts customizados rotula automaticamente os nós e os contamina para tornar os nós como dedicados. Para obter mais informações, consulte Definindo grupos de hosts customizados.

A seguir está uma configuração de exemplo de como criar nós dedicados:

Inclua o novo grupo de hosts no arquivo `/<installation_directory>/cluster/hosts`.

```
[worker] 2.2.2.2 ...
2.2.2.9 .
.
[hostgroup-somestorage]
6.6.6.6
...
6.6.6.8
```

Após a instalação do IBM Cloud Private, seu cluster possui nós do trabalhador [2.2.2.2-2.2.2.9], que estão disponíveis para carga de trabalho do aplicativo regular. Os nós [6.6.6.6-6.6.6.8] são nós do trabalhador dedicados nos quais é possível instalar um aplicativo de armazenamento.

- GlusterFS
- Armazenamento de bloco do Ceph usando Rook
- Minio

GlusterFS

Configure o armazenamento do GlusterFS nos nós em seu cluster do IBM® Cloud Private. Os pods em seu aplicativo podem, então, usar esse cluster de armazenamento GlusterFS para persistência de dados.

GlusterFS é um sistema de arquivos Scale Out Network Attached Storage. O GlusterFS possui um modelo de cliente/servidor. Os servidores possuem tijolos de armazenamento e executam o Daemon `glusterfsd`. Os clientes GlusterFS se conectam aos servidores por meio de um protocolo de camada de transporte.

É possível usar o armazenamento do GlusterFS no IBM Cloud Private implementando o GlusterFS em seus nós do cluster do IBM Cloud Private ou integrando um cluster de armazenamento do GlusterFS que é implementado fora do ambiente do IBM Cloud Private.

As seções a seguir fornecem informações sobre como implementar um cluster do GlusterFS em seus nós de cluster do IBM Cloud Private.

Requisito do sistema

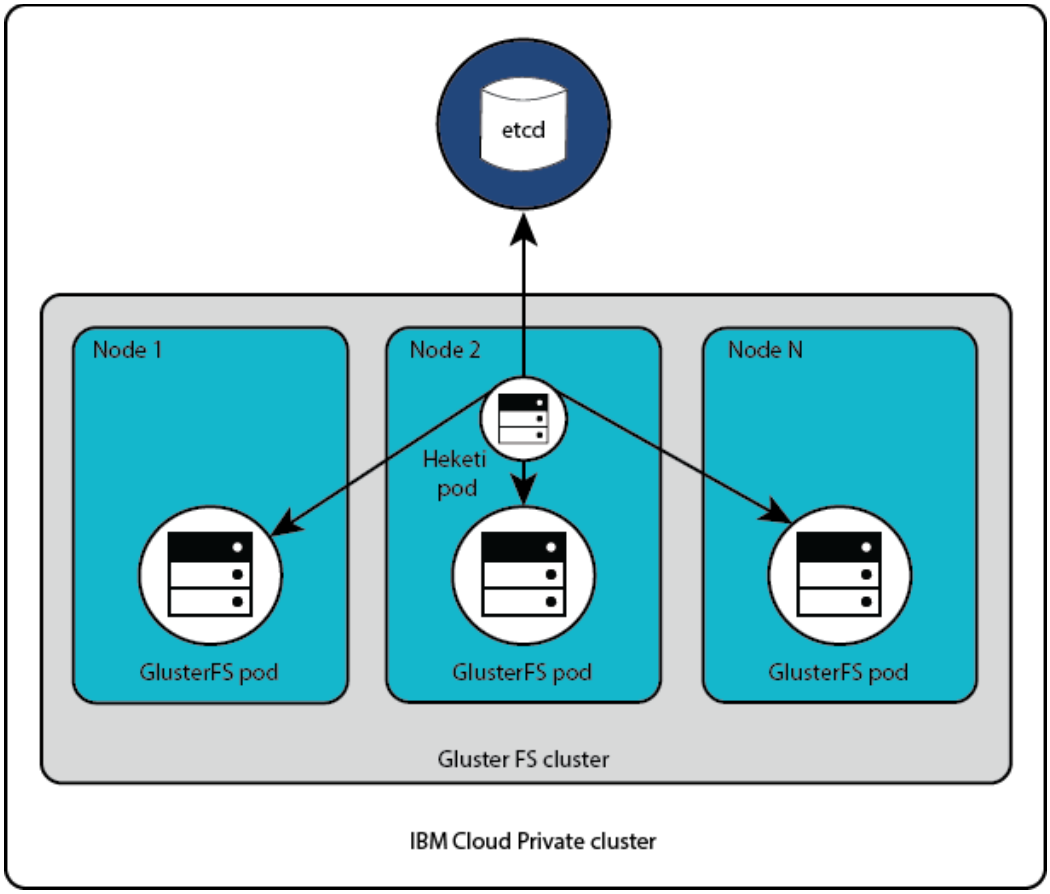
Para obter mais informações, consulte Requisitos do sistema.

Cenários de Implementação

É possível usar nós dedicados para instalar o GlusterFS ou é possível usar nós do trabalhador que são compartilhados com as cargas de trabalho do aplicativo para instalar o GlusterFS.

É necessário um mínimo de um nó para implementar o GlusterFS. Em um cluster GlusterFS, todos os nós possuem pods GlusterFS que são executados como Daemonsets. O pod Heketi é executado como uma implementação da réplica 1.

O diagrama a seguir mostra uma instalação típica do GlusterFS no IBM Cloud Private.



Para obter mais informações, consulte Cenários de implementação.

Pré-requisitos e preparação do nó

Para obter mais informações, consulte Preparando os nós.

Implementando o GlusterFS

É possível implementar o GlusterFS juntamente com a instalação do IBM Cloud Private. Se for necessário incluir o GlusterFS após a instalação do IBM Cloud Private, será possível implementá-lo como um serviço de complemento ou implementar o gráfico Helm a partir do catálogo do IBM Cloud Private.

Opções de configuração

Para obter mais informações, consulte Configurando o GlusterFS.

Criando classe de armazenamento

Ao implementar o GlusterFS, por padrão, uma classe de armazenamento é incluída para aplicativos para provisionar o volume dinâmico.

É possível criar mais classes de armazenamento, se necessário. Para obter mais informações sobre como criar uma classe de armazenamento para o GlusterFS, consulte Criando uma classe de armazenamento para o GlusterFS.

Verificando a Configuração

É possível validar a configuração do GlusterFS implementando um aplicativo que requer armazenamento persistente. Implemente o aplicativo a partir do catálogo do IBM Cloud Private. Por exemplo, consulte PostgreSQL.

Gerenciando seu cluster

É possível gerenciar sua configuração do GlusterFS depois que seu cluster do IBM Cloud Private estiver instalado.

Para obter mais informações, consulte Gerenciando seu cluster GlusterFS.

Expandindo o tamanho de uma solicitação de volume persistente existente

Para obter mais informações, consulte Aumentando a capacidade de seu volume do GlusterFS existente.

Aumentando a capacidade de armazenamento

É possível aumentar a capacidade de armazenamento de um cluster do GlusterFS existente incluindo um nó de armazenamento ou incluindo discos nos nós de armazenamento existentes.

- Incluindo um dispositivo em um nó de armazenamento do GlusterFS
- Incluindo um nó de armazenamento no cluster do GlusterFS

Atualizando o GlusterFS

Para obter mais informações, consulte Fazendo upgrade do GlusterFS.

Desinstalando o GlusterFS

Para obter mais informações, consulte Desinstalar o GlusterFS.

Aplicativo

O monitoramento interno do IBM Cloud Private usa o Prometheus e o Grafana. É possível usar o painel Grafana para monitoramento, alerta e acionadores para monitorar o armazenamento que está configurado em seus nós do cluster.

Criação de Log

A criação de log interna do IBM Cloud Private usa o ELK. É possível integrar o ELK a um sistema externo usando um arquivo do Cloud Auditing Data Federation (CADF).

Resolução de problemas

Para obter mais informações, consulte Resolução de problemas do GlusterFS.

- Requisitos do sistema
- Cenários de implementação
- Preparando os nós
- Preparando os discos
- Configurando o GlusterFS
- Verificando a configuração do GlusterFS
- Criando uma classe de armazenamento para GlusterFS
- Solicitando um volume do GlusterFS
- Gerenciando seu cluster do GlusterFS
- Monitorando o GlusterFS
- Fazendo upgrade do GlusterFS
- Desinstalando o GlusterFS
- Reinstalando o GlusterFS ou o IBM Cloud Private
- Resolução de problemas do GlusterFS

Requisitos do sistema

Requisitos do sistema para configurar GlusterFS.

Requisitos de hardware

- Deve-se usar pelo menos um nó dedicado para o GlusterFS.
- O dispositivo de armazenamento que é usado para GlusterFS deve ter uma capacidade de pelo menos 25 GB.

- O dispositivo de armazenamento que você usa para GlusterFS deve ser um disco bruto. Ele não deve ser formatado, particionado ou usado para necessidades de armazenamento do sistema de arquivos.

Os requisitos mínimos de CPU, Memória, RAM e espaço em disco para os nós do GlusterFS dedicados são conforme mostrado na Tabela 1.

Tabela 1. Requisitos mínimos de hardware para um nó do GlusterFS

Requisito	Nó GlusterFS	Notes
Número de hosts	1 ou mais	
Núcleos	2 ou mais	Escolha um processador moderno com múltiplos núcleos. Os servidores geralmente são fornecidos com dois soquetes (dois processadores), portanto, o número total de núcleos no sistema é o número de soquetes multiplicado pelo número de núcleos em cada processador. Hoje, os processadores geralmente têm pelo menos oito núcleos por processador. Os processadores avançados podem ter 20 ou mais núcleos por processador. Quando se trata de escolher um processador, pode ser uma boa escolha ter uma abordagem moderada. Ou seja, selecione um processador que tenha um número médio de núcleos, frequência e tamanho do cache (não o mais alto ou o mais baixo).
CPU	>= 2,4 GHz	• Para um cluster do Linux®, escolha uma CPU que suporte SSE 4.2. • Para um cluster do Linux® on Power® (ppc64le), use uma CPU que seja versão Power8 ou superior.
RAM	>= 8 GB	Cada volume do GlusterFS precisa de aproximadamente 30 MB de RAM em cada nó do GlusterFS. A quantidade total de RAM necessária depende do número de volumes que podem ser necessários para o cluster.
Espaço em disco para instalar	>= 150 GB	

Requisitos de Software

- IBM® Cloud Private Versão 3.2.0
- O IBM Cloud Private suporta o IBM GlusterFS Chart v1.4.0, o GlusterFS versão 4.1.5.1 e o Heketi versão 8.0.0.1.
- O cliente GlusterFS deve ser instalado em todos os nós que usam o volume GlusterFS.
- Os módulos kernel `dm_thin_pool` devem ser carregados em todos os nós de armazenamento nos quais os servidores GlusterFS estão instalados.

  Para obter mais informações sobre como instalar o cliente GlusterFS e carregar os módulos de kernel do `dm_thin_pool`, consulte Preparando os nós.

Portas Obrigatórias

Certifique-se de que essas portas estejam abertas, mas não em uso por qualquer serviço:

Tabela 1. Portas do GlusterFS

Porta	Utilização
24007	Daemon do GlusterFS
24008	Gerenciamento de GlusterFS
2222	sshd (usado quando GlusterFS é executado em um pod)
49152:49251	Porta TCP para cada tijolo em um volume

É possível usar o comando `netstat -an | grep <port number> | grep -i listen` para verificar se as portas estão abertas. Se o comando retorna uma saída, isso significa que a porta está aberta, mas está em uso. Pare o serviço que está usando a porta.

Cenários de Implementação

Opções para configurar o GlusterFS no IBM® Cloud Private.

É possível usar o armazenamento GlusterFS no IBM Cloud Private configurando o GlusterFS nos nós do cluster do IBM® Cloud Private ou utilizando um servidor GlusterFS que está configurado fora do ambiente do IBM Cloud Private.

Utilizar os nós do cluster do IBM Cloud Private para configurar um cluster GlusterFS

É possível usar nós dedicados para a configuração do GlusterFS ou utilizar nós do trabalhador para configurar o GlusterFS.

Nós de armazenamento GlusterFS Dedicados

Defina um grupo de hosts customizado com pelo menos um nó. Para obter mais informações sobre como definir um grupo de hosts, consulte Definindo as funções de nó no arquivo de hosts. Esse grupo de hosts customizados rotula automaticamente os nós e os contamina para tornar os nós dedicados para o GlusterFS Storage. Se você ativou o firewall em seu cluster, será necessário abrir todas as portas que são usadas pelo daemon e tijolos do GlusterFS. Para obter mais informações, consulte Portas necessárias .

Nós do Worker como nós de armazenamento GlusterFS

A seguir estão as opções para instalar o GlusterFS nos nós do trabalhador.

- É possível usar os nós do trabalhador do IBM Cloud Private existentes para instalar o GlusterFS. Você deve rotular manualmente esses nós. Por exemplo, a seguir está o comando para incluir `storagenode=glusterfs` como o rótulo:

  ```
  kubectl label nodes <node_1_IP_address> <node_2_IP_address> <node_3_IP_address>
  storagenode=glusterfs --overwrite=true
  ```

(OR)

- É possível usar nós dedicados para configurar o GlusterFS e incluir esses nós como nós do trabalhador durante a instalação do IBM Cloud Private. Os nós devem ser incluídos como nós do trabalhador e também como um grupo de hosts customizado no arquivo `/<installation_directory>/cluster/hosts`. Para obter mais informações, consulte Configurando as funções de nó no arquivo de hosts. Além disso, para ignorar a marcação do nó, deve-se configurar o parâmetro `no_taint_group` no arquivo `/<installation_directory>/cluster/config.yaml`. Para obter mais informações, consulte Configurando o GlusterFS durante a instalação do IBM Cloud Private.

Utilizar um servidor GlusterFS que esteja configurado fora do ambiente do IBM Cloud Private

É possível usar um servidor GlusterFS que está configurado fora do ambiente do IBM Cloud Private. Para obter mais informações, consulte Documentação do GlusterFS.

Para utilizar esse servidor GlusterFS, deve-se criar uma classe de armazenamento. Para obter mais informações, consulte Criando uma classe de armazenamento para GlusterFS. Assegure-se de que o Heketi esteja instalado para gerenciar o servidor GlusterFS que está configurado fora do ambiente do IBM Cloud Private.

Para verificar se o servidor Heketi está acessível por meio do nó principal do IBM Cloud Private, execute o comando a seguir no nó principal:

```
curl http://<Heketi_server_IP>:<Heketi_server_port>/hello
```

Se a saída de comando incluir `Hello from Heketi`, o servidor Heketi estará acessível.

Preparando os nós

Prepare os IBM® Cloud Private nós para a configuração GlusterFS.

1. Em cada nó de armazenamento GlusterFS, conclua estas etapas:

   a. Configure o módulo de kernel `dm_thin_pool`.

   ```
   sudo modprobe dm_thin_pool
   ```

   b. Para assegurar que o carregamento do módulo kernel seja persistido nas reinicializações, inclua o nome do módulo `dm_thin_pool` no arquivo `modules`.

   - No Ubuntu, execute este comando:

     ```
     echo dm_thin_pool | sudo tee -a /etc/modules
     ```

   - No Red Hat Enterprise Linux (RHEL) e SUSE Linux Enterprise Server (SLES), execute este comando:

     ```
     echo dm_thin_pool | sudo tee -a /etc/modules-load.d/dm_thin_pool.conf
     ```

     Se as mudanças forem salvas corretamente, os comandos retornarão a saída a seguir:

     ```
     dm_thin_pool
     ```

2. Em cada nó do cluster do IBM Cloud Private que usa o volume GlusterFS, instale o cliente do GlusterFS.

   - No Ubuntu, execute os comandos a seguir:

     ```
     sudo apt-get update

     sudo apt-get install glusterfs-client
     ```

   - No RHEL, execute o comando a seguir:

     ```
     sudo yum install glusterfs-client
     ```

   - No SLES, execute os seguintes comandos:

     ```
     zypper addrepo
     https://download.opensuse.org/repositories/filesystems/SLE_12_SP3/filesystems.repo

     atualização do zypper

     instalação do zypper glusterfs
     ```

   - No RHEL Linux® on Power® (ppc64le), primeiro crie um repositório, em seguida, instale o cliente GlusterFS:

     1. Crie um repositório:

        ```
        [centos-gluster41]
        name=CentOS-$releasever - Gluster 4.1
        baseurl=https://buildlogs.centos.org/centos/7/storage/$basearch/gluster-4.1/
        gpgcheck=0
        enabled=1
        ```

     2. Instale o cliente do GlusterFS:

        ```
        sudo yum install glusterfs-client
        ```

   - No SLES Linux® on Power® (ppc64le), primeiro inclua um repositório e, em seguida, instale o pacote GlusterFS:

     1. Importe a chave:

        ```
        rpm --import https://oplab9.parqtec.unicamp.br/pub/ppc64el/glusterfs/sles/openpower-
        gpgkey-public.asc
        ```

        Verifique se a chave foi importada de forma correta:

```
rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE}\t%{SUMMARY}\n'
```

2. Inclua o repositório.

```
zypper ar -t YUM https://oplab9.parqtec.unicamp.br/pub/ppc64el/glusterfs/sles/
OpenPower\ Unicamp\ Lab
```

3. Instale o pacote GlusterFS:

```
instalação do zypper glusterfs
```

- No SLES no IBM Z, primeiro inclua um repositório e, em seguida, instale o pacote GlusterFS:

1. Importe a chave:

```
rpm --import https://oplab9.parqtec.unicamp.br/pub/key/openpower-gpgkey-public.asc
```

Verifique se a chave foi importada de forma correta:

```
rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE}\t%{SUMMARY}\n'
```

2. Inclua o repositório.

```
zypper ar -t YUM https://oplab9.parqtec.unicamp.br/pub/s390x/glusterfs/sles/12sp4/
OpenPower\ Unicamp\ Lab
```

3. Instale o pacote GlusterFS:

```
instalação do zypper glusterfs
```

- No RHEL no IBM Z, primeiro inclua um repositório e, em seguida, instale o pacote GlusterFS:

1. Importe a chave:

```
rpm --import https://oplab9.parqtec.unicamp.br/pub/key/openpower-gpgkey-public.asc
```

Verifique se a chave foi importada de forma correta:

```
rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE}\t%{SUMMARY}\n'
```

2. Inclua o repositório.

```
yum install yum-utils && yum-config-manager --add-repo
https://oplab9.parqtec.unicamp.br/pub/s390x/glusterfs/rhel/7.6/
```

3. Instale o pacote GlusterFS:

```
yum update && yum install glusterfs-client gluster-cli
```

- No Ubuntu 18.04 no IBM Z, primeiro inclua o repositório e, em seguida, instale o pacote GlusterFS:

1. Inclua o repositório.

```
add-apt-repository "deb
https://oplab9.parqtec.unicamp.br/pub/s390x/glusterfs/ubuntu/ bionic main"
```

2. Instale o pacote GlusterFS:

```
apt-get update
apt-get install glusterfs-client=4.1.5-ubuntu1~bionic1
```

Verifique se o pacote GlusterFS está instalado corretamente:

```
/usr/sbin/glusterfs --version
```

- No Ubuntu 16.04 no IBM Z, primeiro inclua o repositório, em seguida, instale o pacote GlusterFS:

1. Inclua o repositório.

```
add-apt-repository "deb
https://oplab9.parqtec.unicamp.br/pub/s390x/glusterfs/ubuntu/ xenial main"
```

2. Instale o pacote GlusterFS:

```
apt-get update
apt-get install glusterfs-client=4.1.5-ubuntu1~xenial2
```

Verifique se o pacote GlusterFS está instalado corretamente:

```
/usr/sbin/glusterfs --version
```

Preparando os discos

Prepare seus discos para a configuração do GlusterFS.

Deve-se usar symlinks dos discos ao configurar o GlusterFS. Há a opção de configurar os discos como volumes de caminhos múltiplos e para criptografar os volumes.

Volumes múltiplos

É possível usar volumes de caminhos múltiplos para o GlusterFS.

Para obter mais informações sobre como configurar caminhos múltiplos, consulte a documentação a seguir:

- Para Ubuntu, consulte https://help.ubuntu.com/lts/serverguide/dm-multipath-chapter.html
- Para o Red Hat Enterprise Linux (RHEL), consulte https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/dm_multipath/index
- Para o SUSE Linux Enterprise Server (SLES), consulte https://www.suse.com/documentation/sles-12/stor_admin/data/cha_multipath.html.

Volumes criptografados

É possível usar volumes criptografados para o GlusterFS. Para obter mais informações sobre como criptografar volumes, consulte Criptografando volumes usando dm-crypt.

Os volumes criptografados estão disponíveis no local `/dev/mapper/<encrypted-volume-name>`.

Symlinks

Deve-se usar o link simbólico (symlink) para identificar o dispositivo de armazenamento do GlusterFS. Não use nomes de dispositivos, como `/dev/sdb`, porque o nome pode mudar entre as reinicializações do sistema.

- Utilizar symlinks gerados pelo sistema
- Usar symlinks criados manualmente

Nota: os caracteres especiais que o Heketi permite que sejam utilizados no nome do dispositivo são `^/[a-zA-Z0-9_.:/-]+$`. Caso o nome do dispositivo ou o symlink gerado pelo sistema tenham caracteres especiais que não são permitidos pelo Heketi, o symlink deve ser criado manualmente.

Utilizar symlinks gerados pelo sistema

Para obter o symlink que o sistema designa a um dispositivo, conclua estas etapas:

1. Identifique os dispositivos de armazenamento a serem usados. É possível listar os dispositivos de armazenamento disponíveis inserindo este comando:

   ```
   fdisk -l
   ```

   A saída é semelhante a este exemplo:

   ```
   root@icps-worker-4:~# fdisk -l
      Disk /dev/sdb: 48 GiB, 51539607552 bytes, 100663296 sectors
      Units: sectors of 1 * 512 = 512 bytes
      Sector size (logical/physical): 512 bytes / 512 bytes
      I/O size (minimum/optimal): 512 bytes / 512 bytes

   Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
      Units: sectors of 1 * 512 = 512 bytes
      Sector size (logical/physical): 512 bytes / 512 bytes
      I/O size (minimum/optimal): 512 bytes / 512 bytes
   ```

```
   Disklabel type: dos
   Disk identifier: 0xe96e93a3

Device Boot Start End Sectors Size Id Type
   /dev/sda1 * 2048 7813119 7811072 3.7G 82 Linux swap / Solaris
   /dev/sda2 7815166 83884031 76068866 36.3G 5 Extended
   /dev/sda5 7815168 83884031 76068864 36.3G 83 Linux
```

Nota: se você usar dispositivos de caminhos múltiplos ou criptografados, esses dispositivos estão disponíveis no local `/dev/mapper/`.

2. Identifique os dispositivos que têm pelo menos 25 GB de capacidade de armazenamento. No exemplo, o dispositivo `/dev/sdb` é considerado.

3. Apague todas as assinaturas do sistema de arquivos, do raid e da tabela de partição usando o comando `wipefs`. Por exemplo, para apagar as assinaturas no dispositivo `/dev/sdb`, execute o comando a seguir:

```
sudo wipefs --all --force /dev/sdb
```

4. Obtenha o symlink do dispositivo inserindo este comando:

```
ls -altr /dev/disk/*
```

A saída é semelhante a este exemplo:

```
root@icps-worker-4:~# ls -altr /dev/disk/*
   /dev/disk/by-path:
   total 0
   drwxr-xr-x 2 root root 160 Oct 23 02:34 .
lrwxrwxrwx 1 root root 9 Oct 23 02:34 pci-0000:00:10.0-scsi-0:0:1:0 -> ../../sdb
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 pci-0000:00:10.0-scsi-0:0:0:0 -> ../../sda
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 pci-0000:02:01.0-ata-1 -> ../../sr0
   drwxr-xr-x 6 root root 120 Oct 23 02:34 ..
lrwxrwxrwx 1 root root 10 Oct 23 02:34 pci-0000:00:10.0-scsi-0:0:0:0-part2 -> ../../sda2
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 pci-0000:00:10.0-scsi-0:0:0:0-part1 -> ../../sda1
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 pci-0000:00:10.0-scsi-0:0:0:0-part5 -> ../../sda5

/dev/disk/by-id:
   total 0
   drwxr-xr-x 2 root root 260 Oct 23 02:34 .
lrwxrwxrwx 1 root root 9 Oct 23 02:34 wwn-0x6000c29ddc232994ce32cad1533c25e1 -> ../../sda
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 wwn-0x6000c296507a2c4be3b79a74f921d283 -> ../../sdb
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 scsi-36000c29ddc232994ce32cad1533c25e1 -> ../../sda
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 scsi-36000c296507a2c4be3b79a74f921d283 -> ../../sdb

lrwxrwxrwx 1 root root 9 Oct 23 02:34 ata-VMware_Virtual_SATA_CDRW_Drive_00000000000000000001 -
> ../../sr0
   drwxr-xr-x 6 root root 120 Oct 23 02:34 ..
lrwxrwxrwx 1 root root 10 Oct 23 02:34 wwn-0x6000c29ddc232994ce32cad1533c25e1-part2 ->
../../sda2
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 wwn-0x6000c29ddc232994ce32cad1533c25e1-part1 ->
../../sda1
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 scsi-36000c29ddc232994ce32cad1533c25e1-part2 ->
../../sda2
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 scsi-36000c29ddc232994ce32cad1533c25e1-part1 ->
../../sda1
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 wwn-0x6000c29ddc232994ce32cad1533c25e1-part5 ->
../../sda5
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 scsi-36000c29ddc232994ce32cad1533c25e1-part5 ->
../../sda5

/dev/disk/by-uuid:
   total 0
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 2017-02-15-20-36-22-00 -> ../../sr0
   drwxr-xr-x 6 root root 120 Oct 23 02:34 ..
drwxr-xr-x 2 root root 100 Oct 23 02:34 .
lrwxrwxrwx 1 root root 10 Oct 23 02:34 6d48a0d5-846b-48d3-bb0f-a626bb49c916 -> ../../sda1
   lrwxrwxrwx 1 root root 10 Oct 23 02:34 60f23142-a575-455d-8a99-6edca077d990 -> ../../sda5

/dev/disk/by-label:
   total 0
   lrwxrwxrwx 1 root root 9 Oct 23 02:34 Ubuntu-Server\x2016.04.2\x20LTS\x20amd64 -> ../../sr0
   drwxr-xr-x 6 root root 120 Oct 23 02:34 ..
```

```
drwxr-xr-x 2 root root 60 Oct 23 02:34 .
root@icps-worker-4:~#
```

O symlink de um dispositivo pode estar em qualquer um desses caminhos link: `/dev/disk/by-path`, `/dev/disk/by-id`, `/dev/disk/by-uuid` ou `/dev/disk/by-label`.

5. Anote o symlink e seu caminho do link. Para o dispositivo de exemplo `sdb`, `/dev/disk/by-id` é o caminho do link e `scsi-36000c296507a2c4be3b79a74f921d283` é o symlink. Para cada dispositivo que está sendo usado para a configuração do GlusterFS, é preciso incluir o `<link path>/<symlink>` no arquivo `config.yaml`. Para o dispositivo de exemplo `sdb`, você incluiria `/dev/disk/by-id/scsi-36000c296507a2c4be3b79a74f921d283` no arquivo `config.yaml`.

Usar symlinks criados manualmente

Em alguns ambientes, como o IBM Cloud VSI ou o SLES, nenhum symlink é gerado automaticamente para os dispositivos. Os symlinks devem ser criados manualmente, configurando regras udev (userspace /dev) customizadas. Ao criar o symlink, use os atributos que são exclusivos para o dispositivo.

O exemplo a seguir inclui etapas para gerar symlinks manualmente. As etapas podem variar de acordo com os sistemas operacionais e ambientes.

1. Obtenha informações sobre os atributos do dispositivo.

   ```
   info udevadm -- root --name=/dev/vdb
   ```

   A saída se assemelha ao código a seguir:

   ```
   P: /devices/pci0000:00/0000:00:10.0/virtio4/block/vdb
   N: vdb
   E: DEVNAME=/dev/vdb
   E: DEVPATH=/devices/pci0000:00/0000:00:10.0/virtio4/block/vdb
   E: DEVTYPE=disk
   E: MAJOR=253
   E: MINOR=16
   E: SUBSYSTEM=block
   E: TAGS=:systemd:
   E: USEC_INITIALIZED=6705725
   E: elevator=noop
   ```

   Use os atributos `DEVTYPE`, `SUBSYSTEM` e `DEVPATH` para criar o symlink do dispositivo.

2. Crie um arquivo de regras udev customizado.

   ```
   vi /lib/udev/rules.d/10-custom-icp.rules
   ```

   Inclua estas linhas de código no arquivo. Substitua os valores de atributo pelos valores de atributo do dispositivo.

   ```
    ENV{DEVTYPE}=="disk", ENV{SUBSYSTEM}=="block",
   ENV{DEVPATH}=="/devices/pci0000:00/0000:00:10.0/virtio4/block/vdb" SYMLINK+="disk/gluster-disk-
   1"
   ```

3. Recarregue as regras udev para criar os symlinks.

   ```
    udevadm control --reload-rules
    udevadm trigger --type=devices --action=change
   ```

4. Verifique se os symlinks foram criados.

   ```
   ls -ltr /dev/disk/gluster-*
   ```

   A saída se assemelha ao código a seguir:

   ```
   lrwxrwxrwx 1 root root 3 Jul  4 23:12 /dev/disk/gluster-disk-1 -> vdb
   ```

Configurando GlusterFS

Configure GlusterFS durante ou após a instalação do IBM® Cloud Private .

Se a pré-verificação de instalação do GlusterFS falhar, consulte Falha na pré-verificação de instalação do GlusterFS para resolver o problema.

Configurando o GlusterFS durante a instalação do IBM Cloud Private

Configure o GlusterFS ao instalar o cluster do IBM® Cloud Private.

Conclua estas etapas para configurar o GlusterFS:

1. Configure um grupo de hosts com os nós que você está usando para configurar o GlusterFS. É possível usar nós dedicados para o GlusterFS ou configurar o GlusterFS nos nós que também são usados como nós do trabalhador.
    - Se você estiver usando nós dedicados para configurar o GlusterFS, conclua estas etapas. Esses nós não são usados como nós do trabalhador.
        - Configure um grupo de hosts customizados com os nós de armazenamento do GlusterFS dedicado. Para obter mais informações sobre como incluir um grupo de hosts, consulte Definindo grupos de hosts customizados.

          A seguir está uma configuração de exemplo de um grupo de hosts com nós de armazenamento do GlusterFS dedicados. Inclua essa configuração no arquivo `/<installation_directory>/cluster/hosts`.

          Nota: os nós do trabalhador e os nós do GlusterFS não são os mesmos.

          ```
          [worker] 2.2.2.2 ...
          2.2.2.9 .
          .
          [hostgroup-glusterfs]
          6.6.6.6
          ...
          6.6.6.9
          ```
    - Se você estiver configurando o GlusterFS nos nós que também são usados como nós do trabalhador, conclua estas etapas.
        - No arquivo `/<installation_directory>/cluster/hosts`, inclua os mesmos nós que os nós do trabalhador e como um grupo de hosts customizados.
        - No arquivo `/<installation_directory>/cluster/config.yaml`, configure o parâmetro `no_taint_group` com o nome do grupo de hosts customizados.
          A seguir está uma configuração de exemplo de nós de armazenamento compartilhados. Inclua essa configuração no arquivo `/<installation_directory>/cluster/hosts`.

          ```
          [worker] 2.2.2.2 ...
          2.2.2.9 .
          .
          [hostgroup-glusterfs]
          2.2.2.2
          ...
          2.2.2.9
          ```

          A seguir está uma configuração de exemplo de inclusão do parâmetro `no_taint_group`. Inclua essa configuração no arquivo `/<installation_directory>/cluster/config.yaml`.

          ```
          no_taint_group: [ "hostgroup-glusterfs" ]
          ```
2. Se o firewall estiver ativado, inclua a lista de portas necessárias no arquivo `/<installation_directory>/cluster/config.yaml`. Localize a seção `firewall_enabled: true`. Inclua as portas a seguir para o grupo de hosts customizados que você criou com os nós de armazenamento do GlusterFS dedicados.
   A seguir está uma configuração de exemplo do grupo de hosts customizados `hostgroup-glusterfs`:

   ```
   firewall_open_ports:
   hostgroup-glusterfs:
    - 24007/tcp
    - 24008/tcp
    - 2222/tcp
    - 49152-49251/tcp
   ```

   Nota: se você usar nós do trabalhador existentes para implementar o GlusterFS, será necessário abrir manualmente todas as portas em todos os nós.

3. Ativar o GlusterFS Storage. Configure `storage-glusterfs: enabled` na lista de serviços de gerenciamento no arquivo `/<installation_directory>/cluster/config.yaml`.

   ```
   management_services:
    istio: disabled
    vulnerability-advisor: disabled
    storage-glusterfs: enabled
    storage-minio: disabled
   ```

4. Inclua a parte de código a seguir no arquivo `config.yaml`:

```
## GlusterFS Storage Settings
storage-glusterfs:
  nodes:
    - ip: <worker_node_m_IP_address>
      devices:
        - <link path>/<symlink of device aaa>
        - <link path>/<symlink of device bbb>
    - ip: <worker_node_n_IP_address>
      devices:
        - <link path>/<symlink of device ccc>
    - ip: <worker_node_o_IP_address>
      devices:
        - <link path>/<symlink of device ddd>
  storageClass:
    create: true
    name: glusterfs
    isDefault: false
    volumeType: replicate:3
    reclaimPolicy: Delete
    volumeBindingMode: Immediate
    volumeNamePrefix: icp
    additionalProvisionerParams: {}
    allowVolumeExpansion: true
  gluster:
    resources:
      requests:
        cpu: 500m
        memory: 512Mi
      limits:
        cpu: 1000m
        memory: 1Gi
  heketi:
    backupDbSecret: heketi-db-backup
    authSecret: "heketi-secret"
    maxInFlightOperations: "20"
    dbSyncupDelay: "10"
    tls:
      generate: true
      issuer: "icp-ca-issuer"
      issuerKind: "ClusterIssuer"
      secretName: ""
    resources:
      requests:
        cpu: 500m
        memory: 512Mi
      limits:
        cpu: 1000m
        memory: 1Gi
  nodeSelector:
    key: hostgroup
    value: glusterfs
  prometheus:
    enabled: true
    path: "/metrics"
    port: 8080
  tolerations: []
  podPriorityClass: "system-cluster-critical"
```

A seguir estão as descrições de parâmetros. Para obter uma lista de parâmetros disponíveis, consulte Configuração.
Nota: Se estiver configurando os parâmetros de classe de armazenamento padrão e de tipo de volume, certifique-se de usar a sintaxe correta. Use `isDefault: <true or false>` e `volumeType: <volume type>`.

- `ip` é o endereço IP do nó no qual você está configurando o GlusterFS.

- `devices` é o caminho completo para o symlink do dispositivo de armazenamento.

  Nota: não inclua o nome do dispositivo. Inclua os Symlinks .

- `storageClass` cria uma classe de armazenamento para o GlusterFS.

  - `create` é a opção para configurar a classe de armazenamento. O valor-padrão é `true`. Se você não deseja criar uma classe de armazenamento, especifique `false`.
  - `name` é o nome da classe de armazenamento. O valor padrão é `glusterfs`. O nome da classe de armazenamento deve estar em conformidade com a convenção de nomenclatura do Kubernetes: até 253 caracteres e apenas caracteres alfanuméricos minúsculos, – e `.`.
  - `isDefault` é a opção para tornar a classe de armazenamento de GlusterFS a classe de armazenamento padrão. O valor padrão é `false`. Se você deseja que essa classe de armazenamento seja o padrão, especifique `true`.
  - `volumeType` é um parâmetro de configuração para o volume. É possível configurar esse valor como `replicate:3`, que cria um volume de réplica de contagem 3. Os valores de parâmetro válidos são: `none`; `replicate:<replicate_count>`; `disperse:<data>:<redundancy_count>`. O valor padrão é `replicate: 3`.
    Nota: Se estiver configurando o GlusterFS em um único nó, configure `volumeType: none`.
  - `reclaimPolicy` é a política de recuperação da classe de armazenamento do GlusterFS. O valor padrão é `Excluir`.
  - `volumeBindingMode` é o modo de ligação do volume da classe de armazenamento do GlusterFS. O valor padrão é `Imediato`.
  - `volumeNamePrefix` é o prefixo do nome do volume da classe de armazenamento do GlusterFS. O valor padrão é `icp`.
  - `additionalProvisionerParams` são parâmetros extras do fornecedor de classe de armazenamento.
  - `allowVolumeExpansion` é para definir se a expansão de volume é permitida. O valor padrão é `true`.

    Nota: em qualquer momento posterior, será possível criar mais classes de armazenamento para usar o armazenamento GlusterFS. Consulte Criando uma classe de armazenamento para GlusterFS.

- `gluster` configura os parâmetros para a configuração do GlusterFS.

  - `resources` define o máximo e o mínimo de CPU e memória que são necessários.
    - `solicitações`
      - `cpu` é o número mínimo de CPU necessário. O valor padrão é 500 millicpu (m).
      - `memory` é o número mínimo de memória permitido. O valor padrão é 512 Mi.
    - `limites`
      - `cpu` é o número máximo de CPU permitido. O valor padrão é 1000m.
      - `memory` é o número máximo de memória permitido. O valor padrão é 1Gi.
- `heketi` define os parâmetros para a configuração do Heketi.
  - `backupDbSecret` é o banco de dados Heketi a ser submetido a backup em um segredo do kubernetes. O valor padrão é `heketi-db-backup`.
  - `authSecret` é o segredo que possui a senha criptografada do usuário 'admin' do Heketi. **Nota:** O segredo é criado automaticamente usando o `<default_admin_password>` que é incluído no arquivo `/<installation_directory>/cluster/config.yaml`.
  - `maxInFlightOperations` é o número máximo de solicitações simultâneas para o volume persistente. O valor padrão é 20.
  - `dbSyncupDelay` é o atraso em segundos para sincronizar dados no banco de dados Heketi com o segredo de backup. O valor padrão é 10.
  - `tls` define os parâmetros de configuração da segurança da camada de transporte (TLS).
    - `generate` é uma sinalização booleana. Quando configurada para `true`, a sinalização cria um certificado usando a autoridade de certificação (CA) IBM Cloud Private. Se não usar a sinalização, você deverá criar um segredo contendo uma chave privada, um certificado TLS e um certificado de CA. Deve-se fornecer o nome do segredo no parâmetro `heketi.tls.secretName`.
    - `issuer` é o nome do emissor do certificado IBM Cloud Private.
    - `issuerKind` é o tipo de CA do qual os certificados x509 assinados são obtidos. Os valores válidos são: `ClusterIssuer` e `Issuer`.
    - `secretName` é o certificado que você fornece como objeto do segredo. O nome da chave deve ser `tls.key` e o nome do certificado deve ser `tls.crt`. Deve-se configurar esse parâmetro durante a configuração da sinalização `heketi.tls.generate` para `false`.
  - `resources` é o máximo e o mínimo de CPU e memória que são necessários.
    - `solicitações`
      - `cpu` é o número mínimo de CPU necessário. O valor padrão é 500m.
      - `memory` é o número mínimo de memória permitido. O valor padrão é 512Mi.
    - `limites`

        - `cpu` é o número máximo de CPU permitido. O valor padrão é 1000m.
        - `memory` é o número máximo de memória permitido. O valor padrão é 1Gi.
- `prometheus` configura a opção para ativar ou desativar o monitoramento do prometheus.
  Nota: o Prometheus fornece um painel de estatísticas sobre o funcionamento e o uso do GlusterFS. Se você deseja usar o Prometheus para o GlusterFS, deve-se ativá-lo agora. Não será possível ativá-lo depois que seu cluster do IBM Cloud Private estiver instalado.
    - `enabled` é a configuração para ativar ou desativar o monitoramento do prometheus. O valor padrão é `true`.
    - `path` é o caminho do Heketi para puxar as métricas. O valor padrão é `/metrics`.
    - `port` é a porta na qual o serviço do Heketi está exposto. O valor padrão é `8080`.
- `nodeSelector` configura o rótulo do seletor de nó para os recursos de armazenamento. Use o nome do grupo de hosts customizados que você criou para a configuração do GlusterFS. Por exemplo, o grupo de hosts customizados `hostgroup-glusterfs` cria nós com o rótulo `hostgroup=glusterfs`.
    - `key` é a chave da etiqueta. O valor padrão é `hostgroup`.
    - `value` é o valor do rótulo. O valor padrão é `glusterfs`.
- `podPriorityClass` é a preferência de classe de prioridade para os pods GlusterFS e Heketi.
- `tolerations` é a lista de tolerâncias que você deseja especificar, independentemente se os nós de armazenamento possuem contaminações adicionais.
- A configuração do GlusterFS está concluída. Prossiga com a instalação do IBM Cloud Private .

Configurando GlusterFS após a instalação do IBM Cloud Private

Configure o GlusterFS depois de instalar o cluster do IBM® Cloud Private.

Configurar o GlusterFS instalando o gráfico Helm

Instale o gráfico do Helm do GlusterFS Versão 1.4.0. Para obter mais informações, consulte Cluster de armazenamento do GlusterFS.

Configurar o GlusterFS como um serviço de complemento

Conclua estas etapas para configurar o GlusterFS:

1. Ativar o GlusterFS Storage. Consulte a etapa 3 .
2. Atualize o `config.yaml` arquivo. Consulte a etapa 4 .
3. Execute o comando de complemento para configurar o GlusterFS:

```
docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
```

Criando uma classe de armazenamento para GlusterFS

Crie uma classe de armazenamento para provisionar armazenamento do GlusterFS.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Para criar uma classe de armazenamento para GlusterFS, especifique os valores de campo a seguir na definição de classe de armazenamento:

- metadados:
    - `name` é o nome da classe de armazenamento.
      Nota: O nome deve consistir em caracteres alfanuméricos minúsculos e deve começar e terminar com um caractere alfanumérico. É possível usar apenas esses caracteres especiais no nome: `-` e `.`.
- parâmetros:
    - `resturl` é a URL REST do Heketi que provisiona volumes.

      `https://<Heketi_service_cluster_IP>:<Heketi_service_port>`

        - Para obter o serviço Heketi, execute o comando a seguir:

```
kubectl -n kube-system get service  -l glusterfs=heketi-service
```

- Para obter o IP do cluster de serviço Heketi, execute o comando a seguir:

  ```
  kubectl -n kube-system get service <heketi-service-name> -
  o=jsonpath='{.spec.clusterIP}'
  ```

- Para obter o número da porta de serviço Heketi, execute o comando a seguir:

  ```
  kubectl -n kube-system get service <heketi-service-name> -
  o=jsonpath='{.spec.ports[0].port}'
  ```

- O `volumetype` é o parâmetro opcional para o tipo de volume. Os valores de parâmetro válidos são `none`, `replicate:<replicate_count>` e `disperse:<data>:<redundancy_count>`. Se você não especificar um tipo de volume, o fornecedor configurará o tipo de volume para `replicate:3`.
- O `volumenameprefix` é um prefixo para o nome do volume. Por padrão, volumes provisionados dinamicamente possuem o esquema de nomenclatura do formato vol_ . Com o parâmetro `volumenameprefix` na classe de armazenamento, é possível prefixar o nome do volume desejado.

Para criar uma classe de armazenamento para GlusterFS, deve-se concluir as etapas que são mostradas no exemplo a seguir:

1. Crie um arquivo YAML que seja denominado `glusterfs.yaml` e que contenha as definições de classe de armazenamento a seguir:

   ```
   apiVersion: storage.k8s.io/v1
   kind: StorageClass
   metadata:
     name: glusterfs
   provisioner: kubernetes.io/glusterfs
   parameters:
     resturl: "https://10.0.0.185:8080"
     restuser: "admin"
     secretName: "heketi-secret"
     secretNamespace: "kube-system"
     volumetype: replicate:3
     volumenameprefix: "icp"
   ```

   Nota: se o seu cluster GlusterFS estiver em um ambiente do IBM® Cloud Private, será necessário usar os valores de parâmetros a seguir:

   - **restuser** deve ser "admin"
   - **secretName** é o mesmo segredo que você usou para autenticação Heketi durante a instalação do GlusterFS
   - **secretNamespace** deve ser "kube-system"

   Para tornar isso a sua classe de armazenamento padrão, inclua a anotação `storageclass.kubernetes.io/is-default-class` e configure-a como `true`.

2. Crie a classe de armazenamento:

   ```
   kubectl criar -f glusterfs.yaml
   ```

   A saída se assemelha ao código a seguir:

   ```
   storageclass "glusterfs" criado
   ```

3. Verifique se a classe de armazenamento está criada:

   ```
   o kubectl descreve sc glusterfs
   ```

Verificando a Configuração do GlusterFS

Verifique se a configuração do GlusterFS está correta.

Para verificar a configuração, conclua as etapas a seguir:

1. Verifique se o servidor Heketi está em execução e se está acessível. Execute o comando a seguir no nó principal:

   ```
   curl -k https://<Heketi_service_cluster_IP>:<Heketi_server_port>/hello
   ```

Se a saída de comando incluir `Hello from Heketi`, o servidor Heketi estará acessível.

2. Implemente um aplicativo que requeira armazenamento persistente. Implemente o aplicativo a partir de seu catálogo do IBM® Cloud Private do IBM® Cloud Private . Por exemplo, implemente o aplicativo `ibm-postgres-dev`. Para obter mais informações, consulte PostgreSQL .

Recuperando o volume GlusterFS

É possível usar PersistentVolumeClaims (PVCs) para solicitar o volume GlusterFS que está configurado nos nós em seu cluster do IBM Cloud Private.

É possível criar um máximo de 100 PVCs para o volume do GlusterFS.

Para obter mais informações sobre como criar um PVC, consulte Criando PersistentVolumeClaims usando a janela "Criar recurso".

A seguir há um exemplo de configuração de YAML de um PVC para solicitar o volume GlusterFS:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: glusterpvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: glusterfs
```

Gerenciando seu cluster GlusterFS

Gerencie seu cluster do GlusterFS.

- Aumentando a capacidade de seu volume GlusterFS existente
- Recuperando dados de um volume GlusterFS
- Aumentando a capacidade de armazenamento de um cluster GlusterFS
- Alterando o segredo Heketi

Aumentando a capacidade de seu volume do GlusterFS existente

É possível aumentar a capacidade de seu volume GlusterFS existente.

Conclua estas etapas:

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Assegure-se de que o parâmetro `AllowVolumeExpansion` esteja configurado como `True` na definição da classe de armazenamento.

   ```
   kubectl describe storageclass <storage-class-name>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   # kubectl describe storageclass  glusterfs
   Name:                  glusterfs
   IsDefaultClass:        No
   Annotations:           storageclass.beta.kubernetes.io/is-default-class=false
   Provisioner:           kubernetes.io/glusterfs
   Parameters:            resturl=https://10.0.0.236:8080,restuser=admin,secretName=heketi-
   secret,secretNamespace=kube-system,volumenameprefix=icp,volumetype=replicate:3
   AllowVolumeExpansion:  True
   MountOptions:          <none>
   ReclaimPolicy:         Delete
   ```

```
VolumeBindingMode:     Immediate
Events:                <none>
```

3. Obtenha informações sobre as solicitações de volumes persistentes (PVCs).

   ```
   Kubectl get pvc
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME           STATUS    VOLUME                                     CAPACITY   ACCESS MODES
   STORAGECLASS   AGE
   test-storage   Bound     pvc-f4a210f4-aa89-11e8-aed5-00000a290936   1Gi        RWO
   glusterfs      1m
   ```

4. Edite o PVC que está ligado ao volume GlusterFS. Atualize a capacidade conforme necessário e salve as mudanças.

   ```
   teste kubectl de teste pvc-storage-storage
   ```

5. Obtenha informações sobre o PVC atualizado.

   ```
   Kubectl get pvc
   ```

   A saída se assemelha ao texto a seguir:

   ```
   NAME           STATUS    VOLUME                                     CAPACITY   ACCESS MODES
   STORAGECLASS   AGE
   test-storage   Bound     pvc-f4a210f4-aa89-11e8-aed5-00000a290936   10Gi       RWO
   glusterfs      2m
   ```

   A capacidade do volume é aumentada em 10 Gigabytes.

Recuperando dados de um volume GlusterFS

Recupere dados de um volume GlusterFS.

O GlusterFS usa o LVM (Gerenciador de Volume Lógico) para criar e gerenciar o armazenamento. O GlusterFS replica dados para volumes entre os nós em seu cluster GlusterFS. Quando você precisar recuperar dados, poderá acessar os volumes lógicos nos quais os dados são replicados.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Para recuperar dados de volumes lógicos, conclua as etapas a seguir em todos os nós em seu cluster GlusterFS:

1. Execute o comando a seguir para identificar volumes lógicos que possuem um caminho de volume lógico.

   ```
   lvdisplay | grep "LV Path"
   ```

   A saída é semelhante ao texto a seguir.

   ```
   WARNING: Not using lvmetad because config setting use_lvmetad=0.
   WARNING: To avoid corruption, rescan devices to make changes visible (pvscan --cache).
   LV Path /dev/vg_88f4bd7e79e88798c21b45dd9ca854ba/brick_d3f350725ab55fe9504a64d5213cb393
   ```

   Você monta o caminho do volume lógico da mesma forma como monta um disco regular.

2. Crie um diretório na pasta `/mnt` para montar o volume lógico.

   ```
   mkdir -p /mnt/<mount directory name>
   ```

   No exemplo a seguir, o nome do tijolo é usado como o nome de diretório.

   ```
   mkdir -p /mnt/brick_d3f350725ab55fe9504a64d5213cb393
   ```

3. Monte o volume lógico.

   ```
   mount <LV path> /mnt/<mount directory name>
   ```

   O comando é semelhante ao texto a seguir:

   ```
   mount /dev/vg_88f4bd7e79e88798c21b45dd9ca854ba/brick_d3f350725ab55fe9504a64d5213cb393
   /mnt/brick_d3f350725ab55fe9504a64d5213cb393
   ```

4. Verifique se o volume lógico está montado no diretório `/mnt`. Execute o comando a seguir:

   ```
   ls /mnt
   ```

   A saída se assemelha ao texto a seguir:

   ```
   brick_d3f350725ab55fe9504a64d5213cb393
   ```

5. Mude para o diretório `/mnt/<mount directory name>/brick/`.

   ```
   cd /mnt/<mount directory name>/brick/
   ```

6. Visualize os dados recuperados. Execute o comando a seguir:

   ```
   ls
   ```

   A saída se assemelha ao texto a seguir:

   ```
   heketi.db
   ```

   No exemplo, `heketi.db` foi recuperado do volume lógico.

Aumentando a capacidade de armazenamento de um cluster GlusterFS

Aumente a capacidade de armazenamento de seu cluster GlusterFS.

É possível aumentar a capacidade de armazenamento de seu cluster GlusterFS, incluindo um dispositivo em um nó de armazenamento existente ou incluindo um nó de armazenamento no cluster do GlusterFS.

- Incluindo um dispositivo em um nó de armazenamento do GlusterFS
- Incluindo um nó de armazenamento no cluster do GlusterFS

Antes de iniciar, configure a CLI `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

Nota: Sempre que você usar `-- heketi-cli` em um comando `kubectl`, deve fornecer os parâmetros `--user admin` e `--secret <admin_password>` para autenticação de pod Heketi. `<admin_password>` é a senha que é usada para criar o segredo de autenticação do Heketi durante a instalação do GlusterFS.

Incluindo um dispositivo em um nó de armazenamento do GlusterFS

É possível incluir um dispositivo em um nó de armazenamento que existe em seu cluster do GlusterFS. O dispositivo deve atender aos requisitos do dispositivo GlusterFS. Consulte os Requisitos de hardware.

Após incluir o dispositivo, para torná-lo parte da topologia do Heketi, primeiro recupere as informações necessárias, em seguida, execute o comando para atualizar a topologia.

Recuperando as informações necessárias

É necessário o ID do nó no qual você está incluindo um dispositivo. Para obter o ID do nó, execute os comandos a seguir:

1. Obtenha o nome do pod do Heketi.

   ```
   kubectl -n kube-system get pod -l glusterfs=heketi-pod
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME READY STATUS RESTARTS AGE heketi-7b69bb4d48-g4kx9 1/1 Running 0 1h
   ```

2. Obtenha o ID do cluster.

   ```
   kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <admin_password> cluster list
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin -
   -secret admin cluster list

   Clusters:
   ```

```
Id:2a47436e012604919dae5fbb4fc3899c [file][block]
#
```

3. Obtenha o ID do nó:

   ```
   kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <admin_password> cluster info <cluster ID>
   ```

   O comando retorna uma lista de nós.

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin -
   -secret admin cluster info 2a47436e012604919dae5fbb4fc3899c

   Cluster id: 2a47436e012604919dae5fbb4fc3899c
   Nodes:
   6c4c66317f5650e7b61057fede8dbe25
   8164c019b2e7c67b224709a97d05d0e9
   de3d0a9a21868a09f92fb50e166cceda
   Volumes:
   896044a59b18316790ff5554455a7a9e
   b971b2bb34703143e094f9b5241e4ddf
   f3f93e7eddd48e95779361ff35bd2222
   Block: true

   File: true
   ```

4. Identifique o nó que possui o novo dispositivo. É possível fazer isso pelo endereço IP do nó. Para recuperar o endereço IP do nó, execute o comando a seguir para todos os nós. Anote o ID do nó de armazenamento que possui o novo dispositivo.

   ```
   kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <admin_password> node info <node ID>
   ```

   Nota: obtenha o symlink do novo dispositivo. Para obter mais informações sobre como obter o symlink, consulteSymlinks.

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin -
   -secret admin node info de3d0a9a21868a09f92fb50e166cceda

   Node Id: de3d0a9a21868a09f92fb50e166cceda
   State: online
   Cluster Id: 2a47436e012604919dae5fbb4fc3899c
   Zone: 1
   Management Hostname: 192.168.0.96
   Storage Hostname: 192.168.0.96
   Devices:
   Id:4c6ba3295f71a78515d6bd5d5aad58ae   Name:/dev/disk/by-path/pci-0000_00_10_0    State:online
   Size (GiB):24      Used     (GiB):2       Free (GiB):22
   #
   ```

Atualizando a topologia

Atualize a topologia do Heketi com as informações sobre o novo dispositivo. Execute o comando a seguir:

```
kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --secret
<admin_password> device add --name=<device symlink> --node=<Node ID>
```

O seguinte é um comando e uma saída de amostra:

```
kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin --
secret admin device add --name=/dev/disk/by-path/pci-0000_00_11_0 --
node=de3d0a9a21868a09f92fb50e166cceda
Device added successfully
#
```

Incluindo um nó de armazenamento no cluster do GlusterFS

É possível incluir um nó de armazenamento em seu cluster do GlusterFS. O dispositivo no novo nó de armazenamento deve atender aos requisitos do dispositivo do GlusterFS. Consulte os Requisitos de hardware.

Antes de continuar, conclua estas etapas:

- Inclua o novo nó no arquivo `config.yaml` na pasta `/<installation_directory>/cluster`. Consulte Configurando GlusterFS durante a IBM Cloud Private instalação .
- Inclua o novo nó no grupo de hosts customizados que você criou com os nós de armazenamento do GlusterFS. Para obter mais informações, consulte Incluindo um grupo de hosts.

Conclua as etapas a seguir usando a CLI do kubectl:

1. Obtenha o nome do pod do GlusterFS.

   ```
   kubectl -n kube-system get pod -owide -l glusterfs=pod
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system get pod -owide -l glusterfs=pod

   NAME              READY     STATUS              RESTARTS   AGE       IP              NODE
   glusterfs-4srll   1/1       Running             1          3h        192.168.0.96
   192.168.0.96
   glusterfs-5zp55   0/1       ContainerCreating   0          7s        192.168.0.139
   192.168.0.139
   glusterfs-j2t6d   1/1       Running             0          3h        192.168.0.56
   192.168.0.56
   glusterfs-q8mcm   1/1       Running             0          3h        192.168.0.184
   192.168.0.184
   #
   ```

2. Execute o comando de análise do peer em qualquer um dos pods GlusterFS existentes.

   ```
   kubectl -n kube-system exec <GlusterFS pod name> -- gluster peer probe <IP address or host name
   of the new node>
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec glusterfs-4srll -- gluster peer probe 192.168.0.139
   peer probe: success.
   #
   ```

3. Obtenha o nome do pod do Heketi.

   ```
   kubectl -n kube-system get pod -l glusterfs=heketi-pod
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME READY STATUS RESTARTS AGE heketi-7b69bb4d48-g4kx9 1/1 Running 0 1h
   ```

4. Obtenha o ID do cluster.

   ```
   kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <admin_password> cluster list
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin -
   -secret admin cluster list

   Clusters:
   Id:2a47436e012604919dae5fbb4fc3899c [file][block]
   #
   ```

5. Inclua o novo nó para o cluster do GlusterFS.

   ```
   kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <admin_password> node add --zone=1 --cluster=<cluster ID> --management-host-name=<IP
   address or host name of the new node> --storage-host-name=<IP address of the new node>
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin -
   -secret admin node add --zone=1 --cluster=2a47436e012604919dae5fbb4fc3899c --management-host-
   name=192.168.0.139 --storage-host-name=192.168.0.139

   Node information:
   Id: 5df753379b9629eda0c5eb71687ee2ea
   State: online
   ```

```
Cluster Id: 2a47436e012604919dae5fbb4fc3899c
Zone: 1
Management Hostname 192.168.0.139
Storage Hostname 192.168.0.139
#
```

6. Atualize a topologia com as informações sobre o dispositivo do novo nó.

   ```
   kubectl -n kube-system exec <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <admin_password> device add --name=<device symlink> --node=<Node ID>
   ```

   Nota: inclua o symlink do dispositivo como o nome. Para obter mais informações sobre como obter o symlink, consulteSymlinks.

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl -n kube-system exec heketi-7b69bb4d48-g4kx9 -- heketi-cli --insecure-tls --user admin -
   -secret admin device add --name=/dev/disk/by-path/pci-0000_00_10_0 --
   node=5df753379b9629eda0c5eb71687ee2ea
   Device added successfully
   #
   ```

Alterando o segredo do Heketi

Mude o segredo que é usado pelo serviço Heketi para autenticação.

O pod de serviço Heketi usa os parâmetros `--user admin` e `--secret <admin_password>` para autenticação. `<admin_password>` é a senha que é usada para criar o segredo de autenticação de Heketi durante a instalação do GlusterFS.

Para mudar o segredo que o serviço Heketi usa, conclua as etapas a seguir:

1. Configure o IBM® Cloud Private CLI. Para obter mais informações, consulte Gerenciando seu cluster com a CLI do IBM® Cloud Private.
2. Assegure-se de que a CLI `kubectl` esteja configurada. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
3. Obtenha o nome do pod do Heketi.

   ```
   kubectl -n kube-system get pod -l glusterfs=heketi-pod
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME READY STATUS RESTARTS AGE heketi-7b69bb4d48-g4kx9 1/1 Running 0 1h
   ```

4. Verifique se é possível acessar o pod Heketi usando a senha existente. Execute este comando de amostra:

   ```
   kubectl -n kube-system exec -it <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
   secret <existing_password> cluster list
   ```

   A saída se assemelha ao código a seguir:

   ```
   Clusters: Id: 53e34819cc8ba1b28c61718cf0248e5d
   ```

5. Atualize o segredo Heketi.

   ```
   cloudctl pm update-secret kube-system <heketi-secret-name> -d admin_password=<new_password>
   ```

   A saída se assemelha ao código a seguir:

   ```
   Mudanças secretas são válidos.
   1 deployments will be updated:
    - heketi

   Continuar? (S/N) > Y OK

   1 deployments updated:
    - heketi
   ```

6. Verifique se é possível acessar o pod Heketi usando a nova senha. Execute este comando de amostra:

```
kubectl -n kube-system exec -it <Heketi pod name> -- heketi-cli --insecure-tls --user admin --
secret <new_password> cluster list
```

A saída se assemelha ao código a seguir:

```
Clusters: Id: 53e34819cc8ba1b28c61718cf0248e5d
```

Monitorando o GlusterFS

É possível monitorar o funcionamento de seu cluster GlusterFS no painel de monitoramento do cluster do IBM® Cloud Private.

O Heketi exporta dados compatíveis com Prometheus como um terminal em `/metrics`. Para ativar a exportação de métricas do Prometheus no servidor Heketi, defina o parâmetro de configuração `prometheus.enabled: true` enquanto o gráfico do Helm do GlusterFS é instalado.

No gráfico do Helm do GlusterFS Versão 1.4.0, a comunicação Heketi é criptografada por `tls`. Portanto, o Prometheus não coleta métricas por padrão. Após o gráfico do GlusterFS ser instalado, deve-se concluir os seguintes procedimentos para permitir que o Prometheus colete métricas do terminal Heketi. Também é possível visualizar as métricas no painel Grafana do IBM Cloud Private.

1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Obtenha o endereço IP do serviço Heketi.

   ```
   kubectl -n kube-system -l glusterfs=heketi-service get service
   ```

   A seguir está uma saída de amostra.

   ```
   NAME                                          TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)
   AGE
   storage-glusterfs-glusterfs-heketi-service    ClusterIP   10.0.150.224   <none>        8080/TCP
   18h
   ```

   Na saída de amostra, `10.0.150.224` é o endereço IP do serviço Heketi.
3. Edite o mapa de configuração `monitoring-prometheus`.

   ```
   kubectl edit configmap -n kube-system monitoring-prometheus
   ```
4. Inclua uma tarefa de configuração de descarte para o terminal Heketi especificando o número da porta e o endereço IP do serviço Heketi.

   ```
   - job_name: heketi
    static_configs:
      - targets:
        - 10.0.150.224:8080
    metrics_path: /metrics
    scheme: https
    tls_config:
      insecure_skip_verify: true
   ```

Atualizando o GlusterFS

É possível fazer upgrade de seu cluster GlusterFS que foi configurado no IBM® Cloud Private Versão 3.1.2.

Seu cluster GlusterFS é submetido a upgrade automaticamente se você configurou o GlusterFS no IBM Cloud Private Versão 3.1.2 de uma das maneiras a seguir:

- Você configurou o GlusterFS enquanto instalou o cluster do IBM Cloud Private.
- Você usou o comando de complemento para configurar o GlusterFS depois de instalar o IBM Cloud Private.

No IBM Cloud Private Versão 3.1.2 ou Versão 3.1.0, se você configurou o GlusterFS instalando o gráfico do Helm, conclua estas etapas para fazer upgrade do GlusterFS Helm chart para a versão 1.4.0 no IBM Cloud Private Versão 3.2.0:

O IBM Cloud Private Versão 3.1.0 usa o GlusterFS Helm chart Versão 1.1.0, o IBM Cloud Private Versão 3.1.1 usa o GlusterFS Helm chart Versão 1.2.0 e o IBM Cloud Private Versão 3.1.2 usa o GlusterFS Helm chart Versão 1.3.0. Se você não deseja fazer upgrade, é possível continuar a usar o GlusterFS Helm chart Versão 1.3.0, 1.2.0 ou 1.1.0, o que for aplicável, com o IBM Cloud Private Versão 3.2.0.

Nota: o IBM Cloud Private Versão 3.1.0 usa o GlusterFS Helm chart Versão 1.1.0, o IBM Cloud Private Versão 3.1.1 usa o GlusterFS Helm chart Versão 1.2.0 e o IBM Cloud Private Versão 3.1.2 usa o GlusterFS Helm chart Versão 1.3.0. Se você não deseja fazer upgrade, é possível continuar a usar o GlusterFS Helm chart Versão 1.3.0, 1.2.0 ou 1.1.0, o que for aplicável, com o IBM Cloud Private Versão 3.2.0.

Nota: no IBM Cloud Private Versão 3.2.0, o servidor GlusterFS é submetido a upgrade para a Versão 4.1.5. Deve-se fazer upgrade manualmente do cliente GlusterFS para a Versão para 4.1.5.

Antes de iniciar, assegure-se de configurar as CLIs a seguir:

- CLI do Helm
- CLI do kubectl

Conclua estas etapas para fazer upgrade do GlusterFS:

1. Patch GlusterFS daemonset e implementação Heketi. Execute estes comandos:
    1. Obtenha a versão mais recente do gráfico Helm GlusterFS que está instalada.

        ```
        helm list -- tls | grep gluster
        ```

        O seguinte é uma saída de amostra:

        ```
        gfsissue                        1               Thu Jan 25 22:31:53 2019        DEPLOYED
        ibm-glusterfs-1.2.0                     kube-system
        ```

    2. Obtenha o daemonset GlusterFS.

        ```
        kubectl -n kube-system get daemonset -l glusterfs=daemonset
        ```

        O seguinte é uma saída de amostra:

        ```
        NAME                            DESIRED   CURRENT   READY     UP-TO-DATE   AVAILABLE
        NODE SELECTOR         AGE
        gfsissue-glusterfs-daemonset   4         4         4         4            4
        hostgroup=glusterfs   4m
        ```

    3. Corrige o daemonset.

        ```
        kubectl -n kube-system patch daemonset gfsissue-glusterfs-daemonset  --type json  -
        p='[{"op": "remove", "path": "/spec/selector/matchLabels/version"}]'
        ```

        O seguinte é uma saída de amostra:

        ```
        daemonset.extensions/gfsemita-glusterfs-daemonset patched
        ```

    4. Obtenha informações sobre a implementação do Heketi.

        ```
        kubectl -n kube-system get deployment -l glusterfs=heketi-deployment
        ```

        O seguinte é uma saída de amostra:

        ```
        NAME                                 DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
        gfsissue-glusterfs-heketi-deployment   1         1         1            0           3m
        ```

    5. Implementação de Heketi da Correção.

        ```
        kubectl -n kube-system patch deployment gfsissue-glusterfs-heketi-deployment --type json
        -p='[{"op": "remove", "path": "/spec/selector/matchLabels/version"}]'
        ```

        O seguinte é uma saída de amostra:

        ```
        deployment.extensions/gfsemita-glusterfs-heketi-deployment patched
        ```

2. Efetue logon no console de gerenciamento.
3. Localize o gráfico Helm GlusterFS.
4. Clique em **Upgrade** para fazer upgrade do gráfico para a Versão 1.4.0.

Nota: o upgrade com tempo de inatividade zero do cluster GlusterFS não é suportado quando você faz upgrade do IBM Cloud Private Versão 3.1.1 para a Versão 3.1.2. Enquanto os pods do GlusterFS são atualizados, os volumes GlusterFS não ficam acessíveis para pods de aplicativos até que os pods GlusterFS estejam em execução.

Desinstalar GlusterFS

Desinstale o GlusterFS de seu cluster do IBM® Cloud Private .

Conclua estas etapas para desinstalar o GlusterFS.

1. Exclua o gráfico de Helm.

   ```
   leme delete -- purge < release_name> -- tls
   ```

   Nota: o comando de exclusão do Helm exclui todos os objetos, exceto o `heketi.backupDbSecret`. Deve-se excluir o objeto `heketi.backupDbSecret` manualmente, caso ele não seja necessário.

2. Remova os diretórios de configuração daemon Heketi e Gluster de cada nó de armazenamento.

   ```
   rm -rf /var/lib/heketi rm -rf /var/lib/glusterd rm -rf /var/log/glusterfs
   ```

3. Desative o GlusterFS. Configure `storage-glusterfs: disabled` na seção `management_services` do arquivo `<installation_directory>/cluster/config.yaml`.

```
management_services: istio: disabled vulnerability-advisor: disabled storage-glusterfs: disabled
storage-minio: disabled
```

Preparando seus nós para uma reinstalação do GlusterFS ou IBM Cloud Private

Se você configurou o GlusterFS em seu cluster e deseja reinstalar o GlusterFS ou o IBM® Cloud Private no mesmo cluster, primeiramente deve-se preparar seus nós para a reinstalação.

Excluir o Gráfico de Helm

1. Assegure-se de que a CLI do Helm esteja configurada. Para obter mais informações, consulte Instalando a CLI do Helm (helm).

2. Obtenha o nome da liberação.

   ```
   helm list -- tls | grep gluster
   ```

3. Exclua o gráfico.

   ```
   leme delete -- purge < release_name> -- tls
   ```

Excluir o banco de dados Heketi de backup

1. Assegure-se de que a CLI do kubectl esteja configurada. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Obtenha o segredo Heketi.

   ```
   kubectl -n kube-system get secret | grep heketi
   ```

3. Excluir banco de dados Heketi de backup.

   ```
   kubectl -n kube-system delete secret <heketi-db-backup-name>
   ```

Remover os diretórios de configuração

Remova os diretórios de configuração do daemon Heketi e Gluster de cada nó de armazenamento que é usado para reinstalação. Execute estes comandos:

```
rm -rf /var/lib/heketi rm -rf /var/lib/glusterd rm -rf /var/log/glusterfs
```

Prepare os discos a serem usados para a instalação do GlusterFS

É possível reutilizar os discos ou incluir novos discos para uma reinstalação do GlusterFS.

- Se estiver incluindo novos discos, consulte Requisitos de hardware para obter informações sobre os requisitos de disco. Após incluir os discos, deve-se reiniciar os nós para que o sistema identifique os discos.
- Se você estiver reutilizando os discos, conclua estas etapas:

  Nota: o processo de limpeza de disco pode não funcionar em alguns ambientes. Se isso acontecer, poderá ser necessário usar discos novos.

  - Faça backup dos dados nos discos que foram usados em uma instalação anterior. As etapas a seguir podem causar uma perda de dados nos discos antigos.
  - Execute estes comandos para remover os volumes do GlusterFS:

    1. Remova os volumes lógicos e o grupo de volumes.

       ```
       lvscan | grep 'vg_' | awk '{print $2}' | xargs -n 1 lvremove -y
       vgscan | grep 'vg_' | awk '{print $4}' | xargs -n 1 vgremove -y
       ```

    2. Varra os volumes físicos. Anote o nome do volume físico. O nome do volume físico é a sequência após **PV**.

       ```
       pvscan
       ```

       A seguir está uma saída de amostra do comando:

       ```
       PV /dev/sdb VG vg_7a08667f373f2d6fe9977de3b29a754e lvm2 [249.87 GiB / 249.87 GiB
       free]
       PV /dev/sda5 VG ubuntuicp-vg lvm2 [299.52 GiB / 8.00 MiB free]
       Total: 2 [549.39 GiB] / in use: 2 [549.39 GiB] / in no VG: 0 [0][image]
       ```

       Por exemplo, na saída de amostra, `/dev/sdb` é o nome do volume físico.

    3. Remova todos os volumes físicos.

       ```
       pvremove <physical_volume_name>
       ```

    4. Obtenha o nome do dispositivo para o disco que é usado para o GlusterFS. O nome do dispositivo é a sequência após **Disco**.

       ```
       fdisk -l
       ```

       A seguir está uma saída de amostra do comando:

       ```
       Disk /dev/sda: 300 GiB, 322122547200 bytes, 629145600 sectors
       Units: sectors of 1 * 512 = 512 bytes
       Sector size (logical/physical): 512 bytes / 512 bytes
       I/O size (minimum/optimal): 512 bytes / 512 bytes
       Disklabel type: dos
       Disk identifier: 0xfa3b0bf6

       Device Boot Start End Sectors Size Id Type
       /dev/sda1 * 2048 999423 997376 487M 83 Linux
       /dev/sda2 1001470 629143551 628142082 299.5G 5 Extended
       /dev/sda5 1001472 629143551 628142080 299.5G 8e Linux LVM

       Disk /dev/sdb: 250 GiB, 268435456000 bytes, 524288000 sectors
       Units: sectors of 1 * 512 = 512 bytes
       Sector size (logical/physical): 512 bytes / 512 bytes
       I/O size (minimum/optimal): 512 bytes / 512 bytes

       Disk /dev/mapper/ubuntuicp--vg-root: 298.6 GiB, 320574849024 bytes, 626122752
       sectors
       Units: sectors of 1 * 512 = 512 bytes
       Sector size (logical/physical): 512 bytes / 512 bytes
       I/O size (minimum/optimal): 512 bytes / 512 bytes

       Disk /dev/mapper/ubuntuicp--vg-swap_1: 976 MiB, 1023410176 bytes, 1998848 sectors
       Units: sectors of 1 * 512 = 512 bytes
       Sector size (logical/physical): 512 bytes / 512 bytes
       I/O size (minimum/optimal): 512 bytes / 512 bytes
       ```

       Por exemplo, na saída de amostra, `/dev/sdb` é o nome do disco.

    5. Apague todas as assinaturas de sistema de arquivos, raid e tabela de partição.

```
wipefs --all --force <device_name>
```

Em seguida, conclua as tarefas em Configurando o GlusterFS após a instalação do IBM Cloud Private.

Resolução de problemas do GlusterFS

Revise problemas do GlusterFS frequentemente encontrados.

- Falha na pré-verificação de instalação do GlusterFS
- Dispositivo GlusterFS não localizado após reinicialização do sistema
- Travamento do nó GlusterFS
- A reinstalação do IBM Cloud Private não resolve problemas do GlusterFS
- Não é possível criar um PersistentVolumeClaim
- A reinicialização simultânea de nós do trabalhador faz com que o GlusterFS falhe
- Não é possível criar ou excluir um volume persistente ou solicitação de volume persistente
- O status do nó GlusterFS é mostrado como peer rejeitado
- A exclusão de uma solicitação de volume persistente do GlusterFS pode mostrar o status do volume persistente como com falha
- O pod GlusterFS não está planejado após a reinicialização de um nó
- Incompatibilidade de uso do disco do Heketi
- Pod Heketi falha ao iniciar após a reinicialização do Docker
- Pod Heketi parado no estado de inicialização quando o firewall é ativado e as portas necessárias não são abertas
- O pod GlusterFS pode falhar ao iniciar após a reinicialização de um nó do IBM® Z

Falha na pré-verificação de instalação do GlusterFS

O GlusterFS não pode ser instalado porque a pré-verificação falha.

Resolvendo o problema

- Se você estiver instalando o GlusterFS durante a instalação do IBM® Cloud Private, poderá ver uma mensagem de erro semelhante à mensagem a seguir:

```
end: '2018-09-18 10 :36:59.331916'
 msg: non-zero return code
 rc: 1
 start: '2018-09-18 10:36:41.873800'
 stderr: |-
   E0918 10:36:43.815489    7463 portforward.go:316] erro ao copiar da conexão local para o
fluxo remoto: read tcp4 127.0.0.1:38630->127.0.0.1:37206: read: conexão reconfigurada pelo peer
   Erro: tarefa com falha: BackoffLimitExceeded
 stderr_lines: <omitted>
 stdout: |-
   A liberação "storage-glusterfs" não existe. Instalando-o agora.
   ============================================================
   O log do Tiller pode ser localizado em cluster/logs/tiller-deploy-646ff69689-ww4tm
   ============================================================
```

  Para identificar a razão para a falha de pré-verificação, conclua as etapas a seguir no nó principal:

  1. Configure o contexto de alias kubectl.

     ```
     alias kc='kubectl --kubeconfig=/etc/cfc/conf/admin.kubeconfig -n kube-system'
     ```

  2. Obtenha o nome do configmap.

     ```
     kc get configmap -l glusterfs-precheck = precheck-results-cm
     ```

     O seguinte é uma saída de amostra:

     ```
     NAME                                              DATA      AGE
     storage-glusterfs-glusterfs-precheck-results-cm   4         10m
     ```

  3. Obtenha os detalhes do configmap.

     ```
     kc describe configmap storage-glusterfs-glusterfs-precheck-results-cm
     ```

O seguinte é uma saída de amostra:

```
Nome:         storage-glusterfs-glusterfs-precheck-results-cm
Namespace:    kube-system
Rótulos:      app=glusterfs
              chart=ibm-glusterfs-99.99.99
              component=precheck-results-cm
              glusterfs-precheck=precheck-results-cm
              heritage=Tiller
              release=storage-glusterfs
Anotações:    description=Configmap de resultados de pré-verificação do GlusterFS
              helm.sh/hook=pre-install
              helm.sh/hook-delete-policy=before-hook-creation
              helm.sh/hook-weight=-5

Dados == ==
10.41.3.19:
----
status : success # msg: validação com êxito
10.41.3.38:
----
status : fail # msg: Os caminhos do dispositivo de armazenamento GlusterFS [udevice-1]
não existem
10.41.4.108:
----
status : success # msg: validação com êxito
precheckJobStatus:
----
status : fail # msg: o nome da classe de armazenamento do GlusterFS glusterf&$s é
inválido. O nome deve consistir em caracteres alfanuméricos minúsculos, - ou ., e deve
iniciar e terminar com um caractere alfanumérico.
Eventos:  <none>
```

É possível identificar a razão da falha nos detalhes do Configmap.

- Se você estiver instalando o gráfico Helm do GlusterFS após a instalação do IBM® Cloud Private, execute estes comandos para identificar a causa da falha. Deve-se configurar a CLI do Kubectl para executar estes comandos. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

  1. Obtenha o nome do configmap.

     ```
     kubectl --namespace kube-system get configmap -l glusterfs-precheck=precheck-results-cm
     ```

  2. Obtenha os detalhes do configmap.

     ```
     kubectl --namespace kube-system describe configmap <configmap-name>
     ```

Dispositivo GlusterFS não localizado após reinicialização do sistema

O dispositivo que você tinha conectado para criar um volume GlusterFS não foi localizado.

Sintomas

GlusterFS se torna instável quando não vê o dispositivo correto como conectado.

Causas

Durante a reinicializações do sistema, os nomes de dispositivo podem mudar.

Resolvendo o problema

Use o link simbólico do dispositivo (symlink) como o identificador de dispositivo.

Recrie seu cluster GlusterFS. Siga as instruções na seção GlusterFS .

Reinstale o IBM® Cloud Private.

Travamentos do nó GlusterFS

Um nó do trabalhador que fazia parte do cluster GlusterFS pode travar.

É possível incluir um novo nó no cluster GlusterFS. Para obter mais informações, consulte Aumentando a capacidade de armazenamento de um cluster GlusterFS.

A reinstalação do IBM Cloud Private não resolve problemas do GlusterFS

A reinstalação do IBM® Cloud Private não resolve problemas do dispositivo GlusterFS.

Identificando o problema

Reúna informações sobre seu ambiente para identificar as razões para os problemas.

1. Execute o comando de instalação com a opção detalhada para que seja possível capturar os logs.
   - Edições Standard:

     ```
     docker run -e LICENSE=accept --net=host -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee install -vvv
     ```

   - Edição da comunidade:

     ```
     docker run -e LICENSE=accept --net=host -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0 install -vvv
     ```

2. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
3. Localize os pods que são relevantes para GlusterFS e Heketi.

   ```
   kubectl -n kube-system get po | grep -E 'gluster|heketi'
   ```

4. Obtenha os logs de cada um dos pods que são relevantes para GlusterFS e Heketi. Identifique os pods GlusterFS e Heketi que estão em execução.

   ```
   kubectl -n kube-system logs glusterfs-<pod name/ID>
   kubectl -n kube-system logs heketi-<pod name/ID> > heketi.txt
   kubectl -n kube-system exec glusterfs-<pod name/ID> -- cat /var/log/glusterfs/glusterd.log
   ```

5. Localize o estado do volume dos pods GlusterFS que estão em execução:

   ```
   kubectl -n kube-system exec glusterfs-<pod name/ID> -- gluster volume status
   ```

6. Localize o estado da topologia dos pods Heketi que estão em execução:

   ```
   kubectl -n kube-system exec heketi-<pod name/ID> -- heketi-cli topology info
   ```

7. Resolva os problemas:
   - Se os logs GlusterFS indicarem um problema com o acesso ao dispositivo, verifique se o caminho no host está correto e se as permissões corretas estão ativadas. Use o comando `ls -l` para confirmar se o arquivo de dispositivo existe e está acessível.
   - Se o dispositivo estiver corrompido ou se o GlusterFS não puder carregar o tijolo, substitua o mapeamento de dispositivo no GlusterFS.
   - Se o status da topologia Heketi mostrar que os volumes de GlusterFS não foram inicializados corretamente, substitua os dispositivos. Para que o Heketi seja inicializado, os dispositivos não devem ser formatados. Siga as instruções em Configurando o GlusterFS durante a instalação do IBM Cloud Private e atualize o arquivo `config.yaml`.
8. Desinstale o IBM Cloud Private. Consulte Desinstalando.
9. Instale o IBM Cloud Private. Consulte Instalando .

Não é possível criar um PersistentVolumeClaim do GlusterFS

Não é possível criar um PersistentVolumeClaim (PVC) do GlusterFS.

Causas

O nome de PVC é muito longo. Não é possível ligar um terminal de serviço PVC se seu nome tem mais de 63 caracteres. Quando GlusterFS cria um nome de terminal de serviço, ele inclui `glusterfs-dynamic` no nome de PVC e esses caracteres extras podem fazer com que o nome de PVC exceda o limite.

Por exemplo, ao criar um PVC em um StatefulSet usando `volumeClaimTemplates`, o PVC que é criado automaticamente é chamado `<pvc name>-<statefulset name>-<ordinal>`. Por exemplo, se o PVC e o StatefulSet são denominados `default-mq-stocktrader-m`, o novo PVC pode ser denominado `default-mq-stocktrader-m-default-mq-stocktrader-m-0`. Se você usar GlusterFS para criar esse PVC, o prefixo `glusterfs-dynamic-` será incluído no nome de PVC para criar um terminal em serviço. O nome do terminal em serviço `glusterfs-dynamic-default-mq-stocktrader-m-default-mq-stocktrader-m-0` excede 63 caracteres e a ligação de PVC falha.

Ao criar o PVC, você poderá ver `Status: Pending`. Também poderá ver uma mensagem semelhante à mensagem a seguir na seção `Events:` da saída de comando.

```
Service "glusterfs-dynamic-default-mq-stocktrader-m-default-mq-stocktrader-m-0" is invalid: \
metadata.name: Invalid value: \
"glusterfs-dynamic-default-mq-stocktrader-m-default-mq-stocktrader-m-0": \
must be no more than 63 characters
```

Resolvendo o problema

Se você usar GlusterFS, limite seu nome de PVC a 45 caracteres. Se você usar `volumeClaimTemplates` em StatefulSets, use nomes abreviados para o nome de StatefulSet e o nome de PVC.

Se sua ligação de PVC falhar, reduza o comprimento do nome do StatefulSet ou do nome do PVC para que o comprimento total do terminal em serviço GlusterFS, `glusterfs-dynamic-<pvc name>-<statefulset name>-<ordinal>`, não exceda 63 caracteres.

Para obter mais informações sobre esse problema, consulte o problema glusterfs create pvc falhou na comunidade do Kubernetes.

A reinicialização simultânea de nós do trabalhador faz com que o GlusterFS falhe

Quando você reinicia todos os nós do trabalhador no mesmo tempo, o GlusterFS não inicia.

Causas

Devido a uma reinicialização simultânea dos nós do trabalhador, o pod Heketi não inicia. O contêiner Heketi falha ao iniciar pois é impossível montar volumes heketidbstorage. O status de heketidbstorage é exibido como off-line porque os tijolos correspondentes não estão on-line devido a um encerramento não limpo.

Resolvendo o problema

Obtenha as informações do pod do GlusterFS executando o comando a seguir:

```
kubectl -n kube-system get pod | grep gluster
```

A seguir há um exemplo da saída de comando:

```
glusterfs-36nd0 1/1 Running 4 7d
glusterfs-3m5ql 1/1 Running 3 7d
glusterfs-tc279 1/1 Running 16 7d
```

Conclua as etapas a seguir para todos os pods do GlusterFS:

1. Efetue login no pod do GlusterFS:

   ```
   kubectl -n kube-system exec -it <POD ID> bash
   ```

   A seguir está um exemplo do comando e de sua saída:

```
root@BPILICPMSTR001:~/cluster# kubectl -n kube-system exec -it glusterfs-36nd0 bash
[root@bpilicpwrk001 /]#
```

2. Verifique o status do volume do GlusterFS no pod:

   ```
   gluster volume status
   ```

   A seguir está um exemplo do comando e de sua saída:

   ```
   [root@bpilicpwrk001 /]# gluster volume status
   Status of volume: heketidbstorage
   Gluster process TCP Port RDMA Port Online Pid

   Brick 10.10.25.49:/var/lib/heketi/mounts/vg
   _22bbf0fbb483f9c170774d83081c3420/brick_2fb
   3a10c7eafb8bed375829e8aaf782a/brick 49153 0 Y 5858
   Brick 10.10.25.51:/var/lib/heketi/mounts/vg
   _118f22bc13626321606280ea1d79fdc3/brick_649
   4a3b077c38667f07a59197efabea7/brick 49153 0 Y 5318
   Brick 10.10.25.50:/var/lib/heketi/mounts/vg
   _d4d4f2e86c08f571befe7fc272dc4aae/brick_dc9
   416bf4d88e45ff4d0061c08ef5b19/brick 49153 0 Y 5441
   Self-heal Daemon on localhost N/A N/A Y 5878
   Self-heal Daemon on 10.10.25.50 N/A N/A Y 5461
   Self-heal Daemon on 10.10.25.51 N/A N/A Y 5338
   Task Status of Volume heketidbstorage

   There are no active volume tasks

   [root@bpilicpwrk001 /]#
   ```

   Se os tijolos correspondentes ao heketidbstorage estiverem inativos, reinicie os tijolos executando os comandos a seguir:

   ```
   gluster volume stop heketidbstorage

   gluster volume start heketidbstorage force
   ```

3. Verifique o status do pod Heketi:

   ```
   kubectl -n kube-system get pod | grep heketi
   ```

   O status exibe uma mensagem semelhante à mensagem a seguir:

   ```
   heketi-402978595-pjnd7 1/1 Running 0 2h
   ```

Não é possível criar ou excluir um volume persistente ou solicitação de volume persistente

Não é possível criar ou excluir um PersistentVolume (PV) ou PersistentVolumeClaim (PVC).

Se estiver usando o armazenamento GlusterFS, quando tentar criar ou excluir um PV ou PVC, você poderá ver a mensagem `database is in read-only mode`.

Causas

- Se você reiniciou o pod do Heketi, o pod não obteve acesso `write` ao arquivo de banco de dados do Heketi.
- Um nó do cluster GlusterFS está inativo.

Resolvendo o problema

Conclua estas etapas para resolver o problema:

1. Assegure-se de que todos os nós do cluster do GlusterFS estejam ativos e em execução.
2. Assegure-se de que a CLI `kubectl` esteja configurada. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

3. Diminua a implementação do Heketi.

   a. Identifique o nome da implementação do Heketi executando o comando a seguir:

```
kubectl -- namespace = kube-system get deployments -l glusterfs=heketi-deployment
```

A maioria das configurações está executando zero ou uma implementação do Heketi. Use o nome da implementação que é retornado para as etapas restantes que requerem que você especifique o nome da implementação.

b. Escale a implementação do Heketi para 0 inserindo o comando a seguir:

```
kubectl scale --namespace=kube-system deploy -l glusterfs=heketi-deployment --replicas=0
deployment.extensions "heketi-deployment" scaled
```

c. Aguarde o pod para finalizar. Execute o comando a seguir para visualizar o status do pod:

```
kubectl -- namespace = kube-system get pods -l glusterfs=heketi-pod
```

Se o pod é finalizado com êxito, o comando não retorna saída.

4. Aumente a implementação do Heketi.

a. Aumente o número de instâncias do Heketi inserindo o comando a seguir:

```
kubectl scale --namespace=kube-system deploy -l glusterfs=heketi-deployment --replicas=1
```

b. Valide o aumento da implementação inserindo o comando a seguir:

```
kubectl --namespace=kube-system rollout status deployments [name-of-your-deployment]
```

Após a implementação ser apresentada com sucesso, a saída é semelhante ao código a seguir:

```
implementação "heketi" apresentada com sucesso
```

c. Aguarde o pod para iniciar. Execute o comando a seguir para visualizar o status do pod:

```
kubectl -- namespace = kube-system get pods -l glusterfs=heketi-pod
```

Depois que o pod é iniciado, a saída é semelhante ao código a seguir:

```
heketi-68549fdf65-sm8tl                                    1/1       Running
0          23s
```

O status do nó GlusterFS é mostrado como peer rejeitado

O status do nó GlusterFS pode ser mostrado como "Peer rejeitado".

Nota: se múltiplos nós estiverem em um estado "Peer rejeitado", você poderá não ser capaz de recuperar os nós do GlusterFS. Você pode precisar configurar o cluster do GlusterFS novamente.

Resolvendo o problema

1. Assegure-se de que a CLI `kubectl` esteja configurada. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Verifique o status de pods GlusterFS.

```
kubectl -n kube-system get pods -o wide | grep gluster
```

A saída se assemelha ao código a seguir:

```
glusterfs-l95bp        1/1       Running     3          6d        192.168.0.184   worker-2
glusterfs-n85vr        1/1       Running     0          4d        192.168.0.56    worker-1
glusterfs-p66jq        1/1       Running     3          6d        192.168.0.96    worker-3
```

3. Verifique o status do peer de qualquer pod do GlusterFS.

```
kubectl -n kube-system exec <pod name> -- gluster peer status
```

O seguinte é um exemplo de comando e de saída:

```
kubectl -n kube-system exec glusterfs-l95bp -- gluster peer status
Number of Peers: 2

Hostname: 192.168.0.96
```

```
Uuid: 3e518a43-b59f-45e7-a62c-5b213e0fece8
State: Peer in Cluster (Connected)
Other names:
192.168.0.96

Hostname: worker-1
Uuid: c2a81937-e94f-4a22-86e7-bc9c8929c2d3
State: Peer Rejected (Connected)
```

Se você vir um status "Peer rejeitado", isso indica que a configuração do volume nesse peer está fora de sincronização com o resto do cluster.

Para sincronizar com o cluster do GlusterFS, conclua estas etapas no nó do GlusterFS que está no estado "Peer rejeitado":

1. Acesse o shell dentro do pod do GlusterFS:

   ```
   kubectl -n kube-system exec -it <pod name> bash
   ```

2. Mude para o `/var/lib/glusterd` diretório.

   ```
   Cd /var/lib/glusterd
   ```

   A pasta pode conter os arquivos e pastas a seguir:

   ```
   . .. bitd geo-replication glusterd.info glusterfind glustershd groups hooks nfs
   options peers quotad scrub snaps ss_brick vols
   ```

3. Exclua tudo, exceto `glusterd.info`, que é o arquivo de identificador exclusivo universal (UUID).

4. Reinicie o daemon do Gluster.

   ```
   Reinicie glusterd de serviço
   ```

5. Analise um peer que não esteja em um estado "Peer rejeitado".

   ```
   gluster peer probe <node name>
   ```

6. Verifique o status de peer.

   ```
   Gluster peer status
   ```

Note: pode ser necessário repetir as etapas até você não veja mais o status "Peer rejeitado".

A exclusão de uma solicitação de volume persistente do GlusterFS pode mostrar o status do volume persistente como com falha

Um status de PersistentVolume (PV) do GlusterFS é mostrado como "Com falha" quando você exclui o PersistentVolumeClaim (PVC) que está ligado a ele.

Veja os comandos e a saída de exemplo a seguir:

Antes de continuar, configure a CLI `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

1. Obtenha uma lista de todos os PVCs.

   ```
   Kubectl get pvc
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                    STATUS    VOLUME                                     CAPACITY   ACCESS
   MODES   STORAGECLASS          AGE
   gfs-endpoint-test       Bound     pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO
   gluster              22h
   noreplica-pvc-gluster   Bound     pvc-08f87fa7-542f-11e8-89ac-005056a8640c   1Gi        RWO
   gluster-no-replica   5d
   test-pvc-gluster-r2-1   Bound     pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO
   gluster-replica2     5d
   ```

2. Excluir um PVC.

   ```
   kubectl delete pvc <PVC name>
   ```

O seguinte é um exemplo de comando e de saída:

```
kubectl delete pvc noreplica-pvc-gluster
persistentvolumeclaim "my-release-grafana" deleted
```

3. Verifique o status do PVCs.

```
Kubectl get pvc
```

A saída é semelhante ao código a seguir. O PVC é excluído com êxito.

```
NAME                    STATUS    VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS          AGE
gfs-endpoint-test       Bound     pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO
gluster              22h
test-pvc-gluster-r2-1   Bound     pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO
gluster-replica2     5d
```

4. Verifique o status do VFs. O status do PV ao qual o PVC excluído foi ligado pode ser mostrado como "Com falha".

```
kubectl get pv
```

A saída se assemelha ao código a seguir:

```
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
CLAIM                                       STORAGECLASS               REASON    AGE
pvc-08f87fa7-542f-11e8-89ac-005056a8640c   1Gi        RWO            Delete           Failed
default/noreplica-pvc-gluster               gluster-no-replica                  5d
pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO            Delete           Bound
default/test-pvc-gluster-r2-1               gluster-replica2                    5d
pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO            Delete           Bound
default/gfs-endpoint-test                   gluster                             22h
```

Causas

Um nó do GlusterFS está inativo ou nem todos os pods do GlusterFS estão em um estado de execução.

Resolvendo o problema

1. Verifique se todos os nós do GlusterFS estão em execução e são registrados com o cluster do IBM Cloud Private.

```
kubectl get nodes
```

A saída se assemelha ao código a seguir:

```
NAME       STATUS    ROLES     AGE       VERSION
master     Ready     <none>    5d        v1.11.0+icp-ee
worker-1   Ready     <none>    5d        v1.11.0+icp-ee
worker-2   Ready     <none>    5d        v1.11.0+icp-ee
worker-3   Ready     <none>    5d        v1.11.0+icp-ee
```

2. Assegure-se de que o status de todos os pods do GlusterFS seja mostrado como "Em execução". Além disso, verifique se o número de pods é o mesmo que o número de nós do GlusterFS que você configurou.

```
kubectl -n kube-system get pods | grep gluster
```

A saída se assemelha ao código a seguir:

```
glusterfs-l95bp                                                1/1       Running     1
5d
glusterfs-n85vr                                                1/1       Running     0
3d
glusterfs-p66jq                                                1/1       Running     3
5d
```

Se a contagem de nós e pods não corresponder, verifique os rótulos dos nós. Todos os nós do GlusterFS devem ter o rótulo `storagenode=glusterfs`.

```
kubectl get nodes --show-labels
```

A saída se assemelha ao código a seguir:

```
NAME                  STATUS    ROLES     AGE       VERSION          LABELS
master     Ready     <none>    6d        v1.11.0+icp-ee
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,etcd=true,gpu/nvidia=NA,kubernetes.io
/hostname=master,management=true,master=true,proxy=true,role=master
worker-1   Ready     <none>    6d        v1.11.0+icp-ee
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
worker-1,storagenode=glusterfs
worker-2   Ready     <none>    6d        v1.11.0+icp-ee
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
worker-2,storagenode=glusterfs
worker-3   Ready     <none>    6d        v1.11.0+icp-ee
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
worker-3,storagenode=glusterfs
```

3. Para todos os nós do GlusterFS que não possuem um rótulo `storagenode=glusterfs`, inclua o rótulo.

   ```
   Kubectl label node < nós name > storagenode=glusterfs
   ```

4. Verifique o status dos pods do GlusterFS e Heketi.

   ```
   kubectl -n kube-system get pods | egrep "gluster|heketi"
   ```

   A saída se assemelha ao código a seguir:

   ```
   glusterfs-l95bp               1/1       Running     1          5d
   glusterfs-n85vr               1/1       Running     0          3d
   glusterfs-p66jq               1/1       Running     3          5d
   heketi-9f984d759-zcmvb        1/1       Running     0          4d
   ```

5. Quando todos os pods mostrarem o status como "Em execução", verifique o status do PV que foi mostrado como "Com calha". **Nota:** o PV será excluído se a política de recuperação foi "Excluir". Além disso, o status do PV é mostrado como "Liberado".

   ```
   kubectl get pv
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
   CLAIM                                       STORAGECLASS               REASON    AGE
   pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO            Delete           Bound
   default/test-pvc-gluster-r2-1               gluster-replica2                    5d
   pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO            Delete           Bound
   default/gfs-endpoint-test                   gluster                             22h
   ```

O pod do GlusterFS não está planejado após a reinicialização de um nó

Ao reiniciar um nó GlusterFS, o pod GlusterFS não é planejado.

Causas

O rótulo `storagenode=glusterfs` é perdido durante uma reinicialização.

Resolvendo o problema

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Rotule o nó.

   ```
   kubectl label nodes <node IP address> storagenode=glusterfs
   ```

3. Atualize o arquivo de manifesto do serviço do kubelet para tornar a etiqueta persistente entre as reinicializações do sistema.
   1. Abra o arquivo `/etc/systemd/system/kubelet.service` para edição.
   2. Inclua a seguinte parte do código na seção `[Service]`:

      ```
      --node-labels=storagenode=glusterfs\
      ```

      Nota: se outros rótulos forem listados, retenha-os. Inclua o rótulo `storagenode=glusterfs` na lista de rótulos. Por exemplo: `--node-labels=disktype=ssd, foo=bar, storagenode=glusterfs\`.

Após incluir o código, o conteúdo do arquivo é semelhante ao texto a seguir:

```
[Service] EnvironmentFile=-/etc/environment ExecStart=/opt/kubernetes/hyperkube kubelet \
...
--node-labels=storagenode=glusterfs\
...
```

4. Verifique o status do pod.

   ```
   kubectl -n kube-system get po -owide | grep -E "gluster|heketi"
   ```

   Se o status for mostrado como `ContainerCreating`, exclua o pod. Quando o pod é recriado, ele é exibido em um estado `Em execução`.

Incompatibilidade do uso do disco Heketi

Incompatibilidade do espaço em disco real e do espaço em disco usado que é relatado por Heketi.

Ao criar uma solicitação de volume persistente (PVC), é possível ver uma mensagem de erro de que nenhum espaço em disco está disponível.

Resolvendo o problema

Se você vir uma incompatibilidade de uso do disco nas informações de topologia do Heketi, execute estes comandos para sincronizar o uso do disco.

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Obtenha o nome do pod de implementação Heketi.

   ```
   kubectl -n kube-system get pods | grep heketi
   ```

   A seguir está uma saída de exemplo do comando:

   ```
   storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf   1/1       Running     0
   18h
   ```

3. Efetue login no pod Heketi e obtenha a lista de nós.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   node list
   ```

   A seguir está uma saída de exemplo do comando:

   ```
   Id:10cc13f4cd18136fc9b6d2a6d1eac733     Cluster:34fe224a09d8022a3582da817c31a81b
   Id:157012bb7b0391b2c116b35de8d6e7ba     Cluster:34fe224a09d8022a3582da817c31a81b
   Id:95c72d452bc2c21b3a1e2fa87e63f902     Cluster:34fe224a09d8022a3582da817c31a81b
   ```

4. Obtenha informações sobre os dispositivos nos nós. Repita esta etapa para cada nó.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   node info <node-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin node info 10cc13f4cd18136fc9b6d2a6d1eac733
   Node Id: 10cc13f4cd18136fc9b6d2a6d1eac733
   State: online
   Cluster Id: 34fe224a09d8022a3582da817c31a81b
   Zone: 1
   Management Hostname: 10.41.4.108
   Storage Hostname: 10.41.4.108
   Devices:
   Id:c7e93cd3d3d293a78dfb99cb2809f699   Name:/dev/disk/by-path/virtio-pci-0000_00_11_0
   State:online    Size (GiB):699      Used (GiB):497       Free (GiB):202
             Bricks:
                       Id:33cb391e0113ec96ccc546a4e4288018   Size (GiB):100     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_33cb391e0113ec96ccc546a4e42880
   18/brick
                       Id:41a508db035b042b1d35839f0d40f0c5   Size (GiB):20     Path:
   ```

```
/var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_41a508db035b042b1d35839f0d40f0
c5/brick
```

5. Obtenha informações sobre o uso do disco nos dispositivos. Repita esta etapa para cada dispositivo em cada nó.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   device info <device-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin device info c7e93cd3d3d293a78dfb99cb2809f699
   Device Id: c7e93cd3d3d293a78dfb99cb2809f699
   Name: /dev/disk/by-path/virtio-pci-0000_00_11_0
   State: online
   Size (GiB): 699
   Used (GiB): 497
   Free (GiB): 202
   Bricks:
   Id:33cb391e0113ec96ccc546a4e4288018   Size (GiB):100     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_33cb391e0113ec96ccc546a4e42880
   18/brick
   Id:41a508db035b042b1d35839f0d40f0c5   Size (GiB):20     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_41a508db035b042b1d35839f0d40f0
   c5/brick
   ```

6. Sincronize o dispositivo para refletir o uso real do disco. Repita esta etapa para cada dispositivo em cada nó.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   device resync <device-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin device resync c7e93cd3d3d293a78dfb99cb2809f699
   Device  updated
   ```

7. Verifique se as informações de uso do disco estão sincronizadas.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   device info <device-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin device info c7e93cd3d3d293a78dfb99cb2809f699
   Device Id: c7e93cd3d3d293a78dfb99cb2809f699
   Name: /dev/disk/by-path/virtio-pci-0000_00_11_0
   State: online
   Size (GiB): 699
   Used (GiB): 120
   Free (GiB): 579
   Bricks:
   Id:33cb391e0113ec96ccc546a4e4288018   Size (GiB):100     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_33cb391e0113ec96ccc546a4e42880
   18/brick
   Id:41a508db035b042b1d35839f0d40f0c5   Size (GiB):20     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_41a508db035b042b1d35839f0d40f0
   c5/brick
   ```

O Ceph bloqueia o armazenamento de bloco usando o Rook

O Rook é um orquestrador de software-livre para sistemas de armazenamento distribuídos que é executado em ambientes nativos de nuvem.

Para obter mais informações sobre o Rook, consulte https://rook.io/ ↗ e https://github.com/rook/rook ↗.

O Ceph é um sistema de armazenamento distribuído com várias apresentações de armazenamento que incluem armazenamento de objetos, armazenamento de bloco e sistema de arquivos compartilhados compatível com POSIX.

O Rook funciona como um operador e fornece a plataforma, a estrutura e o suporte para soluções de armazenamento para integração nativa com ambientes em nuvem nativa. O IBM® Cloud Private depende do Operador Rook para fornecer

armazenamento do Ceph para a carga de trabalho do aplicativo. O Rook está agora no estado beta.

Cenários de Implementação

É possível usar nós do trabalhador dedicados para implementar o cluster Rook Ceph ou é possível usar nós do trabalhador que são compartilhados com as cargas de trabalho do aplicativo para implementar um cluster do Rook Ceph.

Para implementar o cluster do Rook Ceph em nós do trabalhador dedicados, siga estas etapas:

1. Inclua um grupo de hosts customizados em seu cluster.
2. Especifique os termos e as tolerâncias do NodeSelector no parâmetro `placement` no gráfico RookCeph Helm.

Para obter mais informações, consulte Criando um grupo de hosts customizados.

Pré-requisitos e Limitações

Para obter mais informações, consulte Pré-requisitos e limitações.

Implementando o cluster

É possível implementar o Rook Ceph instalando o gráfico Helm que está disponível no catálogo do IBM® Cloud Private. A implementação do cluster do Rook Ceph, juntamente com a instalação do IBM Cloud Private, não é suportada.

Opções de configuração

A implementação de um cluster do Rook Ceph é um processo de três etapas:

1. Configure o controle de acesso baseado na função (RBAC).
2. Instale o gráfico Helm do operador Rook.
3. Instale o gráfico de cluster de armazenamento do Ceph.

Para obter mais informações, consulte Instalando o gráfico Rook Helm.

Para obter instruções de instalação que estão no arquivo leia-me do gráfico Helm, consulte Instalando o cluster do Rook Ceph.

Criando uma Classe de Armazenamento

Ao implementar o cluster do Rook Ceph, por padrão, uma classe de armazenamento é incluída para os aplicativos para provisionar o volume dinâmico.

Verificando a instalação

Após a instalação bem-sucedida do cluster do Rook Ceph, verifique se todos os pods estão ativos e se o cluster está utilizável.

É necessário verificar todos os pods a seguir no mesmo namespace em que você implementou o gráfico:

- Verifique se há muitos pods de monitoramento (`ceph-mon`) conforme especificado no parâmetro de configuração `cluster.monCount`.
- Verifique se há muitos pods `ceph-osd` como o número de nós de armazenamento que são especificados no parâmetro de configuração `cluster.storageNodes`.
- Verifique se os pods `api` e `ceph-mgr` estão ativos.

A integridade do cluster pode ser executada implementando um aplicativo a partir do catálogo do IBM Cloud Private que requer armazenamento persistente. Por exemplo, consulte PostgreSQL.

Gerenciando seu cluster

As operações a seguir poderão ser executadas após o cluster ser instalado.

Provisionando volume persistente

Para obter mais informações, consulte Provisionando volume persistente.

Atualizando

O Rook Ceph no IBM Cloud Private depende da versão Beta do Rook da comunidade. Atualmente, esse gráfico da comunidade não suporta o upgrade do cluster do Rook Ceph existente.

Desinstalação

Para obter mais informações, consulte Desinstalando o gráfico.

Problemas Conhecidos

Para obter uma lista de problemas conhecidos comuns, consulte Problemas comuns.

Resolução de problemas

Para obter mais informações, consulte Resolução de problemas do cluster Rook Ceph.

- Pré-requisitos e limitações
- Configurando o Rook

Pré-requisitos e Limitações

Pré-requisitos e limitações para instalação do gráfico Rook Ceph em seu cluster do IBM® Cloud Private.

Pré-requisitos

- Deve-se ser um administrador de cluster para instalar o gráfico.
- O controle de acesso baseado na função (RBAC) é ativado por padrão no IBM Cloud Private. Portanto, deve-se incluir determinados objetos RBAC antes de implementar os gráficos de operador Rook e de cluster Rook Ceph. Para obter mais informações, consulte o arquivo Leia-me do gráfico.
- Deve-se primeiramente implementar o gráfico do operador Rook em seu cluster do IBM Cloud Private. Essa implementação deve criar um pod do operador Rook em seu cluster e um pod do agente Rook em cada nó.
- No parâmetro `storage.nodes`, deve-se especificar discos ou diretórios para um nó de armazenamento. Seu nó de armazenamento deve fazer parte de seu cluster do IBM Cloud Private. Se você especificar dispositivos de disco, nenhum sistema de arquivos deverá estar presente nos dispositivos.
- O caminho que você especifica no parâmetro `dataDirHostPath` não deve ter nenhuma entrada pré-existente de uma instalação de cluster anterior. Chaves antigas e outras configurações que existem a partir de uma instalação anterior podem fazer com que a instalação falhe.

Limitações

- O Rook é suportado apenas em Linux® x86_64 clusters. Atualmente, ele não é suportado nos clusters Linux® on Power® (ppc64le) e IBM® Z.
- O Rook suporta múltiplos clusters do Ceph. No entanto, é possível configurar apenas um cluster por namespace.
- O cluster Rook Ceph é suportado no kernel Linux versão 3.15 ou mais recente.
- É possível instalar os gráficos do Rook Operator e do Rook Cluster no mesmo namespace ou em namespaces diferentes. Se você instalar em namespaces diferentes, assegure-se de fornecer privilégio para a conta de serviço padrão do namespace `rook-cluster`, para que o serviço possa obter e listar pods de outros namespaces em seu cluster.

Versões do gráfico

O IBM Cloud Private suporta a versão Beta do gráfico da comunidade do Rook Operator e a Versão 0.8.3 do gráfico de cluster do IBM Rook Ceph.

Configurando o Rook

Configure o Rook em seu cluster do IBM® Cloud Private .

É possível usar um grupo de hosts customizados ou nós do trabalhador IBM Cloud Private para configurar o cluster de armazenamento Rook.

- Criando um grupo de hosts customizados

- Instalando o gráfico Rook Helm

Criando um Grupo de Host Customizado

Crie um grupo de hosts dedicados para instalar o Rook em seu cluster do IBM® Cloud Private.

É possível configurar um grupo de hosts customizados com os nós de armazenamento Rook dedicados. Para obter mais informações sobre como incluir um grupo de hosts, consulte Definindo grupos de hosts customizados.

A seguir está uma configuração de exemplo de um grupo de hosts com nós de armazenamento do Rook dedicados. Inclua essa configuração no arquivo `/<installation_directory>/cluster/hosts`.

```
[ master ] 10.41.1.182

[ trabalhador ] 10.41.2.67

[ proxy ] 10.41.1.182

[hostgroup-cephfs]
10.41.3.214
10.41.4.100
10.41.4.202
```

Depois que seu cluster do IBM Cloud Private é instalado, os nós de armazenamento Rook dedicados são designados a funções e contaminações. A seguir está um exemplo de uma função e de uma contaminação que são designadas a um nó Rook dedicado:

```
kubectl describe node 10.41.3.214
Name:               10.41.3.214
Roles:              cephfs
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    cephfs=true
                    hostgroup=cephfs
                    kubernetes.io/hostname=10.41.3.214
                    node-role.kubernetes.io/cephfs=true
Annotations:        node.alpha.kubernetes.io/ttl=0
                    volumes.kubernetes.io/controller-managed-attach-detach=true
CreationTimestamp:  Tue, 09 Oct 2018 06:57:48 -0700
Taints:             dedicated=cephfs:NoSchedule
```

Como instalar o gráfico Rook Helm

Instale o gráfico Helm do cluster Rook Ceph após ter instalado seu cluster do IBM® Cloud Private.

Antes de instalar o gráfico, especifique a configuração e a tolerância de posicionamento para o cluster para que os pods sejam planejados nos nós Rook dedicados. É possível especificar a configuração de localização para os serviços `mgr`, `mon`, `osd` e `all`. A configuração é salva no arquivo `values.yaml` para planejamento dos serviços no grupo de hosts dedicados. A seguir está um arquivo `values.yaml` de amostra com a configuração e a tolerância de posicionamento:

```
##############################################################################
# Licensed Materials - Property of IBM
# 5737-E67
# (C) Copyright IBM Corporation 2016, 2018 All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
##############################################################################

# Default values for ibm-rook-rbd-cluster.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

arch:
  amd64: "2 - No preference"

rookOperatorNamespace: "default"

cluster:
  #  The path on the host (hostPath) where config and data should be stored for each of the
services.
```

```
  dataDirHostPath: /var/lib/rook
  mon:
    #set the number of mons to be started. The number should be odd and between 1 and 9
    count: 3
    # enable (true) or disable (false) the placement of multiple mons on one node.
    allowMultiplePerNode: true
  network:
    hostNetwork: false
  dashboard:
    enabled: true
  placement:
    all:
      enabled: true
      nodeSelectorTerms:
      - matchExpressions:
        - key: hostgroup
          operator: In
          values:
          - cephfs
      tolerations:
      - key: dedicated
        operator: Equal
        value: cephfs
        effect: NoSchedule

  storage:
    useAllNodes: false
    useAllDevices: false
    deviceFilter: ""
    local: ""
    config:
      # filestore or bluestore, the underlying storage format to use for each OSD.
      storeType: bluestore
      # this value can be removed for environments with normal sized disks (100 GB or larger)
      databaseSizeMB: "1024"
      # this value can be removed for environments with normal sized disks (20 GB or larger)
      journalSizeMB: "1024"
   # Individual nodes and their config can be specified as well, but 'useAllNodes' must
   # be set to false. Then, only the named nodes at the end of this example will be used as storage
resources.
   # Each node's 'name' field should match their 'kubernetes.io/hostname' label.
   #
   #
    nodes:
      - name: "10.41.4.202"
        devices:
          - name: "vdb"
      - name: "10.41.3.214"
        devices:
          - name: "vdb"
      - name: "10.41.4.100"
        devices:
          - name: "vdb"
```

Em seguida, instale o gráfico Helm a partir da console de gerenciamento. Para obter mais informações, consulte Cluster Rook Ceph.

Resolução de problemas do cluster do Rook Ceph

Revise frequentemente os problemas de cluster do Rook Ceph encontrados.

- A instalação do gráfico do Rook Operator usando a console de gerenciamento obtém o erro ESOCKETTIMEDOUT
- A exclusão do gráfico do Rook Operator não exclui o daemonset do Rook
- A instalação do gráfico de cluster do Rook (ibm-rook-rbd-cluster) usando a console de gerenciamento obtém o erro ESOCKETTIMEDOUT
- A instalação do gráfico de cluster do Rook (ibm-rook-rbd-cluster) obtém a falha da tarefa: Erro de BackoffLimitExceeded
    - O gráfico Helm do Rook Operator não está instalado em seu cluster
    - O gráfico de cluster do Rook já está instalado em seu namespace
- O gráfico de cluster do Rook inicia a implementação, mas o rook-ceph-mon obtém um erro CrashLoopBackOff

- A implementação é concluída para o gráfico ibm-rook-rbd-cluster, mas nenhum pod rook-mon, rook-ceph, manager ou api aparece
- Depois que um nó do trabalhador é reiniciado, o pod do agente do Rook permanece no status de erro

Antes de continuar com a resolução de problemas, assegure-se de que seu cluster atenda aos pré-requisitos e que você tenha permissões adequadas para executar operações relacionadas à instalação. Para obter mais informações, consulte Pré-requisitos e limitações.

Nota: é necessário configurar a CLI do kubectl para executar comandos de resolução de problemas. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

A instalação do gráfico do Rook Operator usando a console de gerenciamento obtém

o erro ESOCKETTIMEDOUT

É possível ver um erro `ESOCKETTIMEDOUT` enquanto instala o gráfico do Rook Operator usando a console de gerenciamento:

Resolva o problema

Verifique se os pods rook-agent e do operador estão em execução. Execute os comandos a seguir:

```
kubectl get nodes -o wide
```

```
kubectl -n default get po -o wide
```

Se você tiver tantos agentes quanto o número de nós do trabalhador e se um pod do operador estiver em execução, a instalação foi bem-sucedida. É possível ignorar o erro.

Se os pods do agente ou do operador não estiverem em execução, verifique os logs do Helm para identificar o erro:

```
kubectl -n kube-system get po | grep helm
```

O seguinte é uma saída de amostra:

```
helm-api-66b98d88bc-6psq6                          2/2       Running    0          1d
helm-repo-5495f5c48c-k9mkl                         1/1       Running    0          1d
```

```
kubectl -n kube-system log helm-api-66b98d88bc-6psq6 rudder
```

A exclusão do gráfico do Rook Operator não exclui o daemonset do Rook

1. Obtenha a lista de pods.

   ```
   kubectl -n default get po
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME               READY     STATUS    RESTARTS   AGE
   rook-agent-ckxht   1/1       Running   0          23h
   rook-agent-jnxh6   1/1       Running   0          23h
   rook-agent-wkt26   1/1       Running   0          23h
   ```

2. Obtenha o daemonset.

   ```
   kubectl -n default get ds
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME         DESIRED   CURRENT   READY     UP-TO-DATE   AVAILABLE   NODE-SELECTOR   AGE
   rook-agent   3         3         3         3            3           <none>          23h
   ```

Esse problema é conhecido na liberação alfa do Rook. Exclua manualmente o daemonset do agente. Execute os comandos a seguir:

1. Exclua o daemonset.

   ```
   kubectl -n default delete ds rook-agent
   ```

   O seguinte é uma saída de amostra:

```
daemonset "rook-agent" deleted
```

2. Obtenha uma lista de pods.

   ```
   kubectl -n default get po
   ```

   O seguinte é uma saída de amostra:

   ```
   Nenhum recurso localizado.
   ```

A instalação do gráfico de cluster Rook (ibm-rook-rbd-cluster) usando a

console de gerenciamento obtém o erro ESOCKETTIMEDOUT

É possível ver um erro `ESOCKETTIMEDOUT` enquanto instala o gráfico de cluster Rook usando a console de gerenciamento.

1. Verifique os pods que estão em execução no namespace no qual você está instalando o gráfico. Procure por `rook-cluster-precheck-job` e seu InitContainer. Talvez você veja o erro a seguir:

   ```
   kubectl -n default get po
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME                                READY     STATUS              RESTARTS   AGE
   rook-cluster-precheck-job-mqfv9     0/1       Init:ErrImagePull   0          28s
   ```

2. Verifique se o repositório do Docker que está especificado para a imagem do Hyperkube está correto.

A instalação do gráfico de cluster Rook (ibm-rook-rbd-cluster) obtém a falha da tarefa:

Erro de BackoffLimitExceeded

É possível ver um erro `BackoffLimitExceeded` enquanto instala o gráfico de cluster Rook.

As duas razões a seguir podem estar causando este erro:

- O gráfico Helm do Rook Operator não está instalado em seu cluster.
- O gráfico de cluster Rook já está instalado em seu namespace.

O gráfico Helm do Rook Operator não está instalado em seu cluster

Verifique se o gráfico Helm do Rook Operator está instalado em seu cluster:

```
kubectl get po --all-namespaces | grep rook-operator
```

Se o gráfico não estiver instalado, instale-o primeiramente.

Para obter mais informações sobre como instalar o gráfico Helm do Rook Operator, consulte Gráfico Helm do Ceph Operator.

O gráfico de cluster do Rook já está instalado em seu namespace

Verifique se o gráfico de cluster Rook está instalado em seu cluster:

```
kubectl -n default get cluster
```

O seguinte é uma saída de amostra:

```
NAME              KIND
default-cluster   Cluster.v1alpha1.rook.io
```

Não é possível instalar múltiplos gráficos de cluster Rook em um namespace.

O gráfico de cluster Rook inicia a implementação, mas o rook-ceph-mon obtém um erro

CrashLoopBackOff

1. Obtenha uma lista de pods.

   ```
   kubectl -n default get po
   ```

O seguinte é uma saída de amostra:

```
NAME                            READY     STATUS             RESTARTS   AGE
rook-agent-8tlqf                1/1       Running            0          24m
rook-agent-htjdl                1/1       Running            0          24m
rook-agent-q46vw                1/1       Running            0          24m
rook-ceph-mon0-f2bc6            0/1       CrashLoopBackOff   2          37s
rook-operator-947bf78c6-8hjgj   1/1       Running            0          24m
```

2. Verifique o log `rook-ceph-mon`.

```
kubectl -n default log rook-ceph-mon0-f2bc6
```

O seguinte é uma saída de amostra:

```
2018-05-18 10:51:39.932606 I | rook: starting Rook v0.7.1 with arguments '/usr/local/bin/rook
mon --config-dir=/var/lib/rook --name=rook-ceph-mon0 --port=6790 --fsid=a01d92fb-8191-4343-
8ec1-676abd0de780'
2018-05-18 10:51:39.932749 I | rook: flag values: --admin-secret=*****, --ceph-config-
override=/etc/rook/config/override.conf, --cluster-name=default, --config-dir=/var/lib/rook, --
fsid=a01d92fb-8191-4343-8ec1-676abd0de780, --help=false, --log-level=INFO, --mon-
endpoints=rook-ceph-mon0=10.0.0.185:6790, --mon-secret=*****, --name=rook-ceph-mon0, --
port=6790, --private-ipv4=10.1.19.21, --public-ipv4=10.0.0.185
The keyring does not match the existing keyring in /var/lib/rook/rook-ceph-mon0/data/keyring.
Pode ser
necessário excluir o conteúdo de dataDirHostPath no host a partir de uma implementação
anterior.
```

Esse erro indica que você tinha uma implementação do Rook Ceph anterior.

Para corrigir esse problema, exclua o gráfico com falha e, em seguida, exclua o conteúdo do arquivo `dataDirHostPath` nos hosts que foram usados em uma implementação anterior. Ou especifique uma configuração `dataDirHostPath` diferente. Em seguida, reinstale o gráfico `ibm-rook-rbd-cluster`.

A implementação é concluída para o gráfico ibm-rook-rbd-cluster, mas nenhum pod de rook-ceph-mon,

rook-ceph, manager ou api aparece

Esse problema pode acontecer ao tentar reinstalar o `ibm-rook-rbd-cluster` sem excluir o conteúdo do `dataDirHostPath` nos hosts para limpeza dos discos de armazenamento.

Para resolver o problema, conclua as tarefas a seguir:

1. Excluir o gráfico `ibm-rook-rbd-cluster` com falha
2. Excluir o gráfico do Rook Operator
3. Excluir o conteúdo de `dataDirHostPath`.
4. Limpar o disco que você usou para armazenamento.
5. Reinstalar o gráfico do Rook Operator.
6. Reinstalar o gráfico `ibm-rook-rbd-cluster`.

Depois que um nó do trabalhador é reiniciado, o pod do agente do Rook permanece no status de erro

1. Obtenha as informações do pod.

```
kubectl get po -o wide
```

O seguinte é uma saída de amostra:

```
NAME                            READY     STATUS                 RESTARTS   AGE       IP
NODE
rook-agent-5rst5                1/1       Running                0          2d
9.5.28.147    9.5.28.147
rook-agent-bsrrx                1/1       Running                0          2d
9.5.28.143    9.5.28.143
rook-agent-zq4bm                0/1       CreateContainerError   1          2d
9.5.28.146    9.5.28.146
rook-api-86b5b8849c-fjqf8       1/1       Running                0          7m
10.1.68.153   9.5.28.147
rook-ceph-mgr0-9c56544c8-2mxqr  1/1       Running                0          2d
10.1.19.31    9.5.28.143
```

```
rook-ceph-mon0-g5t7m             1/1       Running                0          2d
10.1.19.30    9.5.28.143
rook-ceph-mon1-zl5px             1/1       Running                5          7m
10.1.0.164    9.5.28.146
rook-ceph-mon2-jjjht             1/1       Running                0          2d
10.1.68.151   9.5.28.147
rook-ceph-osd-9.5.28.143-2bpl6   1/1       Running                0          2d
10.1.19.32    9.5.28.143
rook-ceph-osd-9.5.28.146-8qwbx   1/1       Running                5          7m
10.1.0.165    9.5.28.146
rook-ceph-osd-9.5.28.147-mcksg   1/1       Running                0          2d
10.1.68.152   9.5.28.147
rook-operator-947bf78c6-58nng    1/1       Running                0          2d
10.1.19.22    9.5.28.143
```

2. Obtenha informações sobre o pod do agente Rook.

```
kubectl describe po rook-agent-zq4bm
```

O seguinte é uma saída de amostra:

```
Name:        rook-agent-zq4bm
Namespace:    default
Node:        9.5.28.146/9.5.28.146
...

6m        6m        3    kubelet, 9.5.28.146    spec.containers{rook-agent}    Warning
Failed    Error: Error response from daemon:
Conflict. The container name "/k8s_rook-agent_rook-agent-zq4bm_default_5b2c4423-5a8e-11e8-a2b0-
005056a7db67_2" is already in
use by container ac71dc3e805f470d44afe6660f668e71832753505532625a9f30905c30f2063a. You have to
remove (or rename) that
container to be able to reuse that name.
```

3. Efetue login no nó no qual o pod está falhando. Encerre o contêiner em conflito conforme relatado no erro.

```
docker kill ac71dc3e805f470d44afe6660f668e71832753505532625a9f30905c30f2063a
ac71dc3e805f470d44afe6660f668e71832753505532625a9f30905c30f2063a
```

O pod do agente inicia a execução normalmente.

```
kubectl get po -o wide
```

O seguinte é uma saída de amostra:

```
NAME                             READY     STATUS    RESTARTS   AGE       IP            NODE
rook-agent-5rst5                 1/1       Running   0          2d        9.5.28.147
9.5.28.147
rook-agent-bsrrx                 1/1       Running   0          2d        9.5.28.143
9.5.28.143
rook-agent-zq4bm                 1/1       Running   2          2d        9.5.28.146
9.5.28.146
rook-api-86b5b8849c-fjqf8        1/1       Running   0          12m       10.1.68.153
9.5.28.147
rook-ceph-mgr0-9c56544c8-2mxqr   1/1       Running   0          2d        10.1.19.31
9.5.28.143
rook-ceph-mon0-g5t7m             1/1       Running   0          2d        10.1.19.30
9.5.28.143
rook-ceph-mon1-zl5px             1/1       Running   5          12m       10.1.0.164
9.5.28.146
rook-ceph-mon2-jjjht             1/1       Running   0          2d        10.1.68.151
9.5.28.147
rook-ceph-osd-9.5.28.143-2bpl6   1/1       Running   0          2d        10.1.19.32
9.5.28.143
rook-ceph-osd-9.5.28.146-8qwbx   1/1       Running   5          12m       10.1.0.165
9.5.28.146
rook-ceph-osd-9.5.28.147-mcksg   1/1       Running   0          2d        10.1.68.152
9.5.28.147
rook-operator-947bf78c6-58nng    1/1       Running   0          2d        10.1.19.22
9.5.28.143
```

Minio

O Minio é um servidor Object Storage leve compatível com o Amazon S3.

Visão geral

O Minio é um servidor Object Storage distribuído de alto desempenho, que é projetado para infraestrutura de nuvem privada em larga escala. O Minio agrega volumes persistentes (PVs) no Object Storage distribuído escalável usando as APIs de REST do Amazon S3.

Ele mais adequado para armazenamento de dados não estruturados, como fotos, vídeos, arquivos de log, backups, VMs e imagens de contêiner. Para obter mais informações sobre o Minio, consulte https://minio.io.

O Minio suporta o modo de Gateway de armazenamento conectado à rede (NAS) independente distribuído. Para obter mais informações, consulte Guia de Iniciação Rápida do MinIO Distribuído e Gateway NAS do MinIO.

Requisitos do sistema

O Minio é suportado em clusters Linux® x86_64, Linux® on Power® (ppc64le) e Linux® on IBM® Z and LinuxONE.

O Minio requer um armazenamento de bloco para persistência, que pode ser provisionado usando uma classe de armazenamento. No IBM® Cloud Private, é possível usar o armazenamento do Rook Ceph, do GlusterFS ou do vSphere para criar a classe de armazenamento. Em seguida, é possível fornecer a classe de armazenamento para o gráfico Helm do Minio para criar as instâncias do servidor Minio.

Cenários de Implementação

O servidor Minio pode ser implementado em um modo independente ou distribuído.

- **Modo independente:** em um modo independente, o Minio está limitado a executar um pod. Em um ambiente de produção, o Minio deve ser implementado em um modo distribuído. Para obter mais informações, consulte Implementar o Minio no Kubernetes.
- **Modo distribuído:** com o Minio no modo distribuído, é possível agrupar múltiplas unidades (mesmo em máquinas diferentes) em um servidor Object Storage único. Como as unidades são distribuídas em vários nós, o Minio distribuído pode suportar múltiplas falhas do nó e, ainda assim, assegurar a proteção de dados integral. O Minio distribuído fornece proteção contra múltiplas falhas de nó ou de unidade. Para obter mais informações, consulte Guia de iniciação rápida do Minio distribuído.
- **Modo de gateway NAS:** o Gateway MinIO inclui compatibilidade Amazon S3 no armazenamento NAS. É possível executar várias instâncias do Minio no mesmo volume NAS compartilhado que um gateway de objeto distribuído. Para usar Minio como Gateway NAS, você precisa de um PV que seja executado com plug-ins de volume suportados por ReadWriteMany.

É possível implementar quantas instâncias do Minio desejar. O acesso ao Minio é controlado por segredo e chave de acesso. É possível ter uma instância do Minio para seu cluster do IBM Cloud Private inteiro ou ter uma instância do Minio somente para seu aplicativo.

Pré-requisitos e preparação de nó

Para obter mais informações, consulte Pré-requisitos e limitações.

Implementando o Minio

É possível implementar o servidor Minio durante a instalação do IBM Cloud Private. Se precisar incluir um servidor Minio após a instalação do IBM Cloud Private, isso poderá ser feito implementando o Minio como um serviço de complemento ou implementando o gráfico Helm que está disponível no catálogo do IBM Cloud Private.

Para obter mais informações, consulte Configurando o Minio.

Verificando a instalação do servidor Minio

Para verificar se o servidor Minio foi instalado corretamente, é possível usar a linha de comandos do Minio. Para obter mais informações, consulte o Guia de iniciação rápida do cliente Minio.

Gerenciando seu cluster

As operações a seguir poderão ser executadas após o cluster ser instalado.

Upgrade

É possível usar o recurso de upgrade do HELM para fazer upgrade do gráfico Minio.

Desinstalação

É possível excluir o gráfico Helm para desinstalar o Minio. Para obter mais informações, consulte Desinstalando o gráfico.

Resolução de problemas

Consulte Resolução de problemas do Minio.

- Pré-requisitos e limitações
- Configurando o Minio
- Verificando a configuração do Minio
- Monitoramento do Minio
- Fazendo upgrade do Minio
- Resolução de Problemas do Minio

Pré-requisitos e Limitações

Pré-requisitos e limitações para configurar o Minio em seu cluster do IBM® Cloud Private.

Pré-requisitos

Assegure-se de que as configurações a seguir estejam concluídas em seu cluster:

- O Kubernetes 1.10 ou superior com APIs Beta deve estar ativado.
- Um objeto secreto que contém as chaves de acesso e de segredo que estão no formato codificado base64 deve estar disponível. Para obter mais informações, consulte Minio.
- Se você estiver configurando o Minio em um modo distribuído, deve-se configurar o armazenamento de bloco. Esse armazenamento pode ser GlusterFS, Ceph ou qualquer outro provedor de armazenamento suportado pelo Kubernetes. O armazenamento de bloco deve estar disponível por meio do fornecimento de volume dinâmico, usando uma classe de armazenamento.
- Se você estiver configurando o Minio em um modo independente, será possível usar `emptyDir` para armazenamento de dados. No entanto, quando um pod é removido de um nó, os dados são excluídos permanentemente. Se você precisar de persistência de dados, configure o armazenamento de bloco e use uma classe de armazenamento para fornecimento de volume dinâmico.
- Se você estiver implementando o Minio no modo de gateway de armazenamento conectado à rede (NAS), será necessário um volume persistente (PV) em execução com os plug-ins de volume suportados `ReadWriteMany`. Para provedores que suportam o modo `ReadWriteMany`, consulte https://kubernetes.io/docs/concepts/storage/persistent-volumes/#access-modes.

Requisitos de Software

- O IBM Cloud Private Versão 3.1.0 ou mais recente deve ser instalado.
- O IBM Cloud Private suporta o IBM Minio Objectstore Chart v2.4.7, o Minio versão RELEASE.2019-04-09T01-22-30Z.1 e o Minio mc versão RELEASE.2019-04-03T17-59-57Z.1.

Configurando o Minio

Configure o Minio durante ou após a instalação do IBM® Cloud Private .

- Configurando o Minio durante a instalação do IBM Cloud Private
- Configurando o Minio após a instalação do IBM Cloud Private

Configurando o Minio durante a instalação do IBM Cloud Private

Configure o Minio quando instalar o cluster do IBM® Cloud Private.

Conclua estas etapas para configurar o Minio:

1. Ativar armazenamento Minio. Configure `storage-minio: enabled` na lista de serviços de gerenciamento no arquivo `/<installation_directory>/cluster/config.yaml`.

   ```
   management_services:
    istio: disabled
    vulnerability-advisor: disabled
    storage-glusterfs: disabled
    storage-minio: enabled
   ```

2. Inclua a parte de código a seguir no arquivo `config.yaml`:

```
storage-minio:
  mode: standalone
  accessKey: "admin"
  secretKey: "admin1234"
  minioAccessSecret: "minio-secret"
  configPath: "/minio/.minio/"
  mountPath: "/export"
  replica: 4
  persistence:
    enabled: false
    useDynamicProvisioning: false
    storageClass: standard
    accessMode: ReadWriteOnce
    size: 10Gi
  service:
    type: ClusterIP
    clusterIP: None
    loadBalancerIP: None
    port: 9000
    nodePort: 31311
    prometheusEnable: false
    prometheusPath: '/minio/prometheus/metrics'
    prometheusPort: '9000'
  ingress:
    enabled: false
    annotations: {}
    path: /
    hosts: ""
    tls: []
  tls:
    enabled: false
    type: "cert-manager-generated"
    minioTlsSecret: ""
    issuerRef:
      name: "icp-ca-issuer"
      kind: "ClusterIssuer"
    clusterDomain: "cluster.local"
  nodeSelector: ""
  tolerações: ""
```

A seguir estão as descrições dos parâmetros que são necessários para uma configuração simples. Para obter uma configuração avançada, consulte a lista completa de parâmetros disponíveis: Minio.

- `mode` é o modo do servidor Minio. As opções válidas são `standalone` ou `distribuída`.
- `accessKey` é a chave de acesso do servidor Minio. A chave deve ter de 5 a 20 caracteres.
- `secretKey` é a chave secreta do servidor Minio. A chave deve ter de 8 a 40 caracteres.
- `minioAccessSecret` é o nome do objeto do segredo do Kubernetes.
  Nota: será possível configurar esse parâmetro somente se você estiver configurando o Minio após a instalação do IBM Cloud Private.
- `configPath` é o local do arquivo de configuração padrão.
- `mountPath` é o caminho de montagem padrão para a unidade persistente.
- `replica` é o número de nós do Minio. Esse parâmetro é aplicável somente ao modo distribuído do Minio. O valor deve ser `4 - 32` nós.
- `persistence.enabled` é para configurar se o volume persistente é usado para armazenar dados.
- `persistence.useDynamicProvisioning` é para configurar se a solicitação de volume persistente (PVC) usa uma classe de armazenamento para ligar o volume.

- `persistence.storageClass` é o nome da Classe de armazenamento para ligar o PVC. Especifique um nome de classe de armazenamento se você configurar o `persistence.useDynamicProvisioning` como `true`.
- `persistence.accessMode` configura o modo de acesso. As opções válidas são `ReadWriteOnce` ou `ReadOnly`.
- `persistence.size` é o tamanho do PVC a ser criado usando uma classe de armazenamento.
- `service.type` é o tipo de serviço do Kubernetes. Os valores permitidos são `ClusterIP`, `LoadBalancer` ou `NodePort`.
- `service.clusterIP` é o endereço IP do cluster de serviço do Kubernetes. O endereço IP do cluster padrão será usado se você configurar o parâmetro `service:type` como `ClusterIP`. É possível especificar outro endereço IP, se necessário.
- `service.loadBalancerIP` é o endereço IP do balanceador de carga do serviço do Kubernetes. O endereço IP do balanceador de carga padrão será usado se você configurar o parâmetro `service.type` como `loadBalancer`. É possível especificar outro endereço IP, se necessário.
- `service.port` é a porta do Kubernetes na qual o serviço é exposto. O valor padrão é `9000`.
- `service.nodePort` expõe o serviço no endereço IP do nó em uma porta estática. O valor padrão é `31311`.
- `service.prometheusEnable` ativa a extração do Prometheus. O valor padrão é `false`.
- `service.prometheusPath` é o caminho de métrica. O valor padrão é `/minio/prometheus/metrics`.
- `service.prometheusPort` é a porta para métricas de extração. O valor padrão é `9000`.
- `ingress.enabled` ativa o controlador de ingresso.
- `ingress.annotations` são anotações para ingresso. Por exemplo, `{kubernetes.io/ingress.class: nginx, kubernetes.io/tls-acme: "true"}`.
- `ingresss.path` é o caminho do controlador de ingresso. O valor padrão é `/`.
- `ingress.hosts` são nomes de host que são aceitos pelo controlador de ingresso. Por exemplo, `[ "chart-example1.local "," chart-example2.local " ]`.
- `ingress.tls` é a configuração do TLS do controlador de ingresso. Por exemplo, `[{"secretName": "chart-example-tls", "hosts": ["chart-example.local", "chart-example.local"]}]`.
- `tls.enabled` ativa o servidor Minio com certificados TLS quando configurado como `true`. O valor padrão é `false`.
- `tls.type` é para especificar se um gráfico deve gerar automaticamente um certificado TLS usando o emissor de cert-manager ou deve usar o fornecido. Os valores válidos são `provided` e `cert-manager-generated`. Se você está fornecendo o certificado, deve-se criar um segredo que contenha uma chave privada, um certificado TLS e um certificado de autoridade de certificação (CA). Você fornece o nome do segredo no parâmetro `tls.minioTlsSecret`.
- `tls.minioTlsSecret` é o segredo que você cria e que contém uma chave privada (key private.key), o certificado TLS (key public.crt) e um certificado de CA (key ca.crt) para configurar o servidor Minio com certificados TLS. Deve-se criar e especificar o segredo no mesmo namespace no qual você está implementando o gráfico. Use esse parâmetro se você configurar `tls.type` como `provided`.
- `tls.issuerRef.name` é o nome do ClusterIssuer ou Issuer de quem o certificado x509 assinado é obtido. Você deverá especificar esse valor se tiver especificado o valor de `tls.type` como `cert-manager-generated`.
- `tls.issuerRef.kind` é o tipo de CA de quem o certificado x509 assinado é obtido. Os valores válidos são: `ClusterIssuer` e `Issuer`. Você deverá especificar esse valor se tiver especificado o valor de `tls.type` como `cert-manager-generated`.
- `tls.clusterDomain` é o nome do domínio do cluster usado para gerar um certificado usando cert-manager. Especifique o nome do domínio do cluster aqui. Esse parâmetro é aplicável quando `tls.type` é configurado como `cert-manager-generated`.
- `nodeSelector` está incluindo rótulos de nó para designação de pod. Inclua etiquetas como par {"key":"value"}. Por exemplo, {"role": "minio-node"}.
- `tolerations` são rótulos de tolerância para designação de pod. Inclua etiquetas como par {"key":"value"}. Por exemplo, [{"operator":"Equal"}, {"effect":"NoSchedule"}].

A configuração do Minio está concluída. Prossiga com a instalação do IBM Cloud Private .

Configurando o Minio após a instalação do IBM Cloud Private

Configure o Minio depois de instalar o cluster do IBM® Cloud Private.

Configurar o Minio instalando o gráfico Helm

Instale o gráfico `ibm-minio-objectstore` Versão 2.4.7. Para obter mais informações, consulte Minio.

Configure o Minio como um serviço de complemento

Conclua estas etapas para configurar o Minio:

1. Ative o armazenamento do Minio. Consulte a etapa 1 .
2. Atualize o `config.yaml` arquivo. Consulte a etapa 2 .

3. Execute o comando de complemento para configurar o Minio:

```
docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
```

Verificando a configuração do Minio

Verifique se o Minio está configurado corretamente.

Depois de instalar com sucesso o gráfico do Helm, siga estas etapas para verificar se a configuração do Minio está correta:

1. Verifique se o serviço e os pods do Minio estão em execução.

```
helm status minio -- tls
```

Exemplo de saída:

```
LAST DEPLOYED: Mon Jan 28 01:39:43 2019
NAMESPACE: kube-system
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME                          TYPE       CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
minio-ibm-minio-objectstore   ClusterIP  10.0.38.172  <none>       9000/TCP  13m

==> v1beta1/StatefulSet
NAME                          DESIRED  CURRENT  AGE
minio-ibm-minio-objectstore   4        4        13m

==> v1/Pod(related)
NAME                            READY  STATUS   RESTARTS  AGE
minio-ibm-minio-objectstore-0   1/1    Running  0         13m
minio-ibm-minio-objectstore-1   1/1    Running  0         13m
minio-ibm-minio-objectstore-2   1/1    Running  0         12m
minio-ibm-minio-objectstore-3   1/1    Running  0         12m

==> v1/Secret
NAME                          TYPE    DATA  AGE
minio-ibm-minio-objectstore   Opaque  0     13m

==> v1/ConfigMap
NAME                                    DATA  AGE
minio-ibm-minio-objectstore             2     13m
minio-ibm-minio-objectstore-test-config 1     13m
```

1. Acesse o Minio por meio da porta 9000 ou usando o cliente `mc`.

   - Acesse o Minio por meio da porta 9000 no DNS a seguir de dentro de seu cluster:

     ```
     mini-ibm-minio-objectstore-svc.kube-system.svc.cluster.local
     ```

     Para acessar o Minio a partir do host local, execute os comandos a seguir:

     ```
     export POD_NAME=$(kubectl get pods --namespace kube-system -l "release=minio" -o jsonpath="{.items[0].metadata.name}")

     kubectl port-forward $POD_NAME 9000 -- namespace kube-system
     ```

     Para obter mais informações sobre o encaminhamento de porta, consulte Encaminhamento de porta.

     Agora é possível acessar o servidor Minio em `http://localhost:9000`.

   - Siga estas etapas para conectar-se ao servidor Minio usando o cliente mc:

     1. Faça download do cliente mc do Minio. Para obter mais informações, consulte Guia de Iniciação Rápida do Minio Client.

     2. Configure o servidor Minio:

        ```
        mc config host add minio-ibm-minio-objectstore-local http://localhost:9000 <ACCESSKEY> <SECRETKEY> S3v4
        ```

3. Visualize informações do servidor Minio.

```
mc ls minio-ibm-minio-objectstore-local
```

Também é possível usar seu navegador ou o SDK do Minio para acessar o servidor Minio. Para obter mais informações, consulte SDKS do MINIO.

2. Mude o pod do cliente Minio usando o arquivo YAML a seguir:

Nota:

- Se o pod do cliente Minio for ativado em um namespace diferente do servidor Minio, o `<Minio Service Name>` deverá ser um nome de serviço completo: `<Minio Service Name>.<Namespace of Minio server>.<svc>.<cluster domain>`.
  Por exemplo, quando o serviço Minio é executado em um namespace padrão, o nome do serviço Minio é `minio-ibm-minio-objectstore` e o domínio do cluster é `cluster.local` e, em seguida, o nome do serviço completo é o seguinte: `http://minio-ibm-minio-objectstore.default.svc.cluster.local:9000`
- Substitua os valores `<Minio Service Name>`, `<ACCESSKEY>` e `<SECRETKEY>` no arquivo YAML a seguir.

```
apiVersion: v1
kind: Pod
metadata:
 name: minioclient
 # Note that the Pod does not need to be in the same namespace as the loader.
  labels:
    app: minioclient
spec:
  containers:
  - name: minio
    image: ibmcom/minio-mc:RELEASE.2019-04-03T17-59-57Z.1
    imagePullPolicy: IfNotPresent
    command: ["/test/test-minio.sh"]
    volumeMounts:
      - name: test-minio
        mountPath: "/test"
  volumes:
    - name: test-minio
      configMap:
        name: test-minio
        defaultMode: 0745
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: test-minio
data:
  test-minio.sh: |
    #!/bin/sh
    # If TLS is not enabled use the following URL:
    mc config host add myminio http://<Minio Service Name>:9000 <ACCESSKEY>  <SECRETKEY> S3v4
   # If TLS is enabled for minio service, use the following URL. If certificate is self signed
user --insecure
   # mc config host add myminio https://<Minio Service Name>:9000 <ACCESSKEY>  <SECRETKEY> S3v4
--insecure
    sleep 3600
```

3. Depois que o pod do Minio estiver ativo e em execução, verifique o status de um ou mais servidores Minio.

```
kubectl exec minioclient mc admin info myminio
```

O seguinte é uma saída de amostra:

```
● minio-ibm-minio-objectstore-3.minio-ibm-minio-objectstore.kube-system.svc.cluster.local:9000
  Uptime : online since 32 minutes ago
 Version : 2018-11-30T03:56:59Z
  Region : us-east-1
SQS ARNs : <none>
   Stats : Incoming 0B, Outgoing 0B
 Storage : Used 12MiB
   Disks : 4, 0

● minio-ibm-minio-objectstore-0.minio-ibm-minio-objectstore.kube-system.svc.cluster.local:9000
  Uptime : online since 32 minutes ago
```

```
 Version : 2018-11-30T03:56:59Z
  Region : us-east-1
SQS ARNs : <none>
   Stats : Incoming 466B, Outgoing 638B
 Storage : Used 12MiB
   Disks : 4, 0

• minio-ibm-minio-objectstore-1.minio-ibm-minio-objectstore.kube-system.svc.cluster.local:9000
  Uptime : online since 32 minutes ago
 Version : 2018-11-30T03:56:59Z
  Region : us-east-1
SQS ARNs : <none>
   Stats : Incoming 0B, Outgoing 0B
 Storage : Used 12MiB
   Disks : 4, 0

• minio-ibm-minio-objectstore-2.minio-ibm-minio-objectstore.kube-system.svc.cluster.local:9000
  Uptime : online since 32 minutes ago
 Version : 2018-11-30T03:56:59Z
  Region : us-east-1
SQS ARNs : <none>
   Stats : Incoming 466B, Outgoing 674B
 Storage : Used 12MiB
   Disks : 4, 0
```

4. Verifique se é possível criar e excluir um depósito.

   ```
   kubectl exec minioclient mc mb myminio / test-bucket
   ```

   O seguinte é uma saída de amostra:

   ```
   Bucket criado com sucesso ` myminio/test-bucket `.
   ```

   ```
   kubectl exec minioclient mc rm  myminio/test-bucket
   ```

   Amostra de saída:

   ```
   Removendo ` myminio/test-bucket `.
   ```

Monitorando o Minio

É possível monitorar o funcionamento do seu servidor Minio no painel de monitoramento do cluster do IBM® Cloud Private.

O Minio exporta dados compatíveis do Prometheus como um terminal não autorizado em `/minio/prometheus/metrics`.

Para ativar a exportação de métricas do Prometheus no servidor Minio, defina a configuração `prometheusEnable: true` enquanto você instala o gráfico Minio. Ao configurar o parâmetro como `true`, o monitoramento é ativado para todos os terminais, mesmo quando o servidor Minio usa um certificado TLS autoassinado.

Depois que o gráfico Minio é instalado, é possível visualizar as métricas no painel Grafana do IBM Cloud Private.

Se você deseja ativar o monitoramento apenas para os terminais que usam um certificado TLS autoassinado, siga as etapas em Ativando o monitoramento quando o servidor Minio está configurado com um certificado TLS autoassinado.

Ativando o monitoramento quando o servidor Minio está configurado com um certificado TLS autoassinado

Se a instalação do seu servidor Minio usa um certificado TLS autoassinado, as métricas não são coletadas pelo Prometheus por padrão. É possível permitir que o Prometheus colete métricas de terminais que são configurados com um certificado TLS assinado em qualquer uma das maneiras a seguir:

Configure insecure_skip_verify: true para job_name: 'kubernetes-service-endpoints-with-tls'

1. Edite o mapa de configuração `monitoring-prometheus`.
   Nota: é possível editar o mapa de configuração `monitoring-prometheus` usando a CLI kubectl ou usando o IBM Cloud Private console de gerenciamento. Se você estiver usando a CLI kubectl, primeiro certifique-se de configurá-la. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

   ```
   kubectl edit configmap -n kube-system monitoring-prometheus
   ```

2. Localize a tarefa `kubernetes-service-endpoints-with-tls` e configure `insecure_skip_verify: true`.

   A seguir está um conteúdo do arquivo de amostra:

```
    - job_name: 'kubernetes-service-endpoints-with-tls'

       kubernetes_sd_configs:
         - role: endpoints

       relabel_configs:
         - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_scrape]
           action: keep
           regex: true
...
      tls_config:
         ca_file: /opt/ibm/monitoring/caCerts/tls.crt
         cert_file: /opt/ibm/monitoring/certs/tls.crt
         key_file: /opt/ibm/monitoring/certs/tls.key
         insecure_skip_verify: true                    ===>>> Set this parameter to true
<<<====
```

Inclua detalhes do servidor Minio no configmap na seção scrape_configs

1. Obtenha o endereço IP do serviço Minio.

   ```
   kubectl get svc -n default
   ```

   O seguinte é uma saída de amostra:

```
NAME                          TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
kubernetes                    ClusterIP   10.0.0.1      <none>        443/TCP    21d
minio-ibm-minio-objectstore   ClusterIP   10.0.0.135    <none>        9000/TCP   18m
```

2. Edite o mapa de configuração `monitoring-prometheus`. **Nota:** é possível editar o mapa de configuração `monitoring-prometheus` usando a CLI kubectl ou usando o IBM Cloud Private console de gerenciamento. Se você estiver usando a CLI kubectl, primeiro certifique-se de configurá-la. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

   ```
   kubectl edit configmap -n kube-system monitoring-prometheus
   ```

3. Especifique o endereço IP e a porta do serviço do Minio na seção `static_configs.targets`:

```
   scrape_configs:
     - job_name: minio
       metrics_path: /minio/prometheus/metrics
       static_configs:
         - targets:
           - 10.0.0.135:9000
       scheme: https
       tls_config:
         insecure_skip_verify: true
```

   {pre}

4. Salve e saia do mapa de configuração.

Para acessar o painel de monitoramento do cluster do IBM Cloud Private, efetue login na console de gerenciamento e clique em **Menu** > **Plataforma** > **Monitoramento**. Ou, é possível abrir `https://<IP_address>:8443/grafana`, em que `IP_address` é o DNS ou endereço IP que é usado para acessar o console do IBM Cloud Private. Visualize o painel `Funcionamento do Minio de armazenamento`.

Atualizando o Minio

Faça o upgrade do gráfico do Helm do Minio da Versão 1.6.0 ou 1.6.2 para a Versão 2.4.7 no cluster do IBM® Cloud Private.

Seu serviço Minio será submetido a upgrade automaticamente se você tiver configurado o serviço Minio no IBM Cloud Private Versão 3.1.2 ou 3.1.0 de uma das maneiras a seguir:

- Você configurou o serviço Minio enquanto instalou o cluster do IBM Cloud Private.
- Você usou o comando de complemento para implementar o serviço Minio depois de instalar o IBM Cloud Private.

Nota: o retrocesso de serviço do Minio é suportado apenas no modo `independente`.

Nas liberações anteriores, se você configurou o Minio instalando o gráfico do Helm, conclua as etapas nas seções a seguir.

É possível fazer upgrade da instalação do Helm do Minio usando a CLI do Helm ou usando a console de gerenciamento.

Faça upgrade do Minio usando a CLI do Helm

1. Obtenha o valor de configuração da liberação do Helm instalada usando o comando a seguir:

   ```
   helm list --tls | grep mini
   ```

   O seguinte é uma saída de amostra:

   ```
   minio                           1           Tue Feb  5 01:56:31 2019    DEPLOYED    ibm-mino-
   objectstore-1.6.0         default

   helm get values <Helm release name> --tls > values-minio.yaml
   ```

   Por exemplo, se o nome da liberação do Helm for `minio`, execute o comando a seguir:

   ```
    helm get values minio -- tls > values-minio.yaml
   ```

2. Crie um arquivo de substituição.

   Nota: na Versão 1.6.2 do gráfico do Helm do Minio, as variáveis de configuração `minioAccessSercret` e `tls.minioTlsSercret` são corrigidas para `minioAccessSecret` e `tls.minioTlsSercret`. Você deve preencher essas variáveis no arquivo de substituição com base nos valores das antigas variáveis.

   Crie o arquivo de substituição a seguir:

   ```
   configPath: "/minio/.minio/"
   minioAccessSecret: minio
   tls:
     minioTlsSecret: ""
   ```

   Nota:

   - O valor de `minioAccessSecret` pode ser obtido do antigo values-minio.yaml da variável `minioAccessSercret`.

     ```
     minioAccessSercret: minio
     ```

   - Se você ativou o TLS para o servidor Minio e configurou a `Opção de provisão de certificado TLS` para `Fornecido`, o valor de `tls.minioTlsSecret` poderá ser obtido a partir do antigo values-minio.yaml da variável `tls.minioTlsSercret`.

     ```
     tls:
       minioTlsSercret: "minio-tls-secret"
     ```

     Se você não ativou o TLS para o servidor Minio e não configurou a `Opção de provisão de certificado TLS` para `Fornecido`, não será necessário incluir `tls.minioTlsSecret` no arquivo de substituição.

3. Faça upgrade do gráfico.

   Execute o comando a seguir para fazer upgrade do gráfico:

   ```
     helm upgrade --force -f values-minio.yaml -f override.yaml --version=1.6.2 minio ibm-
   charts/ibm-minio-objectstore --tls
   ```

4. Verifique sua versão do gráfico.

   ```
   helm histórico minio -- tls
   ```

   O seguinte é uma saída de amostra:

   ```
   REVISION        UPDATED                         STATUS          CHART
   DESCRIPTION
   1               Tue Apr 30 03:43:30 2019        SUPERSEDED      ibm-minio-objectstore-1.6.2
   Install complete
   2               Tue Apr 30 03:44:07 2019        DEPLOYED        ibm-minio-objectstore-2.4.7
   Upgrade complete
   ```

Faça upgrade do Minio usando o console de gerenciamento

1. Efetue logon no console de gerenciamento.
2. No menu de navegação, clique em **Cargas de Trabalho** > **Liberações do Helm**.
3. Localize a liberação do Minio.
4. Clique em **ACTION** > **Fazer upgrade** .
5. Selecione a versão 2.4.7.
   Nota: as etapas a seguir são necessárias, pois os nomes de variáveis são mudados na versão mais recente do gráfico do Helm.
6. Especifique o antigo valor `Segredo de Acesso` no campo **Segredo de Acesso**.
7. Nas versões anteriores, se você ativou o TLS para o servidor Minio e configurou a `Opção de provisão de certificado TLS` para `Fornecido`, especifique o valor antigo `Segredo do Minio TLS` no campo `Segredo do Minio TLS`.

Resolução de problemas do Minio

Revise problemas do Minio encontrados frequentemente.

- Reunindo informações
- Os pods do Minio travam com o status ContainerCreating
- O pod do servidor Minio trava no STATUS Pendente
- O Minio no modo distribuído não é acessível ao fornecer um certificado TLS
- Os depósitos e objetos Minio estão intermitentemente inacessíveis

Reunindo informações

Reúna informações para resolução de problemas do Minio.

Para resolução de problemas, deve-se reunir as informações a seguir:

Nota: é necessário configurar a CLI do kubectl para executar esses comandos. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

1. Versão do IBM Cloud Private.
2. Tipo de arquitetura dos nós em seu cluster. Por exemplo, Linux® ou Linux® on Power® (ppc64le)
3. Versão do gráfico Helm do Minio que você instalou. Use o comando a seguir:

   ```
   helm list --tls | grep mini
   ```

   O código a seguir é uma saída de amostra:

   ```
   minio                           1           Fri Sep 14 05:10:28 2018    DEPLOYED    ibm-minio-
   objectstore-1.6.0            default
   ```

4. Estado de implementação do Minio ou estado statefulset. Use os comandos a seguir:

   ```
   kubectl get statefulsets
   ```

   O código a seguir é uma saída de amostra:

   ```
   NAME                          DESIRED   CURRENT   AGE
   minio-ibm-minio-objectstore   4         4         2m

   kubectl describe  statefulsets
   ```

   O código a seguir é uma saída de amostra:

   ```
   Name:               minio-ibm-minio-objectstore
   Namespace:          default
   CreationTimestamp:  Fri, 14 Sep 2018 05:10:37 -0700
   Selector:           app=ibm-minio-objectstore,release=minio
   Labels:             app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-1.6.0
                      heritage=Tiller
   ...
   ```

5. Status do serviço do Minio. Use os comandos a seguir:

    1. Obtenha o serviço.

    ```
    kubectl get svc
    ```

    O código a seguir é uma saída de amostra:

    ```
    NAME                          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
    kubernetes                    ClusterIP   10.0.0.1     <none>        443/TCP    10d
    minio-ibm-minio-objectstore   ClusterIP   10.0.0.68    <none>        9000/TCP   4m
    ```

    2. Obtenha a descrição do serviço.

    ```
    kubectl describe svc
    ```

    O código a seguir é uma saída de amostra:

    ```
    Name:              kubernetes
    Namespace:         default
    Labels:            component=apiserver
                      provider=kubernetes
    Annotations:       <none>
    Selector:          <none>
    Type:              ClusterIP
    IP:                10.0.0.1
    Port:              https  443/TCP
    TargetPort:        8001/TCP
    Endpoints:         10.41.1.182:8001
    Session Affinity:  None
    Events:            <none>
    .
    .
    Name:              minio-ibm-minio-objectstore
    Namespace:         default
    Labels:            app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-1.6.0
                      heritage=Tiller
                      release=minio
    Annotations:       prometheus.io/path=/minio/prometheus/metrics
                      prometheus.io/port=9000
                      prometheus.io/scrape=false
    Selector:          app=ibm-minio-objectstore,release=minio
    Type:              ClusterIP
    IP:                10.0.0.68
    Port:              service  9000/TCP
    TargetPort:        9000/TCP
    Endpoints:         10.1.137.211:9000,10.1.180.251:9000,10.1.236.84:9000 + 1 more...
    Session Affinity:  None
    Events:            <none>
    ```

6. Os pods, os logs e a descrição do servidor Minio.

    1. Obtenha todos os pods do Minio.

    ```
    kubectl get po | grep mini
    ```

    O código a seguir é uma saída de amostra:

    ```
    minio-ibm-minio-objectstore-0         1/1       Running   0          4m
    minio-ibm-minio-objectstore-1         1/1       Running   0          4m
    minio-ibm-minio-objectstore-2         1/1       Running   0          4m
    minio-ibm-minio-objectstore-3         1/1       Running   0          4m
    ```

    2. Obtenha logs e descrição de todos os pods. O código a seguir é um exemplo de comando:

    ```
    kubectl describe po minio-ibm-minio-objectstore-0
    ```

    O código a seguir é uma saída de amostra:

    ```
    Name:               minio-ibm-minio-objectstore-0
    Namespace:          default
    Priority:           0
    PriorityClassName:  <none>
    Node:               10.41.4.202/10.41.4.202
    ```

```
Start Time:         Fri, 14 Sep 2018 05:10:37 -0700
Labels:             app=ibm-minio-objectstore
                  chart=ibm-minio-objectstore-1.6.0
                  controller-revision-hash=minio-ibm-minio-objectstore-7b77fd5658
                  heritage=Tiller
                  release=minio
                  statefulset.kubernetes.io/pod-name=minio-ibm-minio-objectstore-0
Annotations:        kubernetes.io/psp=00-rook-ceph-operator
                  productID=Minio_RELEASE.2018-08-21T00-37-20Z_free_00000
                  productName=Minio
                  productVersion=RELEASE.2018-08-21T00-37-20Z
                  scheduler.alpha.kubernetes.io/critical-pod=
Status:             Running
IP:                 10.1.236.84
Controlled By:      StatefulSet/minio-ibm-minio-objectstore
Containers:
ibm-minio-objectstore:
  Container ID:
docker://5e71782564d1c956d6855006f06472773da59ad22743a52bb64f83f4ac0ccf02
  Image:         minio/minio:RELEASE.2018-08-21T00-37-20Z
  Image ID:      docker-
pullable://minio/minio@sha256:3145ff901d491f46e59dd9fb79dc2771e75a524bbfdba8fa8cd35723960
fe7d5
  Port:          9000/TCP
  Host Port:     0/TCP
  Command:
    /bin/sh
    -ce
    cp /tmp/config.json /root/.minio/ && /usr/bin/docker-entrypoint.sh minio -C
/root/.minio/ server http://minio-ibm-minio-objectstore-0.minio-ibm-minio-
objectstore.default.svc.cluster.local/export http://minio-ibm-minio-objectstore-1.minio-
ibm-minio-objectstore.default.svc.cluster.local/export http://minio-ibm-minio-
objectstore-2.minio-ibm-minio-objectstore.default.svc.cluster.local/export http://minio-
ibm-minio-objectstore-3.minio-ibm-minio-objectstore.default.svc.cluster.local/export
  State:          Running
    Started:      Fri, 14 Sep 2018 05:10:39 -0700
  Ready:          True
  Restart Count:  0
  Requests:
    cpu:     250m
    memory:  256Mi
  Environment:
    MINIO_ACCESS_KEY:  <set to the key 'accesskey' in secret 'minio'>  Optional: false
    MINIO_SECRET_KEY:  <set to the key 'secretkey' in secret 'minio'>  Optional: false
  Mounts:
    /export from export (rw)
    /root/.minio/ from minio-config-dir (rw)
    /tmp/config.json from minio-server-config (rw)
    /var/run/secrets/kubernetes.io/serviceaccount from default-token-tzxvl (ro)
Conditions:
Type              Status
Initialized       True
Ready             True
ContainersReady   True
PodScheduled      True
Volumes:
export:
  Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same
namespace)
  ClaimName:  export-minio-ibm-minio-objectstore-0
  ReadOnly:   false
minio-user:
  Type:        Secret (a volume populated by a Secret)
  SecretName:  minio
  Optional:    false
minio-server-config:
  Type:      ConfigMap (a volume populated by a ConfigMap)
  Name:      minio-ibm-minio-objectstore
  Optional:  false
minio-config-dir:
  Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
default-token-tzxvl:
  Type:        Secret (a volume populated by a Secret)
  SecretName:  default-token-tzxvl
```

```
  Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     CriticalAddonsOnly
                 dedicated
                 node.kubernetes.io/memory-pressure:NoSchedule
Events:
Type    Reason     Age   From                 Message
----    ------     ----  ----                 -------
Normal  Scheduled  5m    default-scheduler    Successfully assigned default/minio-ibm-
minio-objectstore-0 to 10.41.4.202
Normal  Pulled     5m    kubelet, 10.41.4.202  Container image "minio/minio:RELEASE.2018-
08-21T00-37-20Z" already present on machine
Normal  Created    5m    kubelet, 10.41.4.202  Created container
Normal  Started    5m    kubelet, 10.41.4.202  Started container
```

7. Informações sobre a solicitação de volume persistente (PVC), se você usou o fornecimento de armazenamento dinâmico.

   1. Obtenha todos os PVCs.

      ```
      Kubectl get pvc
      ```

      O código a seguir é uma saída de amostra:

      ```
      NAME                                     STATUS    VOLUME
      CAPACITY   ACCESS MODES   STORAGECLASS        AGE
      export-minio-ibm-minio-objectstore-0   Bound     pvc-a35afd44-b811-11e8-bc28-00000a2901b6
      5Gi        RWO            rook-ceph-block   46m
      export-minio-ibm-minio-objectstore-1   Bound     pvc-a71ea92b-b811-11e8-bc28-00000a2901b6
      5Gi        RWO            rook-ceph-block   46m
      export-minio-ibm-minio-objectstore-2   Bound     pvc-ab6f00af-b811-11e8-bc28-00000a2901b6
      5Gi        RWO            rook-ceph-block   46m
      export-minio-ibm-minio-objectstore-3   Bound     pvc-b27b35fc-b811-11e8-bc28-00000a2901b6
      5Gi        RWO            rook-ceph-block   45m
      ```

   2. Obtenha informações sobre um PVC. O código a seguir é um comando de amostra:

      ```
      kubectl describe pvc export-minio-ibm-minio-objectstore-0
      ```

      O código a seguir é uma saída de amostra:

      ```
      Name:          export-minio-ibm-minio-objectstore-0
      Namespace:     default
      StorageClass:  rook-ceph-block
      Status:        Bound
      Volume:        pvc-a35afd44-b811-11e8-bc28-00000a2901b6
      Labels:        app=ibm-minio-objectstore
                     release=minio
      Annotations:   control-plane.alpha.kubernetes.io/leader={"holderIdentity":"ee888338-b654-
      11e8-86f0-16fe371b5da0","leaseDurationSeconds":15,"acquireTime":"2018-09-
      14T11:30:52Z","renewTime":"2018-09-14T11:30:57Z","lea...
                     pv.kubernetes.io/bind-completed=yes
                     pv.kubernetes.io/bound-by-controller=yes
                     volume.beta.kubernetes.io/storage-provisioner=ceph.rook.io/block
      Finalizers:    [kubernetes.io/pvc-protection]
      Capacity:      5Gi
      Access Modes:  RWO
      Events:
      Type    Reason                 Age                From
      Message
      ----    ------                 ----               ----
      -------
      Normal  Provisioning           46m                ceph.rook.io/block rook-ceph-operator-
      5f84847c67-c6nzl ee888338-b654-11e8-86f0-16fe371b5da0  External provisioner is
      provisioning volume for claim "default/export-minio-ibm-minio-objectstore-0"
      Normal  ExternalProvisioning   46m (x2 over 46m)  persistentvolume-controller
      waiting for a volume to be created, either by external provisioner "ceph.rook.io/block"
      or manually created by system administrator
      Normal  ProvisioningSucceeded  46m                ceph.rook.io/block rook-ceph-operator-
      5f84847c67-c6nzl ee888338-b654-11e8-86f0-16fe371b5da0  Successfully provisioned volume
      pvc-a35afd44-b811-11e8-bc28-00000a2901b6
      ```

Os pods do Minio travam com o status ContainerCreating

Quando o Minio é implementado em qualquer modo, o primeiro pod do Minio pode travar com o status *ContainerCreating*.

Reúna informações sobre o problema

1. Obtenha a lista de pods.

   ```
   kubectl get po
   ```

   O código a seguir é uma saída de amostra:

   ```
   NAME                                          READY     STATUS              RESTARTS   AGE
   mc2                                           1/1       Running             53         2d
   minio-ibm-minio-objectstore-848fbcb6f5-2wpq2   0/1       ContainerCreating   0          3m
   ```

2. Verifique os logs. Se os logs estiverem vazios, descreva o pod.

   ```
   kubectl logs minio-ibm-minio-objectstore-848fbcb6f5-2wpq2
   ```

   O código a seguir é uma saída de amostra:

   ```
   Error from server (BadRequest): container "ibm-minio-objectstore" in pod "minio-ibm-minio-
   objectstore-848fbcb6f5-2wpq2" is waiting to start: ContainerCreating
   ```

3. Obtenha a descrição do pod.

   ```
   kubectl describe po minio-ibm-minio-objectstore-848fbcb6f5-2wpq2
   ```

   O código a seguir é uma saída de amostra:

   ```
   Name:               minio-ibm-minio-objectstore-848fbcb6f5-2wpq2
   Namespace:          default
   Priority:           0
   PriorityClassName:  <none>
   Node:               10.41.4.202/10.41.4.202
   Start Time:         Fri, 14 Sep 2018 05:52:01 -0700
   ...

   ...
   Events:
     Type     Reason       Age               From                  Message
     ----     ------       ----              ----                  -------
     Normal   Scheduled    5m                default-scheduler     Successfully assigned
   default/minio-ibm-minio-objectstore-848fbcb6f5-2wpq2 to 10.41.4.202
     Warning  FailedMount  1m (x10 over 5m)  kubelet, 10.41.4.202  MountVolume.SetUp failed for
   volume "minio-user" : secrets "minio" not found
     Warning  FailedMount  1m (x2 over 3m)   kubelet, 10.41.4.202  Unable to mount volumes for
   pod "minio-ibm-minio-objectstore-848fbcb6f5-2wpq2_default(f9036aa0-b81c-11e8-bc28-
   00000a2901b6)": timeout expired waiting for volumes to attach or mount for pod
   "default"/"minio-ibm-minio-objectstore-848fbcb6f5-2wpq2". list of unmounted volumes=[minio-
   user]. list of unattached volumes=[export minio-server-config minio-user minio-config-dir
   default-token-tzxvl]
   ```

   A descrição do pod indica que o pod não é capaz de montar o volume, já que o segredo do Minio está indisponível.

4. Verifique se o segredo está disponível no namespace no qual o Minio está implementado.

   ```
   kubectl get secret minio
   ```

   O código a seguir é uma saída de amostra:

   ```
   Nenhum recurso localizado.
   Error from server (NotFound): secrets "minio" not found
   ```

   A saída indica que o segredo não está disponível no namespace. Crie o segredo seguindo as instruções que estão no Arquivo Leia-me.

Resolva o problema

Para resolver o problema, conclua as etapas a seguir:

1. Exclua a liberação do Helm.
2. Inclua o segredo na configuração do gráfico Helm.

3. Implemente o gráfico de Helm.

O pod do servidor Minio trava no STATUS Pendente

Quando o Minio é implementado no modo distribuído com a alocação de armazenamento dinâmico, o pod do servidor pode travar com o status `Pendente`.

Reúna informações sobre o problema

1. Obtenha a lista de pods.

   ```
   kubectl get po
   ```

   O código a seguir é uma saída de amostra:

   ```
   NAME                                 READY     STATUS    RESTARTS   AGE
   mc2                                  1/1       Running   54         2d
   minio-ibm-minio-objectstore-0        0/1       Pending   0          7s
   ```

2. Obtenha a descrição do pod.

   ```
   kubectl describe po minio-ibm-minio-objectstore-0
   ```

   O código a seguir é uma saída de amostra:

   ```
   Name:               minio-ibm-minio-objectstore-0
   Namespace:          default
   Priority:           0
   PriorityClassName:  <none>
   Node:               <none>
   Labels:             app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-1.6.0
                      controller-revision-hash=minio-ibm-minio-objectstore-7b77fd5658
                      heritage=Tiller
                      release=minio
                      statefulset.kubernetes.io/pod-name=minio-ibm-minio-objectstore-0

   ...

   Volumes:
    export:
      Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same
   namespace)
      ClaimName:  export-minio-ibm-minio-objectstore-0
      ReadOnly:   false
   ...

   Events:
    Type     Reason            Age                 From               Message
    ----     ------            ----                ----               -------
    Warning  FailedScheduling  14s (x25 over 57s)  default-scheduler  pod has unbound
   PersistentVolumeClaims (repeated 5 times)
   ```

   A saída indica que os PVCs estão desvinculados.

3. Descreva o PVC.

   ```
   kubectl describe pvc export-minio-ibm-minio-objectstore-0
   ```

   O código a seguir é uma saída de amostra:

   ```
   Name:          export-minio-ibm-minio-objectstore-0
   Namespace:     default
   StorageClass:  standard
   Status:        Pending
   Volume:
   Labels:        app=ibm-minio-objectstore
                 release=minio
   Annotations:   <none>
   Finalizers:    [kubernetes.io/pvc-protection]
   Capacity:
   ```

```
Access Modes:
Events:
 Type     Reason              Age               From                         Message
 ----     ------              ----              ----                         -------
 Warning  ProvisioningFailed  8s (x19 over 4m)  persistentvolume-controller
storageclass.storage.k8s.io "standard" not found
```

A saída indica que o volume de persistência está tentando ligar por meio da classe de armazenamento denominada `standard`. Verifique se a classe de armazenamento existe em seu cluster.

```
kubectl get sc standard
```

O código a seguir é uma saída de amostra:

```
Nenhum recurso localizado.
Error from server (NotFound): storageclasses.storage.k8s.io "standard" not found
```

A saída indica que a classe de armazenamento não existe.

Resolva o problema

Para resolver o problema, conclua as etapas a seguir:

1. Instale um armazenamento de bloco adequado, como o GlusterFS ou o Ceph, em seu cluster.
2. Assegure-se de que o armazenamento de bloco tenha uma classe de armazenamento.
3. Inclua a classe de armazenamento na configuração do gráfico Helm.
4. Implemente o gráfico de Helm.

O Minio no modo distribuído não é acessível ao fornecer um certificado TLS

Quando o servidor Minio é conectado por meio do Minio Client ou de qualquer cliente compatível com o S3, o erro "Servidor não inicializado" é exibido.

Reúna informações sobre o problema

1. Acesse o contêiner.

   ```
   kubectl exec -it mc2 sh
   ```

   O código a seguir é uma saída de amostra:

   ```
   / # mc config host add myminio https://minio-ibm-minio-objectstore:9000 admin ad
   min1234 S3v4 --insecure
   mc: <ERROR> Unable to initialize new config from the provided credentials. Server not
   initialized,
   please try again.
   / #
   ```

2. Verifique o status do pod.

   ```
   kubectl get po
   ```

   O código a seguir é uma saída de amostra:

   ```
   NAME                                 READY     STATUS    RESTARTS   AGE
   mc2                                  1/1       Running   52         2d
   minio-ibm-minio-objectstore-0        1/1       Running   0          4m
   minio-ibm-minio-objectstore-1        1/1       Running   0          4m
   minio-ibm-minio-objectstore-2        1/1       Running   0          4m
   minio-ibm-minio-objectstore-3        1/1       Running   0          4m
   ```

   A saída indica que todos os pods estão em execução.

3. Verifique os logs do pod.

   ```
   kubectl logs minio-ibm-minio-objectstore-0
   ```

O código a seguir é uma saída de amostra:

```
You are running an older version of Minio released 3 weeks ago
Update: https://docs.minio.io/docs/deploy-minio-on-kubernetes

Waiting for a minimum of 2 disks to come online (elapsed 0s)

Waiting for a minimum of 2 disks to come online (elapsed 1s)

Waiting for a minimum of 2 disks to come online (elapsed 2s)

Waiting for a minimum of 2 disks to come online (elapsed 7s)
...
```

A saída indica que as réplicas do servidor não são capazes de se comunicar entre si. O problema pode estar relacionado ao certificado TLS.

Resolva o problema

Assegure-se de gerar o certificado TLS para servidores Minio para o nome comum (CN) no formato a seguir:

```
"/CN=*.<chart deployment name>-ibm-minio-objectstore.<namespace>.svc.<cluster domain name>"
```

Esta etapa é um requisito para servidores Minio que são configurados com o certificado TLS.

O exemplo a seguir possui as etapas para gerar um certificado para implementação do Minio:

- Nome da liberação do Helm: minio
- Namespace para implementação: default
- Nome do domínio do cluster: cluster.local

```
openssl genrsa -out private.key 2048
openssl req -new -x509 -days 3650 -key private.key -out public.crt -subj  "/CN=*.minio-ibm-minio-
objectstore.default.svc.cluster.local"
cp public.crt ca.crt
kubectl create secret generic tls-ssl-minio --from-file=./private.key --from-file=./public.crt --
from-file=./ca.crt
```

Nota: o certificado é gerado para uma combinação especificada de nome da liberação, namespace e nome do domínio do cluster do Helm. O certificado não funcionará se qualquer um desses valores for diferente. Deve-se criar um certificado diferente para qualquer outra combinação de valores.

Os depósitos e objetos Minio estão intermitentemente inacessíveis

Quando o Minio é implementado no modo NAS e você se conecta a partir de um cliente, os depósitos e objetos criados ficam intermitentemente inacessíveis.

Sintomas

1. Crie um armazenamento de objeto Minio.

   ```
   mc config host add myminio http://minio-nas-ibm-minio-objectstore:9000 admin admin1234 S3v4
   ```

   O seguinte é uma saída de amostra:

   ```
   `myminio` incluído com sucesso.
   ```

2. Crie um depósito.

   ```
   mc mb myminio/test
   ```

   O seguinte é uma saída de amostra:

   ```
   Depósito `myminio/test` criado com sucesso.
   ```

Ao executar o comando MinIO Client (mc) para listar os depósitos e objetos, talvez você veja o erro a seguir:

```
mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
```

```
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
```

Para identificar a causa, descreva um pod Minio e verifique se ele está usando o volume persistente desejado.

1. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

2. Obtenha o nome do pod Minio.

   ```
   kubectl get pods | grep minio
   ```

   O seguinte é uma saída de amostra:

   ```
   minio-nas-ibm-minio-objectstore-974f85dc9-7mlrv   1/1       Running   0          4m
   ```

3. Obtenha as informações do pod Minio.

   ```
   kubectl describe po minio-nas-ibm-minio-objectstore-974f85dc9-7mlrv
   ```

   {codeblock}

   O seguinte é uma saída de amostra:

   ```
   Name:               minio-nas-ibm-minio-objectstore-974f85dc9-7mlrv
   Namespace:          default
   Priority:           0
   PriorityClassName:  <none>
   Node:               10.41.14.22/10.41.14.22
   Start Time:         Thu, 25 Apr 2019 22:13:12 -0700
   Labels:             app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-2.4.7
                      heritage=Tiller
                      pod-template-hash=974f85dc9
                      release=minio-nas
   Annotations:        kubernetes.io/psp: 00-rook-ceph-operator
                      productID: Minio_RELEASE.2019-04-09T01-22-30Z_free_00000
                      productName: Minio
                      productVersion: RELEASE.2019-04-09T01-22-30Z
                      scheduler.alpha.kubernetes.io/critical-pod:
   Status:             Running
   IP:                 10.1.106.199
   Controlled By:      ReplicaSet/minio-nas-ibm-minio-objectstore-974f85dc9
   Containers:
    ibm-minio-objectstore:
      Container ID:  docker://33a355387f1df4db9e932cf2921f095cac61baf174522eabf779db2d6c16a779
      Image:         minio/minio:RELEASE.2019-04-09T01-22-30Z
      Image ID:      docker-
   pullable://minio/minio@sha256:b363f54fc5a64d259d760106ad02c8725999c935f7aeae5348abfc0bed3fef0d
      Port:          9000/TCP
      Host Port:     0/TCP
      Command:
        /bin/sh
        -ce
        /usr/bin/docker-entrypoint.sh minio -C /root/.minio/ gateway nas /export
      State:          Running
        Started:      Thu, 25 Apr 2019 22:13:34 -0700
      Ready:          True
      Restart Count:  0
      Requests:
        cpu:      250m
        memory:   256Mi
      Liveness:   http-get http://:service/minio/health/live delay=5s timeout=1s period=30s
   #success=1 #failure=3
      Readiness:  http-get http://:service/minio/health/ready delay=5s timeout=1s period=15s
   #success=1 #failure=3
   ```

```
    Environment:
      MINIO_ACCESS_KEY:  <set to the key 'accesskey' in secret 'minio'>  Optional: false
      MINIO_SECRET_KEY:  <set to the key 'secretkey' in secret 'minio'>  Optional: false
      MINIO_BROWSER:     on
    Mounts:
      /root/.minio/ from minio-config-dir (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-vccnm (ro)
Conditions:
 Type              Status
 Initialized       True
 Ready             True
 ContainersReady   True
 PodScheduled      True
Volumes:
 export:
   Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
   Medium:
 minio-user:
   Type:        Secret (a volume populated by a Secret)
   SecretName:  minio
   Optional:    false
 minio-config-dir:
   Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
   Medium:
 default-token-vccnm:
   Type:        Secret (a volume populated by a Secret)
   SecretName:  default-token-vccnm
   Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     CriticalAddonsOnly
                 node.kubernetes.io/memory-pressure:NoSchedule
                 node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
 Type    Reason     Age   From                 Message
 ----    ------     ----  ----                 -------
 Normal  Scheduled  13m   default-scheduler    Successfully assigned default/minio-nas-ibm-
minio-objectstore-974f85dc9-7mlrv to 10.41.14.22
 Normal  Pulling    13m   kubelet, 10.41.14.22  pulling image "minio/minio:RELEASE.2019-04-
09T01-22-30Z"
 Normal  Pulled     12m   kubelet, 10.41.14.22  Successfully pulled image
"minio/minio:RELEASE.2019-04-09T01-22-30Z"
 Normal  Created    12m   kubelet, 10.41.14.22  Created container
```

A saída de comando mostra que os pods Minio estão usando o seguinte volume:

```
Volumes:
  export:
    Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
```

Os depósitos e objetos ficam intermitentemente inacessíveis pelos seguintes motivos:

- O volume Persistente não corresponde ao armazenamento ReadWriteMany.
- O Persistente não está rotulado corretamente.

Resolvendo o problema

1. Desinstale a implementação com falha.
2. Use um volume persistente correspondente a uma tecnologia de armazenamento que suporte o volume ReadWriteMany.
3. Ligue a solicitação de volume persistente (PVC) a um volume persistente (PV) específico. Para assegurar que ocorra a ligação, rotule o PV que você precisa ligar como `\"pv: <pv name>\"`. Consulte o seguinte exemplo:

   ```
   apiVersion: v1
   kind: PersistentVolume
   metadata:
    labels:
      pv: shared-pv
    name: shared-pv
   ```

Opções de armazenamento hospedadas fora do IBM Cloud Private

As opções de armazenamento que você configura fora de seu cluster do IBM® Cloud Private.

Essas tecnologias de armazenamento não são hospedadas em nós do IBM Cloud Private. Essas tecnologias são configuradas por um administrador fora do seu ambiente do IBM Cloud Private. Depois que essas tecnologias são integradas com o IBM Cloud Private, as cargas de trabalho do aplicativo podem usar os volumes criando volumes persistentes e solicitações de volume persistente.

O monitoramento e o gerenciamento dessas tecnologias de armazenamento são feitos pelo administrador de armazenamento e o IBM Cloud Private não fornece nenhum suporte nesse sentido.

- vSphere Cloud Provider
- hostPath
- Network File System
- IBM Spectrum Scale
- Ceph RBD Externo

vSphere Cloud Provider

Configure um vSphere Cloud Provider em seu cluster do IBM® Cloud Private.

Visão geral

O VMware vSphere possui uma plataforma comprovada do Software Defined Storage (SDS) que se integra com as ofertas de bloco, arquivo e Hiperconvergentes, como a rede de área de armazenamento virtual do VMware (vSAN). Essas ofertas de armazenamento podem ser expostas como Virtual Machine File System (VMFS), Network File System (NFS), Volumes Virtuais (VVol) e armazéns de dados vSAN. Um armazenamento de dados é uma abstração que oculta os detalhes do armazenamento e fornece uma interface uniforme para armazenamento de dados persistentes. Dependendo do armazenamento de backend usado, os armazenamentos de dados podem ser do tipo vSAN, VMFS, NFS e VVol.

- O vSAN é um armazenamento de infraestrutura hiperconvergida que oferece excelente desempenho e confiabilidade. A vantagem da vSAN é o gerenciamento de armazenamento simplificado com recursos como administração orientada por política.
- O VMFS é um sistema de arquivos de cluster que permite a virtualização para escalar além de um único nó para múltiplos servidores VMware ESX.
- O NFS é um protocolo de arquivo distribuído que é usado para acessar o armazenamento por meio de uma rede, assim como o armazenamento local. O vSphere suporta o NFS como um backend para armazenar arquivos de máquinas virtuais.

Para obter mais informações, consulte vSphere Cloud Provider.

Com a interface do provedor em nuvem do Kubernetes, é possível integrar e oferecer armazenamento do vSphere para pods de carga de trabalho do aplicativo.

Pré-requisitos e limitações

Consulte Pré-requisitos e limitações.

Configurando o vSphere Cloud Provider

Se você escolher o vSphere como a infraestrutura de nuvem para seu cluster do IBM Cloud Private, será possível configurar o vSphere Cloud Provider em seu cluster para oferecer o volume persistente para a carga de trabalho do aplicativo.

- Para configurar o vSphere Cloud Provider durante a instalação do IBM Cloud Private, consulte Configurando um vSphere Cloud Provider durante a instalação do IBM Cloud Private.
- Para configurar o vSphere Cloud Provider após a instalação do IBM Cloud Private, consulte Configurando um vSphere Cloud Provider após a instalação do IBM Cloud Private. **Nota:** Se desejar configurar um vSphere Cloud Provider depois de instalar o cluster do IBM® Cloud Private, certifique-se de configurar `kubelet_nodename: hostname` no arquivo `<installation_directory>/cluster/config.yaml` durante a instalação do IBM® Cloud Private.

Criando uma Classe de Armazenamento

Para provisionar dinamicamente um volume persistente, é necessário criar uma classe de armazenamento com o vsphere-volume como o fornecedor. Se você tiver diferentes tipos de armazenamento de dados e desejar provisionar um volume a partir de qualquer um desses armazenamentos de dados, será necessário criar uma classe de armazenamento separada para cada tipo de armazenamento de dados.

Para obter mais informações sobre como criar uma classe de armazenamento para o vSphere, consulte Criando uma classe de armazenamento para o vSphere.

Verificando a Configuração

É possível verificar se a configuração está correta implementando um aplicativo que requer armazenamento persistente. Implemente o aplicativo a partir de seu catálogo do cluster do IBM Cloud Private. Por exemplo, implemente o aplicativo `ibm-postgres-dev`. Consulte PostgreSQL.

Gerenciando seu cluster

Gerencie as operações relacionadas ao vSphere após a instalação do cluster. Para obter mais informações, consulte Gerenciando seu cluster.

Pré-requisitos e Limitações

Pré-requisitos e limitações para a configuração de um vSphere Cloud Provider no cluster do IBM® Cloud Private.

Pré-requisitos

Certifique-se de que os nós em seu cluster atendam a esses requisitos:

- Todos os nós principais do IBM Cloud Private devem ser capazes de acessar o vCenter.
- O nome do host do nó deve ser igual ao nome da VM.
- Os nomes do host do nó devem estar em conformidade com o regex, `[a-z](([-0-9a-z]+)?[0-9a-z])?(\.[a-z0-9](([-0-9a-z]+)?[0-9a-z])?)*` e também deve estar em conformidade com as seguintes restrições:
  - Eles não devem iniciar com números.
  - Eles não devem usar letras maiúsculas.
  - Eles não devem ter nenhum caractere especial, exceto `.` e `-`.
  - Eles devem conter pelo menos três caracteres, mas no máximo 63 caracteres.
- O UUID do disco nas VMs do nó deve estar ativado: o valor `disk.EnableUUID` deve ser configurado como `True`.
- O usuário que é especificado na configuração de nuvem do vSphere deve ter privilégios para interagir com o vCenter.

Tabela 1. Usuário do vSphere Cloud Provider

Funções	Privilégios	Entidades	Propagar para o filho

Funções	Privilégios	Entidades	Propagar para o filho
manage-k8s-node-vms	• Resource.AssignVMToPool • System.Anonymous • System.Read • System.View • VirtualMachine.Config.AddExistingDisk • VirtualMachine.Config.AddNewDisk • VirtualMachine.Config.AddRemoveDevice • VirtualMachine.Config.RemoveDisk • VirtualMachine.Inventory.Create • VirtualMachine.Inventory.Delete	• Grupo • Hosts • Pasta da VM	SIM
manage-k8s-volumes	• Datastore.AllocateSpace • Datastore.FileManagement • System.Anonymous • System.Read • System.View	Armazenamento de Dados	Não
k8s-system-read-and-spbm-profile-view	• StorageProfile.View • System.Anonymous • System.Read • System.View	vCenter	Não
ReadOnly	• System.Anonymous • System.Read • System.View	• Datacenter • Cluster do Armazenamento de Dados • Pasta de Armazenamento do Armazenamento de Dados	Não

Origem: https://kubernetes.io/docs/getting-started-guides/vsphere/

Limitações

- O vSphere Cloud Provider suporta somente o modo de acesso ReadWriteOnce para volumes persistentes.
- Se um nó estiver inativo, todos os pods, juntamente com os volumes montados, são movidos para outro nó. No entanto, o nó antigo não é reativado porque os volumes não são desmontados automaticamente. Você deve desmontar manualmente os volumes do nó antigo para que ele seja reativado.

Cenários de Implementação

O provedor em nuvem vSphere pode ser configurado em nós IBM® Cloud Private que estão em um único vCenter, em vários datacenters ou em vários vCenters.

Assegure-se de atender aos pré-requisitos. Para obter mais informações, consulte Pré-requisitos e limitações.

Nós do cluster em um único vCenter IBM Cloud Private

Para os nós do cluster do IBM Cloud Private que estão no mesmo data center em um único vCenter, é possível configurar o provedor em nuvem vSphere durante a instalação do IBM Cloud Private. Deve-se atualizar o arquivo `<installation_directory>/cluster/config.yaml` ou usar um arquivo de configuração customizado que tenha os parâmetros de configuração do vSphere.

Para obter mais informações, consulte Configurando um vSphere Cloud Provider durante a instalação do IBM Cloud Private.

A seguir está uma configuração de amostra do arquivo de configuração customizada `<installation_directory>/cluster/misc/cloud_provider/vsphere.conf`:

```
[Global]
user = "administrator@vsphere.local"
password = "xxxxxxx"
port = "443"
insecure-flag = "1"
datacenters = "datacenter1"

[VirtualCenter "1.1.1.1"]

[Workspace]
server = "1.1.1.1"
datacenter = "datacenter1"
default-datastore="datastore1"
folder = "kubernetes"

[Disk]
scsicontrollertype = pvscsi
```

Nós do cluster do IBM Cloud Private localizados em vários data centers em um único vCenter

Para os nós do cluster do IBM Cloud Private que estão localizados em vários data centers em um único vCenter, é possível configurar o vSphere Cloud Provider durante a instalação do IBM Cloud Private. No entanto, deve-se usar um arquivo de configuração customizado que tenha os parâmetros de configuração do vSphere.

Nota: assegure-se de que todos os nós do cluster do IBM Cloud Private tenham acesso ao armazenamento de dados compartilhado.

Para obter mais informações sobre como usar um arquivo customizado para a configuração do vSphere Cloud Provider, consulte Usar um arquivo de configuração customizado para a configuração do vSphere Cloud Provider.

Nota: assegure-se de incluir todos os nomes de data center no parâmetro `datacenters`.

A seguir está uma configuração de amostra do arquivo de configuração customizada `<installation_directory>/cluster/misc/cloud_provider/vsphere.conf`:

```
[Global]
user = "administrator@vsphere.local"
password = "xxxxxxx"
port = "443"
insecure-flag = "1"
datacenters = "datacenter1, datacenter2"

[VirtualCenter "1.1.1.1"]

[Workspace]
server = "1.1.1.1"
datacenter = "datacenter1"
default-datastore="datastore1"
folder = "kubernetes"

[Disk]
scsicontrollertype = pvscsi
```

Nós do cluster do IBM Cloud Private localizados em múltiplos

vCenters

Para os nós do cluster do IBM Cloud Private que estão localizados em vários data centers em um único vCenter, é possível configurar o vSphere Cloud Provider durante a instalação do IBM Cloud Private. No entanto, deve-se usar um arquivo de configuração customizado que tenha os parâmetros de configuração do vSphere.

Nota: assegure-se de que todos os nós do cluster do IBM Cloud Private tenham acesso ao armazenamento de dados compartilhado.

Para obter mais informações sobre como usar um arquivo customizado para a configuração do vSphere Cloud Provider, consulte Usar um arquivo de configuração customizado para a configuração do vSphere Cloud Provider.

Assegure-se de incluir as informações a seguir no arquivo de configuração customizado:

- Inclua todos os nomes do data center no parâmetro `datacenters`.
- Inclua o parâmetro `[VirtualCenter "<IP address>"]` para todos os vCenters que têm nós do cluster do IBM Cloud Private.
- Inclua o nome do data center no parâmetro `datacenters` para cada vCenter.

A seguir está uma configuração de amostra do arquivo de configuração customizada `<installation_directory>/cluster/misc/cloud_provider/vsphere.conf`:

```
[Global]
user = "administrator@vsphere.local"
password = "xxxxxxx"
port = "443"
insecure-flag = "1"
datacenters = "datacenter1, datacenter2"

[VirtualCenter "1.1.1.1"]
datacenters = "datacenter1"

[VirtualCenter "2.2.2.2"]
datacenters = "datacenter2"

[Workspace]
server = "1.1.1.1"
datacenter = "datacenter1"
default-datastore="datastore1"
folder = "kubernetes"

[Disk]
scsicontrollertype = pvscsi
```

Configurando um vSphere Cloud Provider

Configure um vSphere Cloud Provider em seu cluster do IBM® Cloud Private.

- Configurando um vSphere Cloud Provider durante a instalação do IBM Cloud Private
- Configurando um vSphere Cloud Provider após a instalação do IBM Cloud Private

Configurando um vSphere Cloud Provider durante a instalação do IBM Cloud Private

Configure um vSphere Cloud Provider durante a instalação do cluster do IBM® Cloud Private.

É possível configurar a nuvem do vSphere atualizando o arquivo `<installation_directory>/cluster/config.yaml` ou usando um arquivo de configuração customizado que tenha os parâmetros de configuração do vSphere.

Antes de iniciar, certifique-se de estar em conformidade com os pré-requisitos. Para obter mais informações, consulte Pré-requisitos e limitações.

Use o arquivo config.yaml para a configuração do vSphere Cloud Provider

Inclua informações do vSphere no arquivo `config.yaml`.

```
kubelet_nodename: hostname
 cloud_provider: vsphere
 vsphere_conf:
    user: "<vCenter username for vSphere Cloud Provider>"
    password: "<password for vCenter user>"
    server: <vCenter server IP or FQDN>
    port: [vCenter Server Port; default: 443]
  insecure_flag: [set to 1 if vCenter uses a self-signed certificate]
  datacenter: <datacenter name on which Node VMs are deployed>
  datastore: <default datastore to be used for provisioning volumes>
  storageclass:
    name: vsphere
```

```
create: true
isdefault: false
provisionerparams:
  diskformat: thin
  fstype: ext3
```

A seguir estão as descrições de parâmetros:

- `kubelet_nodename`: deve ser configurado como `hostname`.
- `cloud_provider`: deve ser configurado como `vsphere`.
- `user`: nome do usuário para o vCenter. Todas as operações do vCenter são executadas usando as credenciais desse usuário.
- `password`: a senha do usuário que está especificado no parâmetro `user`.
- `server`: endereço IP ou nome completo do domínio (FQDN) do servidor vCenter.
- `port`: número da porta que é usado para configuração do vCenter. O valor padrão é `443`.
- `insecure_flag`: configure como `1` se o vCenter usar um certificado autoassinado.
- `datacenter`: nome do data center no qual as VMs do nó são implementadas.
- `datastore`: nome do armazenamento de dados que é usado para colocar os PersistentVolumes que são criados usando uma classe de armazenamento. Especifique apenas o nome, mesmo se o armazenamento de dados estiver localizado em uma pasta ou for um membro de um cluster de armazenamento de dados. O nome do armazenamento de dados faz distinção entre maiúsculas e minúsculas.
- `storageclass.name`: Nome da classe de armazenamento. O valor padrão é `vsphere`.
- `storageclass.create` - Um valor booleano. Se configurado como `true`, a classe de armazenamento será criada com os parâmetros do fornecedor especificados.
- `storageclass.isdefault` - Um valor booleano. Se configurado como `true`, essa classe de armazenamento será marcada como a classe de armazenamento padrão.
- `storageclass.provisionerparams` - Use esta opção para configurar parâmetros adicionais do fornecedor. Se você não configurou nenhum `armazenamento de dados` como um parâmetro do fornecedor, o `armazenamento de dados` padrão será usado.
  Para obter informações adicionais sobre a classe de armazenamento para vSphere, consulte Criando uma classe de armazenamento para o volume do vSphere.

Usar um arquivo de configuração customizado para a configuração do vSphere Cloud Provider

É possível usar um arquivo de configuração customizado que tenha parâmetros de configuração do vSphere. Deve-se nomear este arquivo como `vsphere.conf` e colocá-lo na pasta `<installation_directory>/cluster/misc/cloud_provider/`.

Se a pasta `cloud_provider` não existir, crie-a no caminho `<installation_directory>/cluster/misc/` e, em seguida, coloque o arquivo customizado na pasta.

Não inclua a seção `vsphere_conf:` no arquivo `config.yaml`. Em vez disso, inclua a configuração a seguir no arquivo `config.yaml`:

```
kubelet_nodename: hostname
cloud_provider: vsphere
```

Para obter mais informações sobre como criar um arquivo de configuração customizado, consulte Arquivo de configuração de nuvem do vSphere para Kubernetes versão 1.9.x e superior.

Para obter os arquivos de configuração de amostra, consulte Cenários de implementação.

O que fazer a seguir

Continue com a instalação do IBM Cloud Private.

Configurando um vSphere Cloud Provider após a instalação do IBM Cloud Private

Configure um vSphere Cloud Provider depois de instalar o cluster do IBM® Cloud Private.

Nota: é possível configurar um vSphere Cloud Provider após instalar seu cluster do IBM® Cloud Private apenas quando `kubelet_nodename: hostname` é configurado durante a instalação do IBM® Cloud Private

Antes de iniciar, certifique-se de estar em conformidade com os pré-requisitos. Para obter mais informações, consulte Pré-requisitos e limitações

Em seguida, conclua as etapas nas seções a seguir.

1. Atualize todos os nós do cluster, exceto os nós principais.
2. Atualize todos os nós principais.
3. Remova os nós antigos.
4. Incluindo informações do vSphere no arquivo `config.yaml`

Nota: conclua com cuidado as etapas de configuração. Uma configuração incorreta pode fazer com que os nós principais parem de funcionar. Os nós principais podem mostrar um estado `notReady`. Para obter mais informações sobre esse problema, consulte O nó principal entra no estado "notReady" depois de configurar o provedor em nuvem vSphere.

Atualizar todos os nós do cluster, exceto os nós principais

Conclua estas etapas em todos os nós, exceto nos nós principais.

1. Inclua a parte de código a seguir no arquivo `/etc/systemd/system/kubelet.service` sob a seção `[Service]`.

   ```
   --cloud-provider=vsphere
   ```

   Após incluir o código, o conteúdo do arquivo é semelhante ao texto a seguir:

   ```
   [Unit]
   Description=Kubelet Service
   Documentation=https://github.com/kubernetes/kubernetes
   [Service]
   EnvironmentFile=-/etc/environment
   ExecStart=/opt/kubernetes/hyperkube kubelet \
   ...
    --cloud-provider=vsphere \
   ...
   ```

2. Recarregue o arquivo de unidade systemd do kubelet.

   ```
   systemctl daemon-reload
   ```

3. Reinicie o serviço kubelet.

   ```
   Systemctl restart kubelet.service
   ```

4. Ao reiniciar o serviço, você pode perder qualquer rótulo ou contaminação incluído manualmente no nó. Se sim, é possível incluí-lo novamente agora.

Atualize todos os nós principais

Conclua estas etapas em todos os nós principais em seu cluster do IBM Cloud Private.

1. Crie o arquivo de configuração do vSphere Cloud Provider com os seguintes parâmetros e salve-o como um arquivo `<name>.conf`. Copie o arquivo para o local `/etc/cfc/conf/`. Esse local é um diretório compartilhado que é acessível pelo contêiner kubelet, pod de gerenciador do controlador e pod de servidor de API.

   ```
   [Global]
   user = <vCenter username for vSphere Cloud Provider>
   password = <password for vCenter user>
   port = [vCenter server port; default: 443]
   insecure-flag = [set to 1 if vCenter uses a self-signed certificate]
   datacenters = <datacenter name on which node VMs are deployed>

   [VirtualCenter <vCenter server IP address or FQDN>]

   [Workspace]
   server = <vCenter server IP address or FQDN>
   datacenter = <datacenter name on which node VMs are deployed>
   default-datastore= <default datastore to be used for provisioning volumes>
   folder = "kubernetes"

   [Disk]
   scsicontrollertype = pvscsi
   ```

   Em que:

- `user`: nome do usuário para o vCenter. Todas as operações do vCenter são executadas usando as credenciais desse usuário.
- `password`: a senha do usuário que está especificado no parâmetro `user`.
- `server`: endereço IP ou nome completo do domínio (FQDN) do servidor vCenter.
- `port`: número da porta que é usado para configuração do vCenter. O valor padrão é `443`.
- `insecure-flag`: Configurado como `1` se o vCenter usar um certificado autoassinado.
- `datacenter`: nome do data center no qual as VMs do nó são implementadas. É possível incluir vários nomes que são separados por vírgula.
- `default-datastore`: Nome do armazenamento de dados que é usado para localizar os PersistentVolumes que são criados usando uma classe de armazenamento. Especifique apenas o nome, mesmo se o armazenamento de dados estiver localizado em uma pasta ou for um membro de um cluster de armazenamento de dados. O nome do armazenamento de dados faz distinção entre maiúsculas e minúsculas.
- `scsicontrollertype` é o tipo de controlador Small Computer System Interface (SCSI) usado para acessar o disco virtual.

2. Inclua a parte de código a seguir no kubelet e os arquivos manifest do gerenciador de controlador e do servidor de API.

```
--cloud-provider=vsphere
--cloud-config=<full-path-to-the-vsphere.conf-file>
```

   - Inclua o código no arquivo `/etc/systemd/system/kubelet.service` sob a seção `[Service]`.

     Após incluir o código, o conteúdo do arquivo é semelhante ao texto a seguir:

```
 [Unit] Description=Kubelet Service
Documentation=https://github.com/kubernetes/kubernetes

 [Service] EnvironmentFile=-/etc/environment ExecStart=/opt/kubernetes/hyperkube kubelet
\
 ...
  --cloud-provider=vsphere \
--cloud-config=/etc/cfc/conf/vsphere.conf \
 ...
```

   - Inclua o código no arquivo `/etc/cfc/pods/master.json` sob as seções `"spec">"containers">"name": "controller-manager">"command"` e `"spec">"containers">"name": "apiserver">"command"`.

     Após incluir o código, o conteúdo do arquivo é semelhante ao texto a seguir:

```
  "spec":{
      "hostNetwork": true,
      "containers":[
        {
          "name": "controller-manager",
          "image": "registry.ng.bluemix.net/mdelder/hyperkube:v1.11.0",
          "imagePullPolicy": "IfNotPresent",
          "command": [
            "/hyperkube",
            "controller-manager",
            "--master=https://127.0.0.1:8001",
            "--service-account-private-key-file=/etc/cfc/conf/server.key",
            "--cloud-provider=vsphere",
            "--cloud-config=/etc/cfc/conf/vsphere.conf",
            .
            .
            .
            "name": "apiserver",
            "image": "registry.ng.bluemix.net/mdelder/hyperkube:v1.11.0",
            "imagePullPolicy": "IfNotPresent",
            "command": [
              "/hyperkube",
              "apiserver",
              "--secure-port=8001",
              "--bind-address=0.0.0.0",
              "--advertise-address=10.10.25.206",
              "--cloud-provider=vsphere",
              "--cloud-config=/etc/cfc/conf/vsphere.conf",
              .
              .
              .
```

3. Recarregue o arquivo de unidade systemd do kubelet.

```
systemctl daemon-reload
```

4. Reinicie o serviço kubelet.

```
Systemctl restart kubelet.service
```

5. Ao reiniciar o serviço, você pode perder qualquer rótulo ou contaminação incluído manualmente no nó. Se sim, é possível incluí-lo novamente agora.

Em seguida, conclua as etapas em Remover os nós antigos.

Remova os nós antigos

Conclua estas etapas somente caso o `kubelet_nodename: hostname` não tenha sido configurado no arquivo `config.yaml` durante a instalação.

Exclua os nós antigos que possuem um endereço IP. Execute estes comandos de qualquer nó em seu cluster ou de uma estação de trabalho remota.

1. Assegure-se de que a CLI do kubectl esteja configurada. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Remova os nós antigos.

   a. Obtenha informações sobre os nós.

   ```
   kubectl get nodes
   ```

   A saída se assemelha ao texto a seguir:

   ```
   NAME            STATUS     ROLES     AGE       VERSION
   10.10.25.6      NotReady   <none>    7d        v1.11.0
   10.10.25.3      NotReady   <none>    7d        v1.11.0
   master          Ready      <none>    55s       v1.11.0
   worker1         Ready      <none>    45s       v1.11.0
   ```

   b. Remova os nós que possuem um endereço IP sob o campo `Name`.

   ```
   kubectl delete node <IP-address>
   ```

   O seguinte é um comando e uma saída de amostra:

   ```
   kubectl delete node 10.10.25.3
   node "10.10.25.3" deleted
   ```

   c. Verifique se o nó foi excluído.

   ```
   kubectl get nodes
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME            STATUS     ROLES     AGE       VERSION
   10.10.25.6      NotReady   <none>    7d        v1.11.0
   master          Ready      <none>    7m        v1.11.0
   worker1         Ready      <none>    7m        v1.11.0
   ```

Incluindo informações do vSphere no arquivo `config.yaml`

Conclua estas etapas para atualizar o arquivo `config.yaml`:

1. Abra o arquivo `config.yaml` na pasta `/<installation_directory>/cluster`.
2. No arquivo `config.yaml`, inclua estas linhas de código:

```
kubelet_nodename: hostname
cloud_provider: vsphere
vsphere_conf:
   user: "<vCenter username for vSphere Cloud Provider>"
   password: "<password for vCenter user>"
   server: <vCenter server IP or FQDN>
   port: [vCenter Server Port; default: 443]
   insecure_flag: [set to 1 if vCenter uses a self-signed certificate]
```

```
datacenter: <datacenter name on which Node VMs are deployed>
datastore: <default datastore to be used for provisioning volumes>
```

A seguir estão as descrições de parâmetros:

- `kubelet_nodename`: deve ser configurado como `hostname`.
- `cloud_provider`: deve ser configurado como `vsphere`.
- `user`: nome do usuário para o vCenter. Todas as operações do vCenter são executadas usando as credenciais desse usuário.
- `password`: a senha do usuário que está especificado no parâmetro `user`.
- `server`: endereço IP ou nome completo do domínio (FQDN) do servidor vCenter.
- `port`: número da porta que é usado para configuração do vCenter. O valor padrão é `443`.
- `insecure_flag`: configure como `1` se o vCenter usar um certificado autoassinado.
- `datacenter`: nome do data center no qual as VMs do nó são implementadas.
- `datastore`: nome do armazenamento de dados que é usado para colocar os PersistentVolumes que são criados usando uma classe de armazenamento. Especifique apenas o nome, mesmo se o armazenamento de dados estiver localizado em uma pasta ou for um membro de um cluster de armazenamento de dados. O nome do armazenamento de dados faz distinção entre maiúsculas e minúsculas.

O que fazer a seguir

Crie uma classe de armazenamento para usar o armazenamento do vSphere. Para obter mais informações, consulte Criando uma classe de armazenamento para o volume vSphere .

Criando uma classe de armazenamento para o volume do vSphere

Crie uma classe de armazenamento para fornecer PersistentVolume em um armazenamento de dados vSphere.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Pré-requisito

o IBM® Cloud Private deve ser configurado com um vSphere Cloud Provider. Consulte Configurando um vSphere Cloud Provider.

Crie uma classe de armazenamento

Para criar uma classe de armazenamento para um volume do vSphere, especifique os valores de campo a seguir na definição de classe de armazenamento:

- metadados:
  - `name`: nome do objeto da classe de armazenamento.
- parâmetros:
  - `diskformat`: `thin`, `zeroedthick` ou `eagerzeroedthick`.
  - `datastore`: nome do armazenamento de dados. O volume é criado no armazenamento de dados que é especificado na classe de armazenamento.
  - `storagePolicyName`: nome da política de armazenamento que você criou em sua configuração do vSphere. Um volume persistente (PV) é provisionado dinamicamente com base na política de armazenamento. Para obter mais informações sobre o gerenciamento baseado em política de armazenamento (SPBM), consulte Gerenciamento baseado em política de armazenamento para o fornecimento dinâmico de volumes.

A seguir há um exemplo de como criar uma classe de armazenamento:

1. Crie um arquivo `YAML` com as definições de classe de armazenamento:

```
 vim vsphere-volume-storage-class-1.yaml

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
 name: storage-class-1
provisioner: kubernetes.io/vsphere-volume
parameters:
 diskformat: thin
 datastore: datastore-1
 storagePolicyName: vsanStoragePolicy
```

2. Crie a classe de armazenamento:

   ```
   kubectl create -f vsphere-volume-storage-class-1.yaml
   ```

   A saída se assemelha ao código a seguir:

   ```
   storageclass "storage-class-1" created
   ```

3. Verifique se a classe de armazenamento está criada:

   ```
   kubectl describe sc storage-class-1
   ```

   A saída se assemelha ao código a seguir:

   ```
   Name:           storage-class-1
   IsDefaultClass: No
   Annotations:    <none>
   Provisioner:    kubernetes.io/vsphere-volume
   Parameters:     datastore=datastore-1,diskformat=thin,storagePolicyName=vsanStoragePolicy
   Events:         <none>
   ```

vSphere Cloud Provider

Configure um vSphere Cloud Provider em seu cluster do IBM® Cloud Private.

Visão geral

O VMware vSphere possui uma plataforma comprovada do Software Defined Storage (SDS) que se integra com as ofertas de bloco, arquivo e Hiperconvergentes, como a rede de área de armazenamento virtual do VMware (vSAN). Essas ofertas de armazenamento podem ser expostas como Virtual Machine File System (VMFS), Network File System (NFS), Volumes Virtuais (VVol) e armazéns de dados vSAN. Um armazenamento de dados é uma abstração que oculta os detalhes do armazenamento e fornece uma interface uniforme para armazenamento de dados persistentes. Dependendo do armazenamento de backend usado, os armazenamentos de dados podem ser do tipo vSAN, VMFS, NFS e VVol.

- O vSAN é um armazenamento de infraestrutura hiperconvergida que oferece excelente desempenho e confiabilidade. A vantagem da vSAN é o gerenciamento de armazenamento simplificado com recursos como administração orientada por política.
- O VMFS é um sistema de arquivos de cluster que permite a virtualização para escalar além de um único nó para múltiplos servidores VMware ESX.
- O NFS é um protocolo de arquivo distribuído que é usado para acessar o armazenamento por meio de uma rede, assim como o armazenamento local. O vSphere suporta o NFS como um backend para armazenar arquivos de máquinas virtuais.

Para obter mais informações, consulte vSphere Cloud Provider.

Com a interface do provedor em nuvem do Kubernetes, é possível integrar e oferecer armazenamento do vSphere para pods de carga de trabalho do aplicativo.

Pré-requisitos e limitações

Consulte Pré-requisitos e limitações.

Configurando o vSphere Cloud Provider

Se você escolher o vSphere como a infraestrutura de nuvem para seu cluster do IBM Cloud Private, será possível configurar o vSphere Cloud Provider em seu cluster para oferecer o volume persistente para a carga de trabalho do aplicativo.

- Para configurar o vSphere Cloud Provider durante a instalação do IBM Cloud Private, consulte Configurando um vSphere Cloud Provider durante a instalação do IBM Cloud Private.
- Para configurar o vSphere Cloud Provider após a instalação do IBM Cloud Private, consulte Configurando um vSphere Cloud Provider após a instalação do IBM Cloud Private. **Nota:** Se desejar configurar um vSphere Cloud Provider depois de instalar o cluster do IBM® Cloud Private, certifique-se de configurar `kubelet_nodename: hostname` no arquivo `<installation_directory>/cluster/config.yaml` durante a instalação do IBM® Cloud Private.

Criando uma Classe de Armazenamento

Para provisionar dinamicamente um volume persistente, é necessário criar uma classe de armazenamento com o vsphere-volume como o fornecedor. Se você tiver diferentes tipos de armazenamento de dados e desejar provisionar um volume a partir de qualquer um desses armazenamentos de dados, será necessário criar uma classe de armazenamento separada para cada tipo de armazenamento de dados.

Para obter mais informações sobre como criar uma classe de armazenamento para o vSphere, consulte Criando uma classe de armazenamento para o vSphere.

Verificando a Configuração

É possível verificar se a configuração está correta implementando um aplicativo que requer armazenamento persistente. Implemente o aplicativo a partir de seu catálogo do cluster do IBM Cloud Private. Por exemplo, implemente o aplicativo `ibm-postgres-dev`. Consulte PostgreSQL.

Gerenciando seu cluster

Gerencie as operações relacionadas ao vSphere após a instalação do cluster. Para obter mais informações, consulte Gerenciando seu cluster.

- Pré-requisitos e limitações
- Cenários de implementação
- Configurando um vSphere Cloud Provider
- Criando uma classe de armazenamento para o volume do vSphere
- Verificando a configuração
- Gerenciando seu cluster
- Resolução de problemas do vSphere Cloud Provider

Gerenciando seu cluster

Gerenciar operações do vSphere em seu cluster.

As operações a seguir poderão ser executadas após o cluster ser instalado.

Mudando o ID do usuário e a senha do vCenter

Se precisar mudar o ID do usuário ou a senha do servidor vCenter configurado, deverá atualizar o arquivo de configuração do vSphere Cloud Provider. No IBM Cloud Private, esse arquivo de configuração está disponível em `/etc/cfc/conf/vsphere.conf` em todos os nós principais.

Conclua estas etapas para atualizar a configuração em todos os nós principais:

1. Atualize os parâmetros de ID do usuário e senha no arquivo `/etc/cfc/conf/vsphere.conf`.
2. Recarregue o arquivo de unidade systemd do kubelet.

   ```
   systemctl daemon-reload
   ```

3. Reinicie o serviço kubelet.

   ```
   Systemctl restart kubelet.service
   ```

   Nota: ao reiniciar o serviço kubelet, talvez você perca qualquer rótulo ou contaminação que você incluiu manualmente no nó. Se sim, é necessário incluí-lo novamente.

Provisionando o volume a partir de um armazenamento de dados específico

O armazenamento de dados que você fornece quando configura o vSphere Cloud Provider é usado para o armazenamento de dados padrão para volumes de fornecimento. Se você desejar usar algum outro armazenamento de dados ou incluir um armazenamento de dados em seu cluster, será necessário criar uma nova classe de armazenamento e fornecer o nome do armazenamento de dados específico.

A seguir está um exemplo de uma classe de armazenamento que usa `new-datastore` na seção de configuração da classe de armazenamento.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast
provisioner: kubernetes.io/vsphere-volume
parameters:
    diskformat: thin
    datastore: new-datastore
```

Ativando o gerenciamento de armazenamento baseado em política

O vSphere fornece gerenciamento de armazenamento baseado em política. No IBM Cloud Private, é possível especificar a política de armazenamento a ser usada para provisionar o volume persistente a partir de um armazenamento de dados do vSphere. Para obter mais informações, consulte Criando uma classe de armazenamento para o volume vSphere .

As políticas de gerenciamento baseado em política de armazenamento (SPBM) existentes podem ser usadas para configurar um volume persistente usando o parâmetro `storagePolicyName` em uma classe de armazenamento. O exemplo de classe de armazenamento usa o parâmetro `storagePolicyName`.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: storage-class-spbm
provisioner: kubernetes.io/vsphere-volume
parameters:
    diskformat: thin
    datastore: vsan-datastore
    storagePolicyName: vsanStoragePolicy
```

Quando o parâmetro `storagePolicyName` é configurado, um volume persistente (PV) é provisionado dinamicamente com base na política de armazenamento.

Para obter mais informações sobre o SPBM, consulte Gerenciamento de Armazenamento Baseado em Política para fornecimento dinâmico de volumes.

Resolução de problemas do vSphere Cloud Provider

Revise problemas do vSphere Cloud Provider encontrados com frequência.

- Coletar logs
- O nó principal entra no estado "notReady" depois de configurar o provedor em nuvem do vSphere
- Falha no fornecimento de volume persistente

Coletar logs

Colete logs para identificar a causa raiz de um problema.

Para resolução de problemas do vSphere Cloud Provider, deve-se reunir os arquivos de log a seguir:

Antes de iniciar, assegure-se de que a CLI kubectl esteja configurada. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

Log do gerenciador do controlador

1. Obtenha o nome do pod principal do Kubernetes.

   ```
   kubectl -n kube-system get pod | grep k8s-master
   ```

   O seguinte é uma saída de amostra:

   ```
   k8s-master-<nodename> 4/4 Running 3 1d
   ```

2. Obtenha o log do gerenciador do controlador.

   ```
   kubectl -n kube-system logs k8s-master-<nodename> -c controller-manager > controller-manager-
   <nodename>.log
   ```

Log do servidor da API

1. Obtenha o nome do pod principal do Kubernetes.

   ```
   kubectl -n kube-system get pod | grep k8s-master
   ```

   O seguinte é uma saída de amostra:

   ```
   k8s-master-<nodename> 4/4 Running 3 1d
   ```

2. Obtenha o log do servidor de API.

   ```
   kubectl -n kube-system logs k8s-master-<nodename> -c apiserver > apiserver-<nodename>.log
   ```

Log do Kubelet

Em cada nó, execute o comando a seguir para obter o log kubelet a partir desse nó.

```
journalctl -u kubelet > kubelet-< nodename> .log
```

O nó principal entra no estado "notReady" após a configuração do vSphere Cloud Provider

Depois de configurar um provedor em nuvem vSphere, seu cluster do IBM® Cloud Private fica inativo e os nós principais estão no estado "notReady".

Causas

Para configurar um provedor em nuvem, os parâmetros `-- cloud-provider` e `--cloud-config` precisam ser configurados para transmitir o tipo de provedor em nuvem e suas informações de configuração para o kubelet.

Se você fornecer informações de configuração incorretas, o processo `kubelet` não será iniciado.

Resolvendo o problema

- Verifique se você forneceu o nome do usuário e a senha corretos do vCenter.
- Verifique os logs kubelet e o journalctl para localizar a causa raiz do problema. Para obter mais informações, consulte Log do Kubelet .

Depois de identificar o erro, conclua as etapas a seguir:

1. Faça as correções necessárias no arquivo de configuração. Para obter detalhes de configuração, consulte Configurando um vSphere Cloud Provider após a instalação do IBM Cloud Private.

2. Recarregue o arquivo de unidade systemd do kubelet.

   ```
   systemctl daemon-reload
   ```

3. Reinicie o serviço kubelet para trazer o nó de volta no estado ativo.

   ```
   Systemctl restart kubelet.service
   ```

   Nota: ao reiniciar o serviço kubelet, é possível que algum rótulo ou contaminação que você incluiu manualmente no nó seja perdido. Se sim, é possível incluí-lo novamente agora.

Falha no fornecimento de volume persistente

Falha no fornecimento de volume persistente (PV) com um erro "No VM found".

Causas

Esse problema ocorre quando o identificador do modo principal não é configurado ou é configurado incorretamente.

Talvez você veja as seguintes mensagens de log no arquivo de log do gerenciador do controlador:

```
[nodemanager.go:419] Error "No VM found" node info for node "master-node-01" not found
[vsphere_util.go:134] Error while obtaining Kubernetes node nodeVmDetail details. error : No VM
found
[vsphere.go:1160] Failed to get shared datastore: No VM found
```

Resolvendo o problema

1. Verifique o ID do provedor e o número de série do produto dos nós principais.
   a. Obtenha o número de série do produto do nó principal.

   ```
   sudo cat /sys/class/dmi/id/product_serial | sed -e 's/^VMware-//' -e 's/-/ /' | awk '{ print
   toupper($1$2$3$4 "-" $5$6 "-" $7$8 "-" $9$10 "-" $11$12$13$14$15$16) }'
   ```

   O seguinte é uma saída de amostra:

   ```
   4228B6D2-87BA-4094-2578-129A9085585C
   ```

   b. Obtenha o ID do provedor das informações do nó do Kubernetes.

   ```
   kubectl get node <master-node-01> -o json | jq '[.metadata.name, .spec.providerID]'
   ```

   O seguinte é uma saída de amostra:

   ```
   [
     "master-node-01",
     "vsphere://4228154b-efa3-51e0-80e2-53000dcdf383"
   ]
   ```

2. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

3. Configure o ID do provedor com o valor correto do número de série do produto.

   ```
   kubectl patch node <master-node-01> -p '{"spec":{"providerID":"vsphere://4228B6D2-87BA-4094-
   2578-129A9085585C"}}'
   ```

   Nota: se não for possível modificar o objeto do nó, deve-se excluir as informações do nó do inventário do Kubernetes e reiniciar o kubelet para que o nó seja registrado novamente.

hostPath

Configure hostPath para armazenamento de dados.

Um volume hostPath monta um arquivo ou diretório a partir do sistema de arquivos do nó do host em seu pod.

Deve-se criar o diretório no nó do host antes que um pod seja planejado no nó. Quando o pod é replanejado em um nó diferente, ele perde todos os dados persistentes. Portanto, a opção hostPath não é ideal para cargas de trabalho de produção, a menos que você tenha algum outro mecanismo em vigor para replicar dados de armazenamento.

Para obter mais informações sobre o hostPath, consulte hostPath.

Cenários de Implementação

Um volume hostPath é útil quando você deseja acessar os diretórios do sistema no nó do host. A seguir estão alguns cenários em que o volume hostPath pode ser usado:

- Um contêiner precisa acessar o Docker interno. Exemplo: `/var/lib/docker`.
- Você deseja executar o cAdvisor em um contêiner.

Se for necessário usar o hostPath e manter os dados persistentes, assegure-se de que seu Pod esteja consistentemente planejado no mesmo nó.

Para obter informações sobre a criação de um volume hostPath no IBM® Cloud Private, consulte Criando um PersistentVolume hostPath.

Network File System

O IBM® Cloud Private suporta o uso do Network File System (NFS) para armazenamento persistente.

O servidor NFS está localizado fora de seu cluster do IBM Cloud Private e deve ser instalado e configurado independentemente. O servidor NFS deve estar acessível a partir de seu nó principal do IBM Cloud Private e o sistema de arquivos NFS deve ter acesso de leitura e gravação.

Para obter mais informações sobre como configurar um servidor NFS, consulte Configurando um servidor NFS.

Para obter mais informações sobre os pré-requisitos, a preparação do nó e a criação de um volume persistente no IBM Cloud Private, consulte Criando um PersistentVolume do NFS.

IBM Spectrum Scale

Aprenda como usar o IBM Spectrum Scale™ para armazenamento em seu cluster do IBM® Cloud Private.

- Usando o IBM Spectrum Scale para armazenamento em seu cluster do IBM Cloud Private

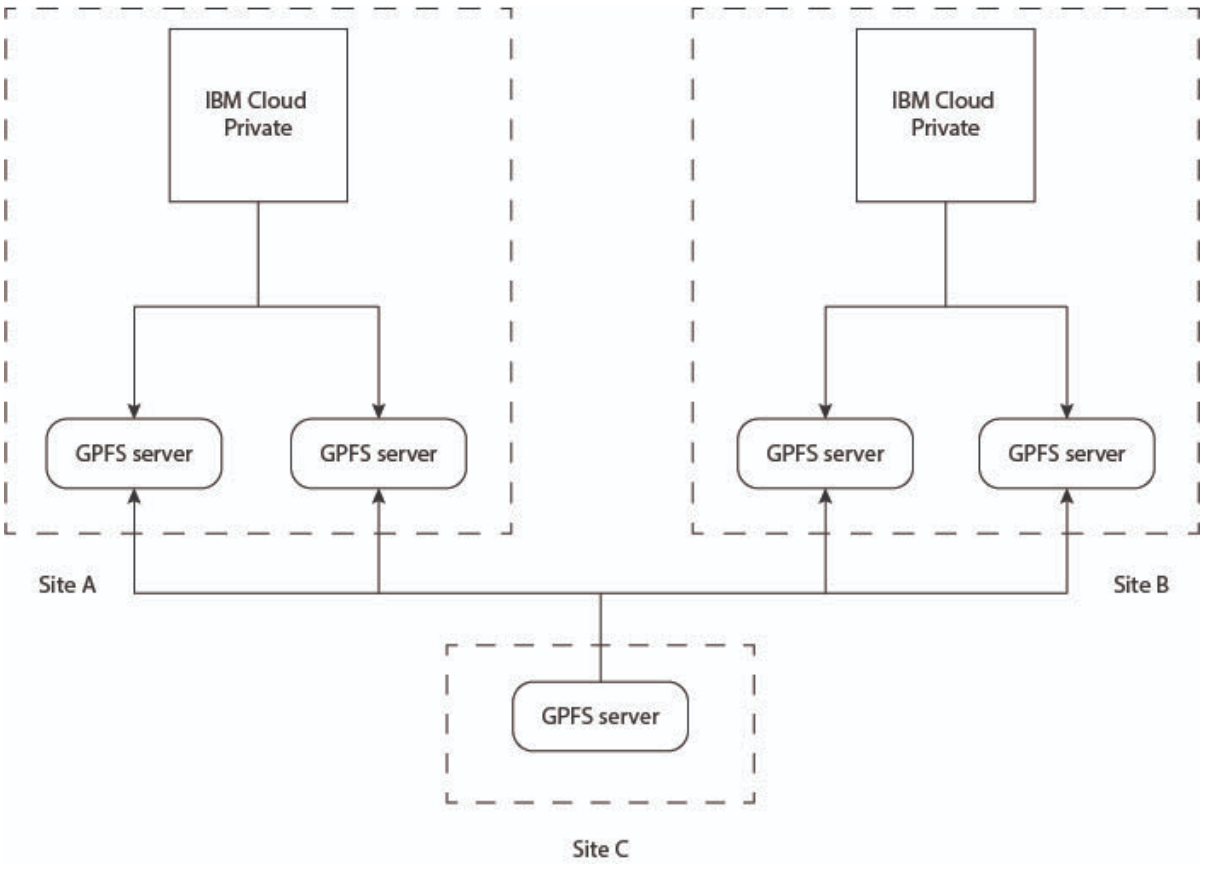
Usando o IBM Spectrum Scale para armazenamento em seu cluster do

IBM Cloud Private

Crie volumes e use-os para armazenamento persistente.

O IBM Spectrum Scale é um sistema de arquivo em cluster que fornece acesso simultâneo a um único sistema de arquivos ou conjunto de sistemas de arquivos por meio de vários nós. Para obter mais informações sobre o IBM Spectrum Scale, consulte Visão geral do IBM Spectrum Scale.

É possível usar um volume do IBM Spectrum Scale para um pod em seu cluster do IBM Cloud Private. Consulte o seguinte cenário de implementação de exemplo do IBM Cloud Private com o IBM Spectrum Scale.



É possível usar o IBM Spectrum Scale para fornecer um sistema de arquivos de alta disponibilidade usando o hostPath ou o Network File System (NFS). As seções a seguir fornecem informações sobre como usar o hostPath e o NFS para criar e solicitar um volume.

Configurando o IBM Spectrum Scale para IBM Cloud Private

Para integrar o IBM Cloud Private e o IBM Spectrum Scale, deve-se concluir as tarefas a seguir:

1. Configurar o cluster do IBM Spectrum Scale
2. Criar um sistema de arquivos
3. Montar um sistema de arquivos
4. Criar um conjunto de arquivos
5. Montar o conjunto de arquivos

Configurar o cluster do IBM Spectrum Scale

Configure seu cluster do IBM Spectrum Scale. Para obter mais informações, consulte Configurações de cluster do IBM Spectrum Scale e Etapas para estabelecer e iniciar seu cluster do IBM Spectrum Scale.

Deve-se incluir todos os nós do trabalhador do IBM Cloud Private como nós do cliente do IBM Spectrum Scale. Para obter mais informações, consulte Criando um cluster do IBM Spectrum Scale.

Criar um sistema de arquivos

Deve-se criar um sistema de arquivos no cluster do IBM Spectrum Scale. Para obter mais informações sobre como criar um sistema de arquivos, consulte Considerações sobre a criação do sistema de arquivos. Para obter mais informações sobre o comando para criar um sistema de arquivos, consulte Comando mmcrfs.

Em seguida, monte o sistema de arquivos.

Montar um sistema de arquivos

Monte o sistema de arquivos em todos os nós no cluster do IBM Spectrum Scale. Para obter mais informações, consulte Montando um sistema de arquivos.

Criar um conjunto de arquivos

Crie um conjunto de arquivos no dispositivo do sistema de arquivos. Os conjuntos de arquivos fornecem um meio de particionar o sistema de arquivos para permitir operações administrativas em uma granularidade mais fina do que o sistema de arquivos inteiro.

Para obter mais informações sobre conjuntos de arquivos, consulte Conjuntos de arquivos.

Para obter mais informações sobre o comando para criar um conjunto de arquivos, consulte Comando mmcrfileset.

Montar o conjunto de arquivos

Monte o sistema de arquivos para vincular o conjunto de arquivos. Para obter mais informações, consulte Vinculando um conjunto de arquivos.

Usando o hostPath para criar um volume persistente

Com o hostPath, o nó cliente do IBM Spectrum Scale é configurado como um nó do trabalhador para o IBM Cloud Private. O nó do trabalhador é usado para executar aplicativos para o IBM Cloud Private e também é usado para armazenamento.

Exemplo de como usar o hostPath para criar um volume persistente

A seguir está uma configuração de exemplo de como usar o hostPath para criar um volume persistente.

Criar um volume persistente

Crie um volume persistente de hostPath no cluster do IBM Cloud Private. Para obter mais informações, consulte Criando um hostPath PersistentVolume.

Considere este arquivo YAML de amostra:

```
---
apiVersion: v1
kind: PersistentVolume
```

```
metadata:
  name: gpfsstorage
  namespace: default
  labels:
    storage: gpfs
spec:
  capacity:
    storage: 1Gi
  hostPath:
    path: "/ibm/fs1/filesetpath"            <======= Path where you mounted the filseset
  accessModes:
  - ReadWriteMany
```

Revise o status do volume persistente. Antes de poder usar o volume persistente, o status do volume persistente deve estar `Disponível`.

Criar uma solicitação de volume persistente

Crie uma solicitação de volume persistente para torná-lo disponível para ser consumido por um aplicativo ou pod. Para obter informações adicionais sobre como criar uma solicitação de volume persistente, consulte Criando um PersistentVolumeClaim.

Considere este arquivo YAML de amostra:

```
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gpfsclaim
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  selector:
    matchLabels:
      storage: gpfs            <======== Label of the persistent volume
  resources:
    requests:
      storage: 512Mi
  volumeName: gpfsstorage
```

Revise o status da solicitação de volume persistente. Antes de poder usar a solicitação de volume persistente, o status da solicitação de volume persistente deve ser `Ligado`.

A solicitação de volume persistente agora está disponível para ser usada por um pod ou um aplicativo.

Ao criar um aplicativo ou pod, especifique o nome da solicitação de volume persistente na seção `Volumes:` para torná-lo disponível para o pod ou aplicativo.

Usando o NFS para criar um volume persistente

Com o NFS, é possível usar o nó cliente do IBM Spectrum Scale para armazenamento dedicado. É possível exportar o NFS para uso com o IBM Cloud Private. O NFS pode ser configurado para alta disponibilidade com o IBM Spectrum Scale com o modo de acesso `ReadWriteMany`.

Usando o IBM Storage Enabler for Containers

O IBM Storage Enabler for Containers é um plug-in que permite que sistemas de armazenamento de bloco IBM e o IBM Spectrum Scale sejam usados como dispositivos de armazenamento para clusters de contêineres do Kubernetes. Para obter informações adicionais sobre o IBM Storage Enabler for Containers, consulte a documentação do IBM Storage Enabler for Containers versão 2.0.0. Para obter informações adicionais sobre o instalador do plug-in, consulte Instalador para IBM Storage Enabler for Containers.

RBD externo do Ceph

Saiba como usar o cluster RBD do Ceph Externo para armazenamento em seu cluster do IBM® Cloud Private.

- Pré-requisitos

Pré-requisitos

Pré-requisitos para integrar um cluster do Ceph externo com seu cluster do IBM® Cloud Private.

- Um cluster do Ceph que esteja ativo e em execução.
- Um conjunto de armazenamentos do Ceph pré-criado e ativado com um aplicativo RADOS Block Device (RBD).
- Uma chave do usuário para o ID do cliente Ceph que pode criar imagens no conjunto.
- Uma chave do usuário para o ID do cliente Ceph que é usada para mapear a imagem do RBD e ter o recurso 'allow rwx' para o conjunto de armazenamentos do Ceph que você pretende usar.
- Nós do cluster do Ceph com acesso aos nós do cluster do IBM Cloud Private.
- Um cluster do IBM Cloud Private que está ativo e em execução.

Nota: para obter os IDs do cliente Ceph e criar um conjunto Ceph, você deve ter acesso ao cluster do Ceph como usuário administrador. Ou, o administrador de cluster do Ceph deve tornar esses requisitos disponíveis para você.

Integrando o cluster do Ceph externo com seu cluster do IBM Cloud Private

Integre um cluster do Ceph que está configurado fora do ambiente do IBM® Cloud Private.

O IBM Cloud Private usa o armazenamento do Ceph usando o fornecedor Kubernetes `kubernetes.io/rbd` integrado. As cargas de trabalho do aplicativo usam o armazenamento de bloco do Ceph usando o fornecimento de volume dinâmico do Kubernetes com base em uma classe de armazenamento configurada.

Nota: as instruções nas seções a seguir são baseadas no Ceph v12.2.10 Luminous e IBM Cloud Private Versão 3.1.2.

Prepare seu cluster do Ceph externo

Conclua estas etapas para preparar seu cluster do Ceph para integração com seu cluster do IBM Cloud Private. Você deve ser um administrador para executar esses comandos.

Crie um conjunto do Ceph e um ID do usuário para o conjunto

Conclua as etapas a seguir para criar um conjunto do Ceph e um ID do usuário que possa ser usado em sua classe de armazenamento do IBM Cloud Private:

1. Crie um conjunto Ceph.

```
ceph osd pool create demo 8 8
```

O seguinte é uma saída de amostra:

```
conjunto 'demo' criado
```

1. Designe um aplicativo RBD ao conjunto para que ele possa ser usado como um dispositivo de bloco.

```
ceph osd pool application enable demo rbd
```

O seguinte é uma saída de amostra:

```
enabled application 'rbd' no conjunto 'demo'
```

1. Crie um usuário `auth` para o conjunto para montar o volume do RBD em seus nós do cluster do IBM Cloud Private.

```
ceph auth add client.demo mon 'allow r' osd 'allow rwx pool=demo'
```

O seguinte é uma saída de amostra:

```
chave incluída para client.demo
```

1. Verifique se o usuário foi criado.

```
ceph auth ls
```

O seguinte é uma saída de amostra:

```
entradas de auth instaladas:

osd.0
    key: AQB5hEVclZvxFRAAnIhvzBMHgaN+cqpEXQStmQ==
    caps: [mgr] allow profile osd
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.1
    key: AQCXhEVc40LyNhAABYGlOVafoVXgVQgCttdvIw==
    caps: [mgr] allow profile osd
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.2
    key: AQDjhEVcaKoIFhAAwiXG6puVjWsrVmzgVv4Q/g==
    caps: [mgr] allow profile osd
    caps: [mon] allow profile osd
    caps: [osd] allow *
client.admin
    key: AQDOe0VcCle6ERAA6L82BeosLNJ7FJwqq5W1+A==
    caps: [mds] allow *
    caps: [mgr] allow *
    caps: [mon] allow *
    caps: [osd] allow *
client.bootstrap-mds
    key: AQDPe0VcGoqTDhAAS/0mJFrdrL+EYkJWJC7BsQ==
    caps: [mon] allow profile bootstrap-mds
client.bootstrap-mgr
    key: AQDQe0VcCKKmBBAA/wcLWNISSOlD2Ju56Pp71w==
    caps: [mon] allow profile bootstrap-mgr
client.bootstrap-osd
    key: AQDRe0VcwZcFBhAAXlAQw/wa1qZhWylrJeMr9g==
    caps: [mon] allow profile bootstrap-osd
client.bootstrap-rgw
    key: AQDRe0VcK6NuOxAAQO8SUEMEQRLud/Wls8BBvA==
    caps: [mon] allow profile bootstrap-rgw
client.demo
    key: AQBLflJcPW5UFxAAYIqBmmT3sRdADV7GbArZPQ==
    caps: [mon] allow r
    caps: [osd] allow rwx pool=demo
mgr.sanverm22-master.fyre.ibm.com
    key: AQApfEVcE9RPExAA3RKGvibVhJzOJOH3OYVVRQ==
    caps: [mds] allow *
    caps: [mon] allow profile mgr
    caps: [osd] allow *
```

Obtenha as chaves de ID do usuário e ID do administrador do Ceph

Obtenha as chaves para o usuário `admin` e o usuário `demo`. As chaves são necessárias para criar segredos e uma classe de armazenamento em seu cluster do IBM Cloud Private.

1. Obtenha a chave para o `adminId`.

   ```
   ceph auth get-key client.admin | base64
   ```

   O seguinte é uma saída de amostra:

   ```
   QVFET2UwVmNDbGU2RVJBQTZMODJCZW9zTE5KN0ZKd3FxNVcxK0E9PQ ==
   ```

2. Obtenha a chave para o `userID`.

   ```
   ceph auth get-key client.demo | base64
   ```

   O seguinte é uma saída de amostra:

   ```
   QVFCTGZsSmNQVzVVRnhBQVlJcUJtbVQzc1JkQURWN0diQXJaUFE9PQ==
   ```

Prepare seu cluster do IBM Cloud Private para integração com o cluster do Ceph externo

Prepare seus nós do cluster do IBM Cloud Private

Instale o software cliente Ceph em cada nó do cluster IBM Cloud Private em que os pods do aplicativo que usam armazenamento Ceph podem ser planejados.

- No Ubuntu, execute o comando a seguir:

  ```
  sudo apt-get install ceph-common
  ```

- No Red Hat Enterprise Linux (RHEL), execute o seguinte comando:

  ```
  sudo yum install ceph-common
  ```

Crie segredos em seu cluster do IBM Cloud Private usando IDs do cliente Ceph

Antes de iniciar, instale o kubectl em seu cluster do IBM Cloud Private. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl). Conclua essas tarefas em seu nó principal do IBM Cloud Private.

Crie um segredo para o ID do cliente Ceph `adminID`

Nota: para a chave do administrador, consulte Obter as chaves do ID do usuário e ID do administrador do Ceph.

1. Use o arquivo YAML a seguir para criar segredo:

   Nota: o segredo deve ser do tipo `kubernetes.io/rbd`.

```
apiVersion: v1
kind: Secret
metadata:
  name: ceph-admin-secret
  namespace: kube-system
type: "kubernetes.io/rbd"
data:
  # ceph auth get-key client.admin | base64
  key: QVFET2UwVmNDbGU2RVJBQTZMODJCZW9zTE5KN0ZKd3FxNVcxK0E9PQ==
```

1. Execute o comando a seguir para criar o segredo:

   ```
   kubectl create -f ceph-admin-secret.yaml
   ```

2. Verifique se o segredo foi criado.

   ```
   kubectl get secrets -n kube-system | grep ceph
   ```

   O seguinte é uma saída de amostra:

   ```
   ceph-admin-secret                                       kubernetes.io/rbd
   1      53m
   ```

Crie um segredo para o ID do cliente Ceph `userID`

Nota: para a chave do usuário, consulte Obter as chaves do ID do usuário e ID do administrador do Ceph.

1. Use o arquivo YAML a seguir para criar segredo:

   Nota: o segredo deve ser do tipo `kubernetes.io/rbd`.

```
apiVersion: v1
kind: Secret
```

```
metadata:
  name: ceph-secret
  namespace: kube-system
type: "kubernetes.io/rbd"
data:
  # ceph auth add client.demo mon 'allow r' osd 'allow rwx pool=kube'
  # ceph auth get-key client.demo | base64
  key: QVFCTGZsSmNQVzVVRnhBQVlJcUJtbVQzc1JkQURWN0diQXJaUFE9PQ==
```

1. Execute o comando a seguir para criar o segredo:

   ```
   kubectl create -f ceph-secret.yaml
   ```

1. Verifique se o segredo foi criado.

   ```
   kubectl get secrets -n kube-system | grep ceph
   ```

   O seguinte é uma saída de amostra:

   ```
   ceph-admin-secret                                          kubernetes.io/rbd
   1      53m
   ceph-secret                                                kubernetes.io/rbd
   1      53m
   ```

Crie uma classe de armazenamento em seu cluster do IBM Cloud Private

Crie uma classe de armazenamento para o fornecedor kubernetes.io/rbd. Para obter mais informações, consulte Ceph RBD

1. Use o arquivo YAML a seguir para criar uma classe de armazenamento. Você deve revisar cada parâmetro e fornecer valores para seu cluster do Ceph.

   ```
   kind: StorageClass
   apiVersion: storage.k8s.io/v1
   metadata:
   name: rbd
   provisioner: kubernetes.io/rbd
   parameters:
   monitors: 1.1.1.1:6789,1.1.1.2:6789,1.1.1.3:6789
   pool: demo
   adminId: admin
   adminSecretNamespace: kube-system
   adminSecretName: ceph-admin-secret
   userId: demo
   userSecretNamespace: kube-system
   userSecretName: ceph-secret
   imageFormat: "2"
   imageFeatures: "layering"
   ```

2. Crie a classe de armazenamento.

   ```
   kubectl create -f rbd-storage-class.yaml storageclasslass.storage.k8s.io/rbd criado
   ```

3. Verifique se a classe de armazenamento foi criada com êxito.

   ```
   Kubectl get sc
   ```

O seguinte é uma saída de amostra:

```
NAME                        PROVISIONER                    AGE
rbd                         kubernetes.io/rbd              5s
```

Forneça um volume persistente no cluster IBM Cloud Private

Use o arquivo YAML de amostra a seguir para criar uma solicitação de volume persistente (PVC) usando a classe de armazenamento criada anteriormente:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: demo
spec:
  accessModes:
    - ReadWriteOnce
```

```
storageClassName: rbd
resources:
  requests:
    storage: 1Gi
```

Use essa PVC em um pod do aplicativo e verifique se o volume está montado no pod.

Resolução de Problemas do Ceph RBD Externo

Revise problemas do Ceph RBD encontrados frequentemente.

- Erro Warning FailedMount

Erro de Warning FailedMount

Você vê um `erro Warning FailedMount` para um pod de aplicativo.

Sintomas

- Você vê o erro `Warning  FailedMount  5s  kubelet, 10.41.14.185 MountVolume.WaitForAttach failed` para um pod do aplicativo que usa a classe de armazenamento Ceph RBD.
- Você vê o erro a seguir no nó em que o pod do aplicativo está planejado:

```
[670104.004657] libceph: mon0 10.41.11.177:6789 feature set mismatch, my 106b84a842a42 <
server's 40106b84a842a42, missing 400000000000000
[670104.010679] libceph: mon0 10.41.11.177:6789 missing required protocol features
```

Causas

O erro pode ser devido ao problema de compatibilidade do cliente (clientes antigos do kernel CephFS ou RBD ou clientes do pre-bobtail librados) nos nós clientes do Ceph. Para obter mais informações, consulte Ajustáveis do Ceph CRUSH.

Resolvendo o problema

Ajuste o Ajuste `CRUSH` de Ceph.

Execute o comando a seguir em seu cluster do Ceph para ajustar o `CRUSH` ajustável para o cliente Ceph mais antigo.

```
Conceh osd crush tunables legacy
```

O seguinte é uma saída de amostra:

```
perfil ajustado ajustáveis ao legado
```

Sistema de arquivos Ceph externo

Integre um cluster de armazenamento CephFS externo com o cluster do IBM® Cloud Private.

- Pré-requisitos
- Integrando o CephFS externo com o cluster do IBM Cloud Private

Pré-requisitos

Pré-requisitos para integrar um cluster de armazenamento CephFS externo com o cluster do IBM® Cloud Private.

Você deve ter um servidor CephFS externo que esteja funcionando. O compartilhamento deve ser exportado antes de poder usar o servidor.

- Um cluster CephFS deve estar funcionando. **Nota:** Estas instruções são validadas no Luminous v12.2.10.
- Todos os nós do IBM Cloud Private devem ser acessíveis pelos nós do cluster CephFS.

- O cluster do IBM Cloud Private deve estar funcionando. **Nota:** essas instruções são validadas no IBM Cloud Private Versão 3.2.0.

Para usar um volume CephFS existente em seu pod, você precisa das seguintes informações do cluster CephFS:

- Lista de monitores Ceph.
- Caminho como a raiz montada, e não a árvore Ceph completa. Se você não fornecer o caminho, o caminho `/` padrão será usado.
- O nome de usuário do CephFS. Se você não fornecer o nome do usuário, o nome do usuário `admin` padrão será usado.
- Um caminho para o arquivo `keyring`. Se você não fornecer o caminho, o caminho padrão `/etc/ceph/user.secret` será usado.
- Referência aos segredos de autenticação do Ceph. Se você fornecer a referência, o segredo substituirá o arquivo `keyring`.
- Você deve especificar se o sistema de arquivos é usado como `readOnly` ou não.

Nota: para obter o CephFS e criar uma exportação, você deve ter acesso a um cluster Ceph como um usuário administrador. Ou você deve obter as informações de seu administrador de cluster Ceph.

Integrando o CephFS externo com o cluster do IBM Cloud Private

Integre um cluster do CephFS que está localizado fora do ambiente IBM Cloud Private com o cluster do IBM Cloud Private.

O volume do CephFS é usado no IBM Cloud Private montando-o em pods do aplicativo. Ao contrário de `emptyDir`, que é apagado quando um pod é removido, o conteúdo de um volume do CephFS é preservado e o volume é simplesmente desmontado. Os dados podem ser carregados em um volume do CephFS e esses dados podem ser distribuídos entre os pods.

NOTA: O CephFS pode ser montado simultaneamente por vários gravadores.

Preparação do nó do IBM Cloud Private

Todos os nós do cluster do IBM Cloud Private devem estar acessíveis aos nós de monitoramento do CephFS.

Preparar o cluster do IBM Cloud Private para integração do CephFS

Antes de configurar o IBM Cloud Private para usar um armazenamento externo do CephFS, você deve obter as informações de chaves de exportação compartilhada e do ID do usuário do Ceph necessárias.

- Se você tiver acesso de administrador ao cluster do Ceph, execute o seguinte comando para obter a chave para o usuário administrador:

  ```
  ceph auth get-key client.admin | base64
  ```

  O seguinte é uma saída de amostra:

  ```
  QVFET2UwVmNDbGU2RVJBQTZMODJCZW9zTE5KN0ZKd3FxNVcxK0E9PQ ==
  ```

  Nota: É possível usar qualquer usuário do CephFS. Se o usuário do CepfFS não for o administrador, entre em contato com o administrador de cluster do CephFS para obter os detalhes do usuário e da chave.

- Crie segredos no cluster do IBM Cloud Private usando a chave do usuário cliente do Ceph

  Use o seguinte arquivo YAML para criar um segredo:

  ```
  kind: Secret
  metadata:
    name: ceph-admin-secret
    namespace: default
  data:
    # ceph auth get-key client.admin | base64
    key: QVFET2UwVmNDbGU2RVJBQTZMODJCZW9zTE5KN0ZKd3FxNVcxK0E9PQ==
  ```

Nota: Especifique o `namespace` que você deseja usar para o volume do CephFS no segredo.

```
kubectl create -f secret.yaml
```

O seguinte é uma saída de amostra:

```
secret/ceph-admin-secret created
```

Crie um PersistentVolume no cluster do IBM Cloud Private

Depois de criar o segredo para a chave do usuário do CephFS, é possível criar um PersistentVolume (PV).

Use o seguinte YAML para criar um PV. Você deve revisar cada parâmetro e fornecer o valor baseado no cluster do Ceph.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cephfs
  labels:
    pv: cephfs
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  cephfs:
    monitors:
    - 10.41.11.177:6789
    user: admin
    path: /
    secretRef:
      name: ceph-admin-secret
    readOnly: false
```

Os parâmetros a seguir são especificados na seção de especificação do CephFS do PV:

- `monitors`: Matriz de monitores do Ceph. Consulte o seguinte exemplo:

  ```
  monitors:
    - 10.16.154.78:6789
    - 10.16.154.82:6789
    - 10.16.154.83:6789
  ```

- `path`: Caminho como a raiz montada e não a árvore completa do Ceph. Se você não fornecer o caminho, o caminho padrão `/` será usado. É possível substituir e montar um caminho específico do sistema de arquivos usando o atributo do caminho. Consulte o seguinte exemplo:

  ```
  path: /some/path/in/side/cephfs
  ```

- `user`: O nome do usuário do RAS. Se você não fornecer o nome do usuário, o nome do usuário padrão `admin` será usado.
- `secretRef`: Referência aos segredos de autenticação do Ceph. Esta referência é o nome do segredo criado na seção **Configurar o cluster do IBM Cloud Private para integração do CephFS**.
- `readOnly`: Especifique se o sistema de arquivos é usado como `readOnly`.

Nota: A mesma seção `cephfs` também pode ser especificada na especificação do pod. Se você fizer isso, não será necessário criar `PersistentVolume` e `PersistentVolumeClaim`. O pod monta o CephFS diretamente da especificação que você forneceu. Para obter informações adicionais, consulte o documento cephfs-with-secret.yaml.

Crie o PV executando o seguinte comando:

```
kubectl create -f cephfs-pv.yaml
```

O seguinte é uma saída de amostra:

```
persistentvolume/cephfs created
```

Crie um PersistentVolumeClaim no cluster do IBM Cloud Private

Use o seguinte YAML para criar um PersistentVolumeClaim.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: claim1
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 2Gi
  selector:
    matchLabels:
      pv: cephfs
```

Nota: Para corresponder o PV criado nas etapas anteriores, use os rótulos de correspondência do seletor `pv: cephfs` na definição de PersistentVolumeClaim (PVC).

Crie o PVC executando o seguinte comando:

```
kubectl create -f cephfs-pvc.yaml
```

O seguinte é uma saída de amostra:

```
persistentvolumeclaim/claim1 created
```

Verifique se o PVC está ligado ao PV.

```
kubectl get persistentvolumeclaim/claim1
```

O seguinte é uma saída de amostra:

```
NAME     STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
claim1   Bound    cephfs   2Gi        RWX                           2m31s
```

Use o PVC em um pod

A parte de código a seguir é um arquivo YAML de pod de amostra no qual o PVC é usado. O arquivo YAML faz a criação de um pod e monta o volume.

```
kind: Pod
apiVersion: v1
metadata:
  name: cephfs
spec:
  containers:
  - name: cephfs-rw
    image: gcr.io/google_containers/busybox:1.24
    command:
      - "/bin/sh"
    args:
      - "-c"
      - "touch /mnt/cephfs/SUCCESS && exit 0 || exit 1"
    volumeMounts:
    - mountPath: "/mnt/cephfs"
      name: cephfs
  restartPolicy: "Never"
  volumes:
  - name: cephfs
    persistentVolumeClaim:
      claimName: claim1
```

Crie o pod e verifique se o pod foi criado com sucesso.

Crie o pod:

```
kubectl create -f test-pod-pvc.yaml
```

O seguinte é uma saída de amostra:

```
pod/cephfs created
```

Verifique se o pod foi criado com sucesso.

```
kubectl get pod/cephfs
```

O seguinte é uma saída de amostra:

```
NAME     READY   STATUS      RESTARTS   AGE
cephfs   0/1     Completed   0          6s
```

Apêndice

A seção a seguir contém instruções para criar um compartilhamento de exportação do Ceph FS em seu cluster do CephFS. Você deve ter privilégio de administrador no cluster do CephFS para executar esses comandos.

Criar um compartilhamento de exportação do Ceph FS

1. Crie um conjunto de daemon de armazenamento de objetos (OSD) para exportação.

   ```
   ceph osd pool create cephfs_data 8
   ```

   O seguinte é uma saída de amostra:

   ```
   pool 'cephfs_data' created
   ```

2. Crie metadados.

   ```
   ceph osd pool create cephfs_metadata 8
   ```

   O seguinte é uma saída de amostra:

   ```
   pool 'cephfs_metadata' created
   ```

3. Crie um CephFS usando os conjuntos do OSD criados nas etapas anteriores.

   ```
   ceph fs new cephfs cephfs_metadata cephfs_data
   ```

   O seguinte é uma saída de amostra:

   ```
   new fs with metadata pool 7 and data pool 6
   ```

4. Visualize as informações do CephFS.

   ```
   ceph fs ls
   ```

   O seguinte é uma saída de amostra:

   ```
   name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
   ```

5. Crie servidores de Metadados (MDS) usando qualquer nó de monitoramento.

   ```
   ceph mds stat
   ```

   O seguinte é uma saída de amostra:

   ```
   cephfs-0/0/1 up
   ceph-deploy mds create tony-worker-1.abc.com
   ceph mds stat
   ```

   O seguinte é uma saída de amostra:

   ```
   cephfs-1/1/1 up  {0=tony-worker-1.abc.com=up:active}
   ```

Obtenha a chave client.admin a partir do servidor CephFS

A chave client.admin que você obtém do servidor CephFS é usada para criar um segredo em seu cluster do IBM Cloud Private.

Obtenha a chave client.admin do servidor CephFS

```
ceph auth get-key client.admin | base64
```

O seguinte é uma saída de amostra:

```
QVFET2UwVmNDbGU2RVJBQTZMODJCZW9zTE5KN0ZKd3FxNVcxK0E9PQ ==
```

Para obter informações adicionais sobre a configuração do CephFS, consulte os seguintes artigos:

- Exemplos de CephFS do Kubernetes
- Criar um Sistema de Arquivos Ceph

Opções de armazenamento disponíveis como gráficos do Helm da comunidade

Os gráficos do Helm da comunidade podem ser instalados no cluster do IBM® Cloud Private.

Para obter as opções disponíveis, consulte Gráficos da comunidade.

Medição, monitoramento e criação de log

Revise os recursos, as métricas e os logs do sistema.

- Serviço de medição do IBM® Cloud Private
- Monitoramento do IBM Cloud Private
- Monitoramento do sistema e de recursos
- Visualizando informações do pod
- Criação de log do IBM Cloud Private
- Planejamento da capacidade de criação de log e métricas doIBM Cloud Private
- Customizando nós Filebeat do IBM Cloud Private para o serviço de criação de log

Serviço de medição do IBM Cloud Private

É possível usar o serviço de medição para visualizar e fazer download de métricas de uso detalhadas para seus aplicativos e cluster. As medidas precisas são visíveis por meio da IU de medição e os dados são mantidos por até três meses. Os relatórios de resumo mensais também estão disponíveis para download e são mantidos por até 24 meses.

O serviço de medição é instalado automaticamente como parte do ambiente do IBM Cloud Private.

Nota: o serviço de medição não mede cargas de trabalho ou armazenamento no namespace `kube-system`, já que esses itens são considerados serviços do sistema.

- Visualizando relatórios de medição
- Usando o serviço de medição para gerenciar cobranças retroativas
- Rastreando o uso de produtos IBM que estão em execução fora de seu cluster do IBM Cloud Private

Visualizando relatórios de medição

1. Na console de gerenciamento do IBM Cloud Private, abra o menu de navegação e clique em **Plataforma > Medição**.
2. Visualize e faça download de dados de medição.
    - Para visualizar os dados de medição de aplicativos, clique em **Cargas de trabalho** e navegue para o aplicativo desejado.
        - É possível visualizar métricas para Processadores Disponíveis e Processadores Limitados.
        - É possível atualizar o intervalo de tempo exibido para mostrar um período de interesse.
        - É possível fazer download dos dados exibidos para um aplicativo específico como um arquivo `.csv`.
    - Para visualizar dados de medição para armazenamento, clique em **Armazenamento** e navegue para a solicitação de volume persistente (PVC) desejada.
        - É possível visualizar a métrica de armazenamento para ver a quantia de armazenamento alocada para essa PVC.
        - É possível atualizar o intervalo de tempo exibido para mostrar um período de interesse.
        - É possível fazer download dos dados de armazenamento exibidos como um arquivo `.csv`.
        - Os PVCs são agrupados por namespace. O grupo `Armazenamento` indica a quantia de armazenamento que é usada pelo cluster em todos os namespaces.

- Para fazer download de um arquivo `.csv` que contém um relatório mensal de dados de métricas do aplicativo, clique em **Fazer download do relatório** e, em seguida, selecione para fazer download do **Relatório de carga de trabalho** ou do **Relatório integral**.
  - O **Relatório de carga de trabalho** contém um resumo simplificado de dados de uso nos níveis do produto e do namespace e é mais fácil de usar para reembolsos.
  - O **Relatório integral** contém um resumo detalhado de dados de uso para o produto e para cada instância de contêiner e pode ser usado para auditoria e análise detalhadas.
  - Seu uso de Núcleo de Processamento Virtual (VPC) é o número de *Núcleos* que está listado no relatório.
- Para visualizar dados de medição para nós em seu cluster, clique em **Plataforma**.
- Se o Multicloud Manager (MCM) estiver instalado, o serviço de medição no cluster de hub do MCM incluirá mais métricas do MCM.
  - Para visualizar dados do MCM, selecione **Clusters gerenciados**.
  - Para fazer download de um arquivo `.csv` que contém um resumo mensal de dados do MCM, clique em **Fazer download do relatório** e selecione a opção **Relatório de múltiplos clusters**.
  - O **Relatório de múltiplos clusters** mostra o número máximo de nós que são gerenciados pelo MCM em qualquer ponto em cada mês e o máximo de nós do trabalhador para cada cluster gerenciado. O máximo para MCM pode ou não corresponder à soma dos nós em cada cluster. Ele depende de quando, durante o mês, cada cluster foi incluído ou removido do controle do MCM. Os níveis máximos de cluster são incluídos apenas para permitir o reembolso.
  - Para visualizar dados do MCM na IU de medição ou para poder fazer download do Relatório de Múltiplos Clusters, deve-se ter a função `Administrador de cluster`.

3. Leia o arquivo `.csv` do relatório.
   - O relatório está no formato CSV, que pode ser carregado por qualquer software de planilha moderno. No entanto, como os arquivos CSV não incluem detalhes de formatação, pode ser necessário expandir colunas de interesse para ver todos os dados.
   - O relatório é organizado em seções por mês. O mês rastreado mais antigo aparece no início da seção e os meses mais recentes aparecem no término da seção. A coluna `Period` exibe o ano e o mês da seção em cada linha.
   - A coluna `Status` exibe se a seção é `FINAL` ou `PENDING`. Os dois meses mais recentes são marcados como `PENDING`. Mais dados de medição podem ainda ser processados e essas seções de relatório são atualizadas. Quando uma seção é marcada como `FINAL`, ela não está mais sendo atualizada.
   - Dentro de cada seção mensal, os dados são mais separados nas categorias a seguir:
     - Resumo geral
     - Resumo por grupo (por exemplo, namespace) e produto
     - Resumo para cargas de trabalho individuais.

Algumas cargas de trabalho, especialmente as cargas de trabalho externas ao IBM Cloud Private, podem incluir métricas adicionais além dos Processadores Disponíveis (Acores) e Processadores Limitados (Ccores) padrão coletados automaticamente para contêineres pelo serviço de medição. No download do **Relatório integral**, essas métricas extras são representadas por duas colunas, a métrica em si e uma coluna extra, intitulada `breakdown`. A coluna `breakdown` fornece informações técnicas detalhadas que podem ajudar a equipe de suporte ou auditores, mas que normalmente não precisam ser inspecionadas.

Para fazer download dos relatórios de medição a partir da linha de comandos em vez de a partir da IU, use o comando `metering` que faz parte da CLI do IBM Cloud Private. Para obter mais informações, consulte Comandos de medição da CLI do IBM (metering).

Usando o serviço de medição para gerenciar cobranças retroativas

É possível rastrear o uso do IBM Cloud Private por um subconjunto dos usuários do cluster, como equipes, departamentos, indivíduos ou clientes específicos. Use designações estratégicas de namespace para rastrear os dados. É possível usar os dados que o serviço de medição reúne sobre o uso do núcleo para gerenciar o faturamento para os usuários do cluster. Essa prática é comumente conhecida como estorno.

Para usar namespaces para gerenciar cobranças retroativas, determine o subconjunto de usuários que você deseja rastrear. Em seguida, você cria um namespace para cada grupo de usuários e designa cada acesso de usuário apenas ao namespace para o grupo de usuários. Quando for necessário gerenciar os reembolsos para os grupos, revise o uso para cada namespace. É possível usar as informações de uso para cada namespace para designar o faturamento para o grupo.

Rastreando o uso de produtos IBM que estão em execução fora de seu cluster do IBM Cloud Private

Se você deseja enviar dados de produtos que são executados fora da plataforma IBM para o serviço de medição, os parâmetros a seguir deverão ser configurados:

- Um terminal `https` para o qual os dados são enviados.

- A chave API para autenticação.

Para criar a chave API, conclua as etapas a seguir:

1. Crie um `serviceid` que se vuncule ao namespace ao que o usuário do IBM Cloud Private tem acesso. Por exemplo:

   ```
   cloudctl login -a https://<icp-cluster-ip>:8443 -n default --skip-ssl-validation
   cloudctl iam service-id-create my-serviceid-for-metering -d "Metering access serviceid in the
   default namespace"
   ```

2. Crie uma `política` de serviço, que conceda ao `serviceid` o acesso da função de operador ou de uma função mais alta ao `metering-service`. Por exemplo:

   ```
   cloudctl iam service-policy-create my-serviceid-for-metering -r Operator --service-name
   metering-service
   ```

3. Crie uma chave API que se conecte ao `serviceid`. Por exemplo:

   ```
   cloudctl iam service-api-key-create my-apikey my-serviceid-for-metering
   ```

Para obter mais informações sobre como criar `serviceids`, políticas de serviço e chaves API a partir da CLI, consulte Criando um ID de serviço usando a CLI do IBM Cloud Private.

Os usuários com funções de Administrador de Cluster ou de Administrador podem visualizar qualquer chave API que se liga ao namespace para o qual essas funções têm acesso.

1. Na console de gerenciamento do IBM Cloud Private, abra o menu de navegação e clique em **Plataforma > Medição**.
2. No painel de medição, selecione **Gerenciar chaves API**. Nesse formulário, é possível recuperar as chaves API para sua plataforma. Também é possível recuperar o terminal da API de medição para produtos que estão em execução fora da plataforma.

Se um produto IBM estiver ativado para medição, será possível configurar a comunicação entre o produto e o IBM Cloud Private. Revise a documentação do produto externo para obter informações sobre como configurar a comunicação usando chaves API.

Os produtos externos que enviam dados para o serviço de medição do IBM Cloud Private podem fornecer métricas específicas do produto. Essas métricas estão disponíveis por meio do painel de medição e no relatório transferível por download. Em alguns casos, o valor no relatório pode ser limitado. Nesses casos, mais detalhes são fornecidos no relatório que explica como o valor é determinado. Revise a documentação do produto externo para obter detalhes sobre a abordagem de medição para cada métrica.

Monitoramento de cluster do IBM Cloud Private

É possível usar o painel de monitoramento de cluster do IBM® Cloud Private para monitorar o status de seu cluster e aplicativos.

O painel de monitoramento usa Grafana e Prometheus para apresentar dados detalhados sobre seus nós e contêineres do cluster. Para obter mais informações sobre Grafana, consulte a documentação do Grafana. Para obter mais informações sobre Prometheus, consulte a documentação do Prometheus.

- Acessando o painel de monitoramento
- Métricas coletadas fora da caixa
- Acesso Baseado em Função
- Instalando o serviço de monitoramento no IBM Cloud Private
- Configurando o servidor Prometheus
- Alertas
- Gerenciando Painéis Grafana
- Configurando aplicativos para usar o serviço de monitoramento
- Gerenciamento de logs e métricas para Prometheus
- Acessando as APIs do serviço de monitoramento

Acessando o painel de monitoramento

1. Efetue login no console de gerenciamento IBM Cloud Private.

   Nota: quando você efetua login no console de gerenciamento, você tem acesso administrativo ao Grafana. Não crie mais usuários no painel do Grafana nem modifique os usuários ou a organização existente.

2. Para acessar o painel do Grafana, clique em **Menu** > **Plataforma** > **Monitoramento**. Como alternativa, é possível abrir `https://<IP_address>:<port>/grafana`, em que `<IP_address>` é o DNS ou endereço IP que é usado para acessar o console do IBM Cloud Private. `<port>` é a porta que é usada para acessar o console do IBM Cloud Private.
3. Para acessar o painel do Alertmanager, clique em **Menu** > **Plataforma** > **Alerta**. Como alternativa, é possível abrir `https://<IP_address>:<port>/alertmanager`.
4. Para acessar o painel do Prometheus, abra `https://<IP_address>:<port>/prometheus`.
5. No painel do Grafana, abra um dos painéis a seguir:
    - **ElasticSearch**
      Fornece informações sobre as estatísticas do cluster do ElasticSearch, shard e outras informações do sistema.
    - **Etcd por Prometheus**
      Etcd Dashboard para o raspador de métricas do Prometheus.
    - **Métricas de Liberação do Helm**
      Fornece informações sobre métricas do sistema, como `CPU` e `Memory`, de cada liberação do Helm que é filtrada por pods.
    - **ICP Namespaces Performance IBM Provided 2.5**
      Fornece informações sobre o desempenho do namespace e métricas de status.
    - **Cluster Network Health (Calico)**
      O Calico hospeda informações de desempenho de carga de trabalho e de métrica do sistema.
    - **Desempenho do ICP IBM Fornecido 2.5**
      Fornece informações de desempenho do sistema TCP sobre `Nodes`, `Memory` e `Containers`.
    - **Monitoramento de Cluster do Kubernetes**
      Monitora clusters Kubernetes que usam o Prometheus. Fornece informações sobre o uso de `CPU`, `Memory` e `Filesystem` do cluster. O painel também fornece estatísticas para pods, contêineres e serviços do sistema individuais.
    - **Visão geral do Kubernetes POD**
      Monitora métricas de pod, como `CPU`, `Memory`, status do pod `Network` e reinicializações.
    - **Controlador de ingresso NGINX**
      Fornece informações sobre as métricas do controlador de Ingresso NGINX que podem ser classificadas por namespace, classe de controlador, controlador e ingresso.
    - **Resumo de Desempenho do Nó**
      Fornece informações sobre as métricas de desempenho do sistema, como `CPU`, `Memory`, `Disk` e `Network`, de todos os nós no cluster.
    - **Estatísticas do Prometheus**
      Painel para monitorar o Prometheus v2.x.x.
    - **Armazenamento GlusterFS de Armazenamento**
      Fornece métricas do GlustersFS Health, como `Status`, `Storage` e `Node`.
    - **Rook-Ceph**
      Painel que fornece estatísticas sobre instâncias Ceph.
    - **Armazenamento Minio de Armazenamento**
      Fornece detalhes de armazenamento e de rede sobre instâncias do servidor Minio.

    Nota: se você configurar os pods para usar o recurso de nível de host como *rede de host*, os painéis exibirão as métricas do host, mas não o próprio pod.

Se você deseja visualizar outros dados, é possível criar novos painéis ou importar painéis por meio de arquivos de definição JSON para Grafana.

Métricas coletadas prontas para uso

O IBM Cloud Private fornece os exportadores a seguir para fornecer métricas. Os exportadores expõem os terminais de métricas como serviços do Kubernetes.

- **node-exporter** Fornece as métricas do nível do nó, incluindo métricas para CPU, memória, disco, rede e outros componentes.
- **kube-state-metrics** Fornece as métricas para objetos do Kubernetes, incluindo métricas para `pod`, `deployment`, `statefulset`, `daemonset`, `replicaset`, `configmap`, `service`, `job` e outros objetos.
- **elasticsearch-exporter** Fornece métricas para o serviço de criação de log do IBM Cloud Private Elasticsearch, incluindo o status para o cluster do Elasticsearch, shards e outros componentes.
- **collectd-exporter**
  Fornece métricas que são enviadas a partir do plug-in de rede coletado.

Alguns pods Kubernetes do IBM Cloud Private fornecem terminais de métricas para o Prometheus:

- **calico-node**
  Fornece métricas para os nós do Calico.
- **nginx-ingress-controller**
  Fornece métricas para o controlador de ingresso do Nginx.

Além disso, o Prometheus possui destinos de extração pré-configurados que se comunicam com vários destinos para extrair métricas:

- O **cAdvisor** fornece métricas do contêiner que incluem CPU, memória, rede e outros componentes.
- **Prometheus**
  Fornece métricas para o servidor Prometheus que incluem as métricas para manipulação de solicitações, avaliação de regra de alerta, status do TSDB e outros componentes.
- **kubernetes-apiservers**
  Forneça métricas para os servidores de API do Kubernetes.
- **etcd**
  Fornece métricas para o IBM Cloud Private `etcd`.

Controle de acesso baseado na função (RBAC)

RBAC para API de monitoramento

Um usuário com a função *ClusterAdministrator*, *Administrator* ou *Operator* pode acessar o serviço de monitoramento. Um usuário com a função *ClusterAdministrator* ou *Administrator* pode executar operações de gravação no serviço de monitoramento, incluindo a exclusão de dados de métricas do Prometheus e atualização das configurações do Grafana.

RBAC para dados de monitoramento

Iniciando com a versão 1.2.0, o gráfico Helm `ibm-icpmonitoring` apresenta um recurso importante. Ele oferece um novo módulo que fornece controles de acesso baseados em função (RBAC) para acesso aos dados de métricas do Prometheus.

O módulo RBAC é efetivamente um proxy que se situa na frente do pod de cliente do Prometheus. Ele examina as solicitações para cabeçalhos de autorização e, nesse ponto, força os controles baseados em função. Em geral, as regras relativas ao RBAC são as seguintes:

Um usuário com a função *ClusterAdministrator* pode acessar qualquer recurso. Um usuário com qualquer outra função pode acessar apenas dados nos namespaces para os quais esse usuário está autorizado.

RBAC para painéis de monitoramento

A partir da versão 1.5.0, o gráfico do Helm `ibm-icpmonitoring` oferece um novo módulo que fornece controles de acesso baseados em função (RBAC) para acesso aos painéis de monitoramento no Grafana.

No Grafana, os usuários podem pertencer a uma ou mais organizações. Cada organização contém suas próprias configurações para recursos, como origens de dados e painéis. Para o Grafana em execução no IBM Cloud Private, cada namespace no IBM Cloud Private tem uma organização correspondente com o mesmo nome. Por exemplo, se você criar um novo namespace denominado *test* no IBM Cloud Private, uma organização denominada *test* será gerada no Grafana. Se você excluir o namespace *test*, a organização *test* também será removida. A única exceção é o namespace `kube-system`. A organização correspondente para `kube-system` é o padrão Grafana de `Main Org`.

Cada organização Grafana inclui uma origem de dados padrão denominada `prometheus`, que aponta para o Prometheus no serviço de monitoramento. Cada organização também inclui os seguintes painéis:

- Visão geral do POD do Kubernetes
- Métricas da Liberação do Helm

Todos os painéis de monitoramento prontos para uso mencionados em Acessando o painel de monitoramento são importados na organização `Main Org`.

Ao efetuar login no IBM Cloud Private, você só pode acessar uma organização Grafana se estiver autorizado a acessar o namespace correspondente. Caso tenha acesso a mais de uma organização Grafana, use o console do Grafana para alternar para uma organização diferente. A mensagem `UNAUTHORIZED` aparece quando você não tem acesso a uma organização Grafana.

Diferentes usuários do IBM Cloud Private acessam organizações Grafana usando diferentes funções de organização. No namespace correspondente, se receber a designação da função de `ClusterAdministrator` ou `Administrator`, você terá acesso de `Administrador` à organização Grafana. Caso contrário, você terá acesso de `Visualizador` à organização Grafana.

Quando você acessa o Grafana como um usuário do IBM Cloud Private, um usuário com o mesmo nome é criado no Grafana. Se o usuário no IBM Cloud Private for excluído, o usuário correspondente não será excluído do Grafana. A conta do usuário se torna antiga. Execute o comando a seguir para solicitar a remoção de usuários antigos:

```
curl -k -s -X POST -H "Authorization:$ACCESS_TOKEN" https://<Cluster Master Host>:<Cluster Master API Port>/grafana/check_stale_users
```

Para obter informações sobre APIs do Grafana, consulte Acessando APIs do serviço de monitoramento.

Nota: o serviço de monitoramento não fornece suporte a RBAC para alertas Prometheus e Alertmanager.

Instalando o serviço de monitoramento no IBM Cloud Private

O serviço de monitoramento é instalado por padrão durante a instalação do IBM Cloud Private. Também é possível selecionar para instalar o serviço de monitoramento por meio do Catálogo ou da CLI.

Instalando o serviço de monitoramento a partir do Catálogo

É possível implementar mais pilhas de monitoramento com configurações customizadas a partir do Catalog no console de gerenciamento do IBM Cloud Private.

1. Na página `Catalog`, clique no gráfico Helm `ibm-icpmonitoring` para configurá-lo e instalá-lo.
2. Forneça os valores necessários para os parâmetros a seguir.

```
Helm release name: "monitoring"
Target namespace: "kube-system"
Mode of deployment: "Managed"
Cluster access address: Specify the Domain Name Service (DNS) or IP address that is used to access the IBM Cloud Private console.
Cluster access port: Specify the port that is used to access the IBM Cloud Private console. A porta padrão é 8443.
etcd address: Specify the Domain Name Service (DNS) or IP address for etcd node(s).
```

Instalando o serviço de monitoramento a partir da CLI

1. Instale a linha de comandos do Kubernetes (`kubectl`). Para obter informações sobre a CLI do `kubectl`, consulte Acessando o cluster a partir da CLI do Kubernetes (kubectl).
2. Instale a interface da linha de comandos (CLI) de Helm. Para obter informações sobre a CLI do Helm, consulte Instalando a CLI do Helm (helm).
3. Instale o gráfico Helm `ibm-icpmonitoring`. Execute o comando a seguir:

```
helm install -n monitoring --namespace kube-system --set mode=managed --set clusterAddress=<IP_address> --set clusterPort=<port> ibm-icpmonitoring-1.4.0.tgz
```

`<IP_address>` é o DNS ou o endereço IP que é usado para acessar o console IBM Cloud Private.

`<port>` é a porta que é usada para acessar o console do IBM Cloud Private.

Para obter mais informações sobre parâmetros que podem ser configurados durante a instalação, consulte Parâmetros.

Configuração de Persistência de Dados

Por padrão, os dados do usuário nos componentes de serviço de monitoramento, como o Prometheus, o Grafana ou o AlertManager, não são armazenados em volumes persistentes. Os dados do usuário serão perdidos se o componente de serviço de monitoramento travar. Para armazenar dados do usuário em volumes persistentes, é necessário configurar parâmetros relacionados ao instalar o serviço de monitoramento. Use uma das opções a seguir para ativar volumes persistentes:

- Use volumes que são dinamicamente provisionados. Deve-se usar um provedor de armazenamento que suporte fornecimento dinâmico. Por exemplo, é possível configurar o GlusterFS para criar dinamicamente volumes persistentes. Durante a configuração, marque a caixa de seleção para `Persistent volume` e forneça valores para os parâmetros a seguir:

  ```
  Size for the persistent volume
  Name of the storageClass for the persistentVolume
  ```

- Use o `PersistentVolume` existente. Deve-se criar manualmente os volumes persistentes antes de instalar o serviço de monitoramento. Ao definir os volumes persistentes, deve-se criar `rótulos` que sejam usados como um mecanismo de identificação para monitoramento de componentes de serviço.

Durante a configuração, marque a caixa de seleção para `Persistent volume` e forneça valores para os parâmetros a seguir:

```
Size for the persistent volume
Name of the storageClass for the persistentVolume
Field to select the volume
Value of the field to select the volume
```

No exemplo a seguir, o valor de `Campo para selecionar o volume` é *component*. O valor de `Valor do campo para selecionar o volume` é *prometheus*.

```
apiVersion: v1
kind: PersistentVolume
metadata:
    name: monitoring-prometheus-pv
    labels:
        component: prometheus
.......
```

- Use o `PersistentVolumeClaims` existente. Deve-se criar manualmente volumes persistentes e solicitações de volume persistente. Durante a configuração, marque a caixa de seleção para `Volume persistente` e forneça um valor para o parâmetro `Name of existing persistentVolumeClaim`.

Para obter informações sobre como criar classes de armazenamento, `PersistentVolume` e `PersistentVolumeClaim`, consulte Armazenamento.

Configurando o servidor Prometheus

É possível configurar os seguintes parâmetros do servidor Prometheus durante a pré-instalação ou pós-instalação:

- **scrape_Interval**

  O parâmetro para a frequência para os destinos de extração. O valor padrão é 1 minuto (`1m`).

- **evaluation_Interval**

  O parâmetro para a frequência de avaliação de regras. O valor padrão é 1 minuto (`1m`).

- **retenção**

  O parâmetro para a frequência de remoção de dados antigos. O valor padrão é 24 horas (`24h`).

- **resources.limits.memory**

  O parâmetro para a limitação de memória para o contêiner do Prometheus. O valor padrão é `4096Mi`. O contêiner do Prometheus trava quando a limitação de memória não é cumprida. Deve-se aumentar o valor desse parâmetro para assegurar-se de que o contêiner do Prometheus possa funcionar corretamente.

Configuração de Pré-instalação

Para a instalação do serviço de monitoramento e o IBM Cloud Private, é possível configurar os parâmetros no `config.yaml` antes da instalação. Por exemplo, seu arquivo `config.yaml` pode ser semelhante ao conteúdo a seguir:

```
monitoring:
  prometheus:
    scrape_Interval: 1m
    evaluation_Interval: 1m
    retention: 24h
    resources:
      limits:
        memory: 4096Mi
```

Se você optar por instalar o serviço de monitoramento a partir do Catalog, será possível configurar os parâmetros nos campos da console relacionados.

Configuração de pós-instalação

Também é possível atualizar os parâmetros depois de instalar o serviço de monitoramento.

- Atualize os parâmetros para `scrape_Interval` e `evaluation_Interval` no ConfigMap `monitoring-prometheus`:
  1. Abra o ConfigMap `monitoring-prometheus` a partir da console do IBM Cloud Private ou use a CLI `kubectl`.
  2. Atualize os valores para `scrape_Interval: 1m` ou `evaluation_Interval: 1m` e envie suas mudanças. As atualizações de parâmetro são aplicadas ao servidor Prometheus ativo. Não é necessário reiniciar o servidor.

- Para atualizar os parâmetros `retention`:

  1. Abra a implementação `monitoring-prometheus` a partir da CLI da console do IBM Cloud Private ou use a CLI `kubectl`.
  2. Atualize o valor para `---storage.tsdb.retention=24h` e envie suas mudanças.

- Para atualizar os parâmetros `resources.limits.memory`:

  1. Abra a implementação `monitoring-prometheus` a partir da CLI da console do IBM Cloud Private ou use a CLI `kubectl`.

  2. Atualize o valor para `resources.limits.memory` e envie suas mudanças.

     Notas:

     - Ao atualizar o valor `retention` ou `resources.limits.memory`, o pod do Prometheus ativo é excluído e um novo pod do Prometheus é iniciado.
     - Modificações para ConfigMap ou implementações serão perdidas se você reimplementar o gráfico de monitoramento. Por exemplo, se você fizer upgrade para uma nova versão.

Alertas

Alertas padrão

A capacidade de instalar alertas padrão está disponível na versão 1.3.0 do gráfico `ibm-icpmonitoring`. Alguns alertas fornecem parâmetros customizáveis para controlar a frequência de alerta. É possível configurar os alertas a seguir durante a instalação.

- **Uso de memória do nó**
  Alerta padrão a ser acionado quando o limite de memória do nó excede 85%. O limite é configurável e é instalado por padrão. Se você usar a CLI, os valores a seguir controlarão esse alerta:

Campo	Valor Padrão
prometheus.alerts.nodeMemoryUsage.nodeMemoryUsage.enabled	true
prometheus.alerts.nodeMemoryUsage.nodeMemoryUsageThreshold	85

- **Alto Uso da CPU**
  Alerta padrão a ser acionado quando o limite da CPU excede 85%. O limite é configurável e é instalado por padrão. Se você usar a CLI, os valores a seguir controlarão esse alerta:

Campo	Valor Padrão
prometheus.alerts.highCPUUsage.enabled	true

Campo	Valor Padrão
prometheus.alerts.highCPUUsage.highCPUUsageThreshold	85

- **Tarefas com Falha**
  Alerta padrão se uma tarefa não competia com sucesso. É instalado por padrão. Se você usar a CLI, os valores a seguir controlarão esse alerta:

Campo	Valor Padrão
prometheus.alerts.failedJobs	true

- **Funcionamento do cluster do Elasticsearch**
  Alerta padrão acionado se o cluster do Elasticsearch do sistema não estiver verde. Este alerta não é instalado por padrão. Se você usar a CLI, os valores a seguir controlarão esse alerta:

Campo	Valor Padrão
prometheus.alerts.elasticsearchClusterHealth	false

- **Pods finalizados**
  Alerta padrão se um pod foi finalizado e não foi concluído com sucesso. Este alerta é instalado por padrão. Se você usar a CLI, os valores a seguir controlarão esse alerta:

Campo	Valor Padrão
prometheus.alerts.podsTerminated	true

- **Pods Reiniciando**
  O alerta padrão será acionado se um pod estiver reiniciando mais de 5 vezes em 10 minutos. Este alerta é instalado por padrão. Se você usar a CLI, os valores a seguir controlarão esse alerta:

Campo	Valor Padrão
prometheus.alerts.podsRestarting	true

Gerenciando regras de alerta

É possível usar o recurso customizado do Kubernetes, AlertRule, para gerenciar regras de alerta no IBM Cloud Private.

O arquivo `sample-rule.yaml` a seguir é um exemplo de uma definição de recurso do AlertRule.

```
apiVersion: monitoringcontroller.cloud.ibm.com/v1
kind: AlertRule
metadata:
  name: sample-rule
spec:
  enabled: true
  data: |-
    groups:
      - name: a.rules
        rules:
          - alert: NodeMemoryUsage
            expr: ((node_memory_MemTotal_bytes - (node_memory_MemFree_bytes +
node_memory_Buffers_bytes + node_memory_Cached_bytes))/ node_memory_MemTotal_bytes) * 100 > 5
            annotations:
              DESCRIPTION: '{{ $labels.instance }}: Memory usage is above the 15% threshold.  O
valor atual é: { }. '
              SUMMARY: '{{ $labels.instance }}: High memory usage detected'
```

Deve-se fornecer os valores de parâmetro a seguir:

- **apiVersion**
  `monitoringcontroller.cloud.ibm.com/v1`

- **amável**
  `AlertRule`

- **spec.data**
  Possui o conteúdo da regra de alerta. Para obter informações detalhadas sobre os arquivos de regras de alerta, consulte Registrando regras.

- **spec.enabled**
  Configure a sinalização para especificar se a regra de alerta está ativada ou não.

Use `kubectl` para gerenciar regras de alerta.

- Criar nova regra de alerta.

  ```
  kubectl aplicar -f sample-rule.yaml -n kube-sistema
  ```

- Edite as regras de alerta existentes.

  ```
  kubectl edit alertrules / sample-rule -n kube-system
  ```

- Exclua as regras de alerta existentes.

  ```
  kubectl delete alertrules / sample-rule -n kube-system
  ```

Configurando o AlertManager

É possível configurar o AlertManager do Prometheus para integrar os receptores de serviço de alerta externos, como o Slack ou o PagerDuty, ao IBM Cloud Private.

Importante: As mudanças no ConfigMap são perdidas quando você faz upgrade, retrocede ou atualiza a liberação de monitoramento. Além disso, o formato ConfigMap pode mudar entre as liberações.

1. Edite o mapa de configuração `monitoring-prometheus-alertmanager` para atualizar as configurações do AlertManager.

   ```
   kubectl edit configmap monitoring-prometheus-alertmanager -n kube-system
   ```

   Para obter mais informações sobre como configurar o AlertManager, consulte exemplos de Configuração do e Modelo de notificação

2. Permita vários minutos para que as atualizações entrem em vigor. Abra o painel do AlertManager em `https://<Cluster Master Host>:<Cluster Master API Port>/alertmanager`. Em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal.

   - Se você configurou alertas e eles estiverem acionados, será possível ver os alertas no painel do AlertManager.
   - Se você configurou um receptor de alerta externo como Slack ou PagerDuty, será possível visualizar os alertas no painel para esse serviço específico.
   - É possível retornar para os painéis para visualizar alertas a qualquer momento.

Gerenciando Painéis Grafana

É possível gerenciar painéis do Grafana operando em um MonitoringDashboard de recurso customizado do Kubernetes no IBM Cloud Private. O arquivo `sample-dashboard.yaml` a seguir é um exemplo de uma definição do recurso MonitoringDashboard.

```
apiVersion: monitoringcontroller.cloud.ibm.com/v1
kind: MonitoringDashboard
metadata:
  name: sample-dashboard
spec:
  enabled: true
  data: |-
    {
        "id": null,
        "uid": null,
        "title": "Marco Test Dashboard",
        "tags": [ "test" ],
        "timezone": "browser",
        "schemaVersion": 16,
        "version": 1
     }
```

Deve-se fornecer os valores de parâmetro a seguir:

- **apiVersion**
  `monitoringcontroller.cloud.ibm.com/v1`

- **amável**
  `MonitoringDashboard`

- **spec.data**
  Possui o conteúdo do arquivo de definição do painel Grafana. Para obter mais informações sobre arquivos de painel, consulte Painel JSON.

- **spec-enabled**
  Configure a sinalização para especificar se o painel está ativado ou não.

É possível usar o `kubectl` para gerenciar o painel. Use a opção `-n` para especificar o namespace no qual esse `MonitoringDashboard` deve ser criado. O painel será importado na organização correspondente no Grafana.

- Crie um novo recurso de painel no namespace `padrão` usando o arquivo `sample-dashboard.yaml`. O painel será importado na organização `padrão` no Grafana.

  ```
  kubectl apply -f sample-dashboard.yaml -n default
  ```

- Edite o painel de amostra.

  ```
  kubectl edit monitoringdashboards/sample-dashboard -n default
  ```

- Exclua o painel de amostra.

  ```
  kubectl edit monitoringdashboards/sample-dashboard -n default
  ```

Configure os aplicativos para usar o serviço de monitoramento

Modifique o aplicativo para expor as métricas.

- Para aplicativos que possuem um terminal de métricas, deve-se definir o terminal de métricas como um serviço Kubernetes usando a anotação `prometheus.io/scrape: 'true'`. A definição de serviço é semelhante ao código a seguir:

  ```
  apiVersion: v1
  kind: Service
  metadata:
    annotations:
      prometheus.io/scrape: 'true'
    labels:
      app: liberty
    name: liberty
  spec:
    ports:
    - name: metrics
      targetPort: 5556
      port: 5556
      protocol: TCP
    selector:
      app: liberty
    type: ClusterIP
  ```

  Nota: para obter mais informações sobre como configurar o terminal de métricas para o Prometheus, consulte BIBLIOTECAS DO CLIENTE na documentação do Prometheus.

- Os aplicativos podem ter mais de uma porta definida na definição de serviço. Não convém expor as métricas de monitoramento nas mesmas portas ou permitir que as portas sejam descobertas pelo Prometheus. É possível incluir a anotação `filter.by.port.name: 'true'` para que a porta cujo nome não comece com `metrics` seja ignorada pelo Prometheus. Na definição de serviço a seguir, o Prometheus coleta métricas da porta `metrics` e ignora as métricas da porta `collector`.

  ```
  apiVersion: v1 kind: Service metadata: annotations:
  prometheus.io/scrape: 'true' filter.by.port.name: 'true'
  labels: app: liberty name: liberty spec: ports:
    - name: metrics targetPort: 5556 port: 5556 protocol: TCP
    - name: collector targetPort: 8443 port: 8443 protocol: TCP selector: app: liberty
  type: ClusterIP
  ```

- Para aplicativos que têm um terminal de métrica com TLS ativado, é necessário usar IBM Cloud Private `cert-manager` para gerar um segredo e usá-lo para configurar o terminal de métrica.

  1. Use `cert-manager` para criar um recurso de certificado para uma carga de trabalho.

     ```
     apiVersion: certmanager.k8s.io/v1alpha1 kind:
     Certificate metadata: name:
     ```

```
{{.Release.Name }}-foo-certs namespace:
{{.Release.Namespace }} spec: secretName:
{{.Release.Name }}-foo-certs issuerRef: name:
icp-ca-issuer kind: ClusterIssuer commonName: "foo"
dnsNames:
            - "*. < MD:CONREF .Release.Namespace > .pod.cluster.local"
```

2. Monte o segredo em seu pod. É possível recuperar o certificado ou a chave do caminho montado. No caminho de montagem, há dois campos chamados `tls.crt` e `tls.key`. `tls.crt` inclui um arquivo de certificado de carga de trabalho e um arquivo de certificado que deve ser usado para configurar o terminal de métricas do aplicativo.

```
contêineres:
  - image: foo-image:latest name: foo volumeMounts:
      - mountPath: "/foo/certs" name: certs volumes:
  - name: certs secret: # secretName should be the same as the one defined in step 1.
     secretName: {{.Release.Name }} -foo-certs
```

3. Defina anotações sobre o serviço de carga de trabalho para permitir que o Prometheus use TLS para extrair métricas, `prometheus.io/scrape` e `prometheus.io/scheme`.

```
 apiVersion: v1 kind: Service metadata: annotations:
prometheus.io/scrape: 'true' prometheus.io/scheme: 'https'
```

- Para aplicativos que usam `collectd` e dependem de `collectd-exporter` para expor métricas, você atualiza o arquivo de configuração `collectd` dentro do contêiner de aplicativo. Nesse arquivo de configuração, deve-se incluir o plug-in de rede e apontar para o exportador `collectd`. Inclua o texto a seguir no arquivo de configuração:

```
LoadPlugin network
<Plugin network>
    Server "monitoring-prometheus-collectdexporter.kube-system" "25826"
</Plugin>
```

Gerenciamento de logs e métricas para Prometheus

É possível modificar o período de tempo para retenção da métrica atualizando o parâmetro `storage.tsdb.retention` no arquivo `config.yaml`. Por padrão, esse valor é configurado em `24h`, o que significa que as métricas são mantidas por 24 horas e depois limpas. Consulte Configurando o serviço de monitoramento.

No entanto, se for necessário remover manualmente esses dados do sistema, será possível usar a API de REST que é fornecida pelo componente Prometheus.

- Para excluir dados de métrica, consulte Excluir série.
- Para remover os dados excluídos do disco e limpar o espaço em disco, consulte Limpar tombstones.

A URL de destino deve ter o formato:

```
https:// < IP_address>: < Port> /prometheus
```

- `<IP_address>` é o endereço IP que é usado para acessar a console de gerenciamento.
- `<Port>` é a porta usada para acessar o console de gerenciamento.
    - O comando para excluir dados de métrica é semelhante ao seguinte código:

      ```
      https:// < IP_address>: < Port> /prometheus/api/v1/admin/tsdb/delete_series? *******
      ```

    - O comando para remover dados excluídos e limpar o disco é semelhante ao seguinte código:

      ```
      https:// < IP_address>: < Port> /prometheus/api/v1/admin/tsdb/clean_tombstones
      ```

Acessando APIs de serviço de monitoramento

É possível acessar APIs de serviço de monitoramento, como as APIs do Prometheus e do Grafana. Para poder acessar as APIs, deve-se obter tokens de autenticação para especificação em seus cabeçalhos de solicitação. Para obter informações sobre como obter tokens de autenticação, consulte Preparando-se para executar os comandos da API de componente ou de gerenciamento.

Depois de obter os tokens de autenticação, conclua as etapas a seguir para acessar as APIs do Prometheus e do Grafana.

1. Acesse a API do Prometheus na url `https://<Cluster Master Host>:<Cluster Master API Port>/prometheus/*` e obtenha tempos de inicialização para todos os nós.

- `$ACCESS_TOKEN` é a variável que armazena o token de autenticação para seu cluster.
- `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

```
curl -k -s -X GET -H "Authorization:Bearer $ACCESS_TOKEN" https://<Cluster Master Host>:
<Cluster Master API Port>/prometheus/api/v1/query?query=node_boot_time_seconds
```

Para obter informações detalhadas sobre as APIs do Prometheus, consulte API Prometheus HTTP.

2. Acesse a API do Grafana na url `https://<Cluster Master Host>:<Cluster Master API Port>/grafana/*` e obtenha o painel de `amostra`.

   - `$ACCESS_TOKEN` é a variável que armazena o token de autenticação para seu cluster.
   - `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

```
curl -k -s -X GET -H "Authorization: Bearer $ACCESS_TOKEN” "https://<Cluster Master Host>:
<Cluster Master API Port>/grafana/api/dashboards/db/sample"
```

Para obter informações detalhadas sobre as APIs do Grafana, consulte Referência de API HTTP do Grafana.

Monitoramento de sistema e recurso

É possível revisar as métricas atuais do sistema e de recursos no **Painel** do IBM® Cloud Private.

Visão geral do sistema

Visualize informações gerais sobre os nós, armazenamento e aplicativos em seu cluster.

Nós

Visualize o número de nós ativos e inativos no cluster.

Armazenamento compartilhado

Visualize o status dos volumes de armazenamento que são definidos no cluster.

- Disponível - A quantidade de armazenamento que não é usada por um volume
- Usado - A quantidade de armazenamento que é usada por volumes
- Liberado - A quantidade de armazenamento que não é recuperada de volumes excluídos
- Com falha - A quantia de armazenamento que falhou na recuperação automática

Aplicativos

Visualize o número de aplicativos com bom funcionamento e mau funcionamento no cluster. Aplicativos com bom funcionamento estão em execução e disponíveis, mas aplicativos com mau funcionamento são interrompidos.

Visão geral de recursos

Visualize informações sobre o uso de recursos. É possível visualizar as informações a seguir sobre cada recurso:

- A quantidade total do recurso que está disponível para todos os nós no cluster.
- A quantidade do recurso que está em uso.
- A quantidade do recurso que está alocada. O valor alocado é a soma do limite de uso máximo desse recurso para cada um dos aplicativos no cluster.

Informações sobre os recursos de cluster a seguir são mostradas:

CPU

Para obter o uso total de CPU em seu cluster, a console de gerenciamento do IBM Cloud Private primeiro chama a API do Kubernetes para obter o total de núcleos de CPU no cluster. A taxa de uso da CPU é, então, calculada usando a fórmula a seguir:

```
(CPU cores used/Total CPU cores) * 100
```

Memória

Para obter o uso total de memória em seu cluster, a console de gerenciamento do IBM Cloud Private primeiro chama a API do Kubernetes para obter a memória total disponível no cluster. Em seguida, a console de gerenciamento chama o servidor de métricas para obter a memória total que está sendo usada por contêineres. A taxa de uso da memória é, então, calculada usando a fórmula a seguir:

```
(Memory used/Total memory available) * 100
```

GPU

Para obter o GPU total disponível em seu cluster, o console do IBM Cloud Private chama a API do Kubernetes.

Nota: atualmente, as métricas não podem relatar o uso de GPU. Portanto, o uso total da GPU que é mostrado no painel é apenas um reflexo da GPU total em seu cluster.

Visualizando informações de pod

É possível visualizar informações sobre os pods em seu cluster.

Visualizando os detalhes de um pod

1. Efetue login no IBM® Cloud Private console de gerenciamento.
2. Clique em **Menu** > **Cargas de trabalho** > **Implementações**. É exibida uma lista de implementações em seu cluster.
3. Clique no nome da implementação. Os detalhes da implementação são exibidos. Os pods são listados na seção `Pods` na página.
4. Clique no nome do pod. A página de detalhes do pod exibe as informações do pod.

Para visualizar métricas de uso do pod, use o painel de monitoramento do cluster do IBM Cloud Private. Para obter mais informações, consulte Monitoramento do cluster do IBM Cloud Private.

Criação de log do IBM Cloud Private

Saiba como configurar e gerenciar a criação de log.

- Visão geral
- Segurança
- Configuração
- Requisitos e recomendações de hardware

Visão geral

O IBM Cloud Private implementa uma pilha ELK, referida como o serviço de criação de log de gerenciamento, para coletar e armazenar todos os logs capturados pelo Docker. Várias opções estão disponíveis para customizar a pilha antes de instalar o IBM Cloud Private, incluindo a criptografia TLS de ponta a ponta. É possível implementar e customizar mais pilhas do ELK a partir do catálogo, ou implementar outras soluções de terceiros, oferecendo flexibilidade máxima para gerenciar seus logs.

O serviço de criação de log de gerenciamento oferece uma ampla variedade de opções para configurar a pilha para adequar às suas necessidades:

- Alocação de memória por pod. Para obter informações adicionais, consulte Planejamento de capacidade de criação de log e de métricas do IBM® Cloud Private)
- Tamanho mínimo do disco
- A criptografia TLS
- Nó de Filebeat e Escopo de Namespace
- Políticas de retenção de dados
- Coleção de logs de
- Autenticação baseada em função
- Acesso ao Kibana através do ingresso

- ELK
- Integração do Docker
- Logs de processamento

- Notas de Pós-implementação
- Visualizando e consultando logs
- APIs do Elasticsearch

ELK

ELK é uma abreviação para três produtos, Elasticsearch, Logstash e Kibana, todos desenvolvidos pelo Elastic. Juntos, eles compõem uma pilha de ferramentas que transmitem, armazenam, pesquisam e monitoram dados, incluindo logs. Um quarto componente Elastic que é denominado Filebeat é implementado para transmitir os logs para o Elasticsearch.

Integração do Docker

Cada nó no cluster deve configurar o Docker para usar o driver de arquivo JSON. O Docker transmite os canais `stdout` e `stderr` de cada contêiner para um arquivo no host do Docker. Por exemplo: se um contêiner tiver o ID do Docker `abcd`, o local padrão para algumas plataformas para armazenar saída do contêiner será `/var/lib/docker/containers/abcd/abcd-json.log`. O gráfico de criação de log do IBM Cloud Private implementa um daemon do Filebeat configurado para cada nó para transmitir os arquivos de log JSON para a pilha ELK.

O Kubernetes inclui sua própria camada de abstração na parte superior de cada log de contêiner. Sob o caminho padrão, `/var/log/containers`, ele cria um symlink que aponta de volta para cada arquivo de log do Docker. O nome do arquivo symlink contém metadados extra do Kubernetes que podem ser analisados para extrair quatro campos:

```
                    |   1    |   2   |    3    |                                  4
|
/var/log/containers/pod-abcd_default_container-
5bc7148c976a27cd9ccf17693ca8bf760f7c454b863767a7e47589f7d546dc72.log
```

1. O nome do pod ao qual o contêiner pertence (armazenado como `kubernetes.pod`)
2. O namespace no qual o pod foi implementado (armazenado como `kubernetes.namespace`)
3. O nome do contêiner (armazenado como `kubernetes.container_name`)
4. O ID do Docker do contêiner (armazenado como `kubernetes.container_id`)

Logs de processamento

Logstash

O Logstash executa duas funções. Primeiro, ele armazena em buffers os dados entre o Filebeat e o Elasticsearch. Esse armazenamento em buffer protege contra a perda de dados e reduz o volume de tráfego para o Elasticsearch. Sua segunda função é analisar mais o registro de log para extrair metadados e tornar os dados no registro mais procuráveis. As etapas padrão a seguir são executadas pelo pod Logstash `ibm-icplogging`:

1. Analise o registro de data e hora do registro de log (armazenado pelo Docker no momento em que ele foi expresso pelo contêiner).
2. Extraia o nome do contêiner, o namespace, o ID do pod e o ID do contêiner em campos individuais.
3. Se o contêiner tiver gerado uma entrada de log formatada por JSON, analise-a e extraia os campos individuais para a raiz do registro de log.

O registro é então armazenado brevemente antes de o Logstash enviá-lo para o Elasticsearch.

Elasticsearch

Quando um registro de log é enviado para o Elasticsearch, ele se torna um *documento*. Cada documento é armazenado dentro de um grupo nomeado que é chamado de *índice*. Quando o Logstash envia um registro para o Elasticsearch, ele o designa a um índice com o padrão `logstash-<YYYY>-<MM>-<dd>`. Designar cada registro a um índice nomeado após o dia em que foi enviado torna mais fácil controlar as políticas de retenção de log.

O próprio Elasticsearch é executado independentemente em três tipos de pod diferentes. Muitas outras configurações são possíveis. Essa é a configuração que é escolhida no gráfico do Helm de `ibm-icplogging`.

- O pod *client* expõe os terminais da API de REST.
- O pod *master* controla o estado do cluster geral e também registra metadados nos quais os documentos são armazenados. Ele desempenha uma função para assegurar o armazenamento eficiente e a recuperação de dados.
- O pod *data* é responsável pelo armazenamento e a recuperação de documentos do Elasticsearch.

Kibana

O Kibana fornece uma consulta amigável ao navegador e uma interface de visualização para o Elasticsearch. Ele pode ser opcionalmente excluído da implementação, embora isso não seja recomendado, já que o Kibana é a ferramenta padrão por meio da qual os logs podem ser procurados.

Notas de Pós-implementação

- O Kibana requer vários minutos para otimizar seus plug-ins. Não é possível acessar o Kibana durante esse processo. Para obter mais informações, veja Atualizando e removendo plug-ins na documentação do Elastic.
- O Kibana pode requerer alguma configuração para índices depois de iniciá-lo. Para obter mais informações, veja Criando um padrão de índice para conectar ao Elasticsearch na documentação do Elastic.
    - A partir do IBM Cloud Private 3.1.2, um padrão de índice padrão é criado e configurado na pilha de criação de log no Kibana. O padrão de índice modelo fornece uma visualização inicial para os logs. Anteriormente, um usuário receberia uma mensagem de saudação e precisaria criar manualmente um padrão de índice para que uma visualização e uma procura úteis ficassem disponíveis. A criação do padrão de índice modelo pode levar vários minutos após a primeira inicialização. Como resultado, a IU do Kibana pode não estar disponível imediatamente após a instalação.

Visualizando e consultando logs

Kibana é a ferramenta primária para a interface com os logs. Ele oferece uma visualização Descoberta, por meio da qual é possível consultar os logs que atendem aos critérios específicos. É possível intercalar logs por meio dessa visualização usando um ou mais dos campos que são incluídos automaticamente pela pilha ELK `ibm-icplogging`.

- **kubernetes.container_id**: um identificador exclusivo que é gerado pelo Docker para cada contêiner.
- **kubernetes.container_name**: o nome legível para um contêiner designado pelo Kubernetes.
- **kubernetes.pod**: o nome do pod no qual um contêiner é implementado.
- **kubernetes.namespace**: o namespace no qual o pod do contêiner é implementado.

Pode ser necessário consultar os logs com base em outros critérios que não podem ser descobertos pela pilha ELK. Por exemplo, o produto de middleware, o nome do aplicativo ou o nível de log. Para obter a maior precisão dos logs do aplicativo, considere a saída formatada por JSON. O JSON declara os nomes dos valores no arquivo de log em vez de antecipar o Elasticsearch para analisá-lo com precisão. O conjunto de daemon Filebeat que é implementado pelo gráfico do Helm `ibm-icplogging` é pré-configurado para analisar entradas de log formatadas por JSON e configurar os valores para que eles sejam pesquisáveis como elementos de nível superior no Elasticsearch.

APIs do Elasticsearch

O Elasticsearch tem um alto grau de flexibilidade e uma API completamente documentada. A instalação segura da pilha ELK restringe o acesso da API a componentes internos que usam autenticação mútua sobre TLS, conforme descrito nas seções anteriores. Portanto, o acesso externo aos dados do Elasticsearch está disponível apenas para usuários que são autenticados por meio do Kibana. Também é possível usar o painel `dev tools` na interface com o usuário do Kibana para acessar a API do Elasticsearch. Se mais pilhas ELK forem implementadas no modo padrão, o acesso do Kibana não será protegido pelos controles de autenticação ou de autorização do IBM Cloud Private.

Nota: essas APIs funcionam apenas para consultar ou operar em dados que são rastreados atualmente no armazenamento de dados do Elasticsearch. Eles não afetam backups.

- Excluir por Consulta Excluir documentos inteiros (por exemplo, entradas de log) que correspondem a uma consulta específica.
- Atualizar por Consulta Semelhante à API `delete-by-query`, porém, é possível modificar o conteúdo dos campos ou remover completamente os campos específicos dos documentos. Para obter um exemplo de remoção de campos, consulte Atualizar API.
- Operações em massa. Conforme os dados do log se acumulam, determinadas operações podem levar mais tempo para serem concluídas. A API em massa é projetada para melhorar o desempenho, ativando várias operações dentro do contexto da mesma solicitação.

Segurança

- PKI no Elasticsearch

- Protegendo Dados em Trânsito
- Certificados
- Protegendo Dados-em-rest
- Acesso Baseado em Função
- Dados sensíveis

PKI no Elasticsearch

Iniciando com a versão 5.0, o plug-in de ativação do TLS antigo foi descontinuado e substituído por um novo plug-in chamado X-Pack. O X-Pack oferece inúmeros recursos extra que são comercializados para usuários corporativos, porém, é necessário ter uma licença. Os recursos são gratuitos para um período de uso limitado de 30 dias, após o qual todas as funções do X-Pack são desativadas.

O Search Guard é outro produto que oferece plug-ins relacionados à segurança para a pilha ELK. Em contraste com o X-Pack, alguns de seus recursos são oferecidos sob uma categoria **community edition** sem limitação em uso. Conforme indicado pelo arquivo leia-me: `O Search Guard oferece todos os recursos de segurança básica gratuitamente. O Community Edition do Search Guard pode ser usado para todos os projetos, incluindo projetos comerciais, sem nenhum custo.` A criptografia TLS com PKI é um desses recursos da edição de comunidade.

Por padrão, a pilha ELK do IBM Cloud Private usa o Search Guard para fornecer PKI. Se você já tiver uma licença para X-Pack, ou planejar comprar uma, será possível especificar os parâmetros a seguir durante a implementação para configurar a pilha ELK para usar a implementação de PKI do X-Pack. O cliente é responsável pela instalação da licença após a implementação.

```
logging:
  security:
    provider: xpack
```

Protegendo Dados em Trânsito

Cada implementação da pilha Elasticsearch é protegida por padrão com autenticação mútua sobre TLS. A pilha ELK gerenciada também é configurada para usar a autoridade de certificação do IBM Cloud Private para assinar os certificados usados pela pilha. Todas as outras pilhas ELK são padronizadas para criar sua própria autoridade de certificação na implementação. Para ativar ou desativar a segurança para mais pilhas ELK, desative a segurança na IU do catálogo ou os valores substituirão o arquivo para implementação do Helm.

Helm

O fragmento a seguir pode ser incluído em um arquivo de substituição de valores para implementação do Helm para ativar ou desativar a segurança.

```
security:
  enabled: true|false
```

Certificados

Todas as conexões com o Elasticsearch devem ser configuradas para trocar um certificado assinado adequadamente quando a segurança está ativada. A arquitetura de pilha ELK do IBM Cloud Private gera um número de certificados para aplicar às funções discretas. Todos são armazenados no mesmo segredo do Kubernetes e usam a seguinte convenção de nomenclatura: `<release_name>-ibm-icplogging-certs`.

Função ELK	Descrição	Nome da chave secreta	Armazenamento de Chaves	Formato de chave (key format)
Inicialização	Inicializa as configurações de Procura da Guarda	Sgadmin	JKS	PKCS12
Superusuário	Administrador do Elasticsearch	superuser	PEM	PKCS1
Filebeat	Cliente Logstash	Filebeat	PEM	PKCS1
Logstash	Server para Filebeat	Logstash	PEM	PKCS8
Logstash	Cliente para fluxo de logs do Elasticsearch	Monitoramento-logstash	JKS	PKCS12
Logstash	Client para monitoramento do Elasticsearch	Logstash-elasticsearch	JKS	PKCS12

Função ELK	Descrição	Nome da chave secreta	Armazenamento de Chaves	Formato de chave (key format)
Elasticsearch	Servidor de API REST	Elasticsearch	JKS	PKCS12
Elasticsearch	Intraperíodo nó de transporte	Elasticsearch-transporte	JKS	PKCS12
Curador	Cliente API REST Elasticsearch	Curador	PEM	PKCS1
Kibana	Cliente API REST Elasticsearch	Kibana	PEM	PKCS8
Kibana proxy	Server para conexões recebidas	Kibanarouter	PEM	PKCS1

Protegendo Dados-em-rest

A pilha Elasticsearch não oferece a criptografia de dados em repouso internamente. A empresa Elastic recomenda soluções de terceiros para atingir esse objetivo. O IBM Cloud Private tem instruções para métodos suportados de criptografia de dados no disco. Para obter mais informações, consulte Criptografando volumes que são usados pelo IBM Cloud Private.

Acesso Baseado em Função

A versão 2.0.0 do gráfico do Helm `ibm-icplogging` (incluída no IBM Cloud Private 3.1.0) introduziu um novo módulo, que fornece controles de acesso baseados em função (RBAC) para todas as chamadas da API de REST do Elasticsearch. O novo módulo está disponível apenas para as pilhas ELK gerenciadas.

O módulo RBAC é efetivamente um proxy que fica em frente de cada pod de cliente do Elasticsearch. Todas as conexões precisam ter certificados assinados pela CA do cluster do Elasticsearch. Por padrão, essa é a CA raiz do IBM Cloud Private. O módulo RBAC examina a solicitação para um cabeçalho de `autorização` e, nesse ponto, força os controles baseados em função. Em geral, as regras RBAC são as seguintes:

1. Um usuário com a função `ClusterAdministrator` pode acessar qualquer recurso, seja log de auditoria ou do aplicativo.
2. Um usuário com a função `Auditor` tem o acesso concedido somente aos logs de auditoria nos namespaces para os quais esse usuário está autorizado. Se os logs de auditoria forem roteados para ELK em vez de para a ferramenta SIEM corporativa existente sugerida, consulte Integração da criação de log de auditoria do IBM Cloud Private com ferramentas do SIEM corporativas.
3. Um usuário com qualquer outra função pode acessar os logs do aplicativo apenas nos namespaces para os quais esse usuário está autorizado.
4. Qualquer tentativa de um auditor de acessar logs do aplicativo, ou de um não auditor de acessar logs de auditoria, é rejeitada.

As regras RBAC fornecem controle de recuperação de dados básico para usuários que acessam o Kibana. As regras não lhe impedem de ver metadados, como nomes de campo de log ou painéis Kibana salvos.

Dados sensíveis

Pode ser necessário mascarar dados sensíveis antes de atingir o Elasticsearch. O Logstash implementa com um plug-in útil denominado Mutate, que oferece muitas funções para localizar dados que são considerados sensíveis. A inclusão dessas máscaras requer a customização da configuração de Logstash, que geralmente é localizada em um recurso `configmap` chamado `<release_name>-ibm-icplogging-logstash-config`. `release_name` refere-se ao nome da liberação fornecido a uma implementação do gráfico do Helm específico.

As modificações na configuração do Logstash serão propagadas automaticamente para os contêineres implementados após um breve atraso.

Modificações nos mapas de configuração serão perdidas se você reimplementar o gráfico de criação de log. Por exemplo, se você fizer upgrade para uma nova versão.

Configuration

- Instâncias de gráfico
- Instalando instâncias de criação de log adicionais
- Escalando serviços de criação de log após a instalação do IBM Cloud Private
- Autoridade de certificação customizada

- Retenção de dados
- Modificando a política de retenção de dados para serviços de criação de log
- Transmitindo logs da plataforma IBM Cloud Private externos
- Local de dados
- Ativando a segurança para serviços de criação de log
- Gerenciando a alocação de recurso para serviços de criação de log
- Atualizando filtros de coleção de dados de serviço de criação de log
- Ativando o monitoramento do Elastic
- Atualizando licenças do Elastic X-Pack
- Customizando nós Filebeat do IBM Cloud Private para o serviço de criação de log

Instâncias de gráfico

É possível implementar tantas instâncias independentes de criação de log quanto a capacidade de hardware permitir. O gráfico do Helm usado para implementar o serviço de criação de log está incluído no repositório `mgmt-charts`. Para obter mais informações sobre como configurar múltiplas instâncias de criação de log para segurança e ocupação variada, consulte Instalando instâncias de criação de log adicionais.

Autoridade de certificação customizada

A configuração padrão da pilha ELK gerenciada usa a autoridade de certificação (CA) do IBM Cloud Private. É possível localizar a CA no segredo `cluster-ca-cert` no namespace `kube-system`. O segredo tem dois campos (`tls.crt` e `tls.key`) que contêm o certificado real e sua chave privada. Todas as implementações posteriores do gráfico do Helm `ibm-icplogging` podem usar uma autoridade de certificação existente. Três requisitos devem ser atendidos:

1. A CA deve ser armazenada em um segredo do Kubernetes.
2. O segredo deve existir no namespace no qual a pilha ELK é implementada.
3. Os conteúdos do certificado e sua chave secreta devem ser armazenados em campos nomeados separadamente (ou chaves) dentro do segredo do Kubernetes.

Por exemplo, dado um segredo de amostra como o código a seguir:

```
apiVersion: v1
kind: Secret
metadata:
  name: my-ca-secret
type: Opaque
data:
  my_ca.crt: ...
  my_ca.key: ...
```

Deve-se, então, configurar o gráfico Helm com o subconjunto de valores a seguir:

```
security:
  ca:
    origin: external
    external:
      secretName: my-ca-secret
      certSecretKey: my_ca.crt
      keySecretKey: my_ca.key
```

Retenção de dados

Um contêiner é implementado como um `curador` dentro de cada pilha ELK. O `curador` remove os índices do Elasticsearch que são mais antigos que a idade máxima de índice configurada. Tenha cuidado ao armazenar logs por longos períodos de tempo. Cada dia adicional de logs retidos aumenta os recursos de memória e de armazenamento que o Elasticsearch requer.

Para modificar os valores padrão para o `curador` de pilha ELK gerenciada, inclua e customize as seguintes linhas em seu arquivo `config.yaml`.

```
logging:
  curator:
    name: log-curator
    image:
      repository: "ibmcom/indices-cleaner"
      tag: "2.0.0"
    # Runs at 23:30 UTC daily
```

```
schedule: "30 23 * * *"
# Application log retention
app:
  unit: days
  count: 1
# Elastcisearch cluster monitoring log retention
monitoring:
  unit: days
  count: 1
# X-Pack watcher plugin log retention
watcher:
  unit: days
  count: 1
```

Para mudar as configurações após a instalação, consulte Modificando a política de retenção de dados para serviços de criação de log.

Cronometr

O curador é configurado para ser executado no horário UTC. O uso de um único padrão de horário facilita a coordenação e a antecipação da curadoria nas regiões geográficas.

O tempo de ativação padrão é configurado para meia hora antes da meia-noite UTC. O propósito é evitar qualquer risco de atraso—talvez devido ao congestionamento ou carregamento do sistema—para iniciar o curador após o limite da meia-noite e armazenar mais logs do que o esperado.

Fluxo de IBM Cloud Private plataforma de fluxo externo

Os componentes da plataforma são implementados no namespace do sistema IBM Cloud Private `kube-system`, por padrão. Além disso, por padrão, apenas os componentes da plataforma serão implementados nos nós rotulados como `principal`, `gerenciamento` ou `proxy`.

Nesse cenário, é possível configurar a pilha ELK gerenciada no namespace do sistema IBM Cloud Private para transmitir logs da plataforma IBM Cloud Private para um serviço de coleta fora da plataforma.

Conclua as etapas a seguir para transmitir todos os logs de plataforma do IBM Cloud Private para um serviço externo.

1. Modifique a definição do Daemonset Filebeat para o namespace do sistema IBM Cloud Private para especificar a afinidade do nó apenas para os nós rotulados como `master`, `management` ou `proxy`.
2. Modifique a configuração do Logstash para a pilha implementada no namespace do sistema IBM Cloud Private para transmitir logs para um serviço de coleta fora da plataforma. Para obter mais informações, consulte a Documentação do Logstash.
3. Se não forem mais necessárias, exclua as implementações do Elasticsearch e Kibana e os StatefulSets definidos no namespace do sistema IBM Cloud Private.

Observações importantes sobre esta configuração:

1. A pilha ELK gerenciada não coletará mais nenhum log do aplicativo.
2. As mudanças na configuração não persistirão no upgrade ou retrocesso da liberação da Criação de Log.
3. Nem todas as mudanças de configuração possíveis do Logstash foram testadas na pilha ELK gerenciada. Dependendo das mudanças feitas, pode ser necessário excluir completamente e recriar a pilha de Criação de Log para retornar para o estado padrão, perdendo a configuração e os logs coletados anteriormente.
4. Alguns serviços de plataforma podem ser executados em nós separados e não teriam seus logs capturados. Por exemplo:
   - O Vulnerability Advisor é executado em nós com um rótulo separado e não seria capturado.
   - A medição, e até mesmo a própria criação de log, utiliza Daemonsets que são executados em nós do `worker`. Os logs dos componentes em execução nos nós do `trabalhador` não seriam capturados.
5. As pilhas de criação de log adicionais ou os serviços de coleta de log ainda podem capturar os logs da plataforma se eles estiverem configurados para coletar logs dos nós do cluster rotulados.
6. O Kibana na pilha ELK gerenciada pode falhar no carregamento ou não ter acesso aos logs coletados, dependendo das mudanças feitas na configuração.

Local de dados

A implementação do Elasticsearch do IBM Cloud Private é configurada para armazenar documentos no diretório `/var/lib/icp/logging/elk-data` de *cada nó de gerenciamento* no qual ele é implementado. É possível alterar esse caminho

antes da instalação incluindo o seguinte parâmetro no `config.yaml`. O novo caminho deve existir em todos os nós de gerenciamento no cluster.

```
elasticsearch_storage_dir: <your_path>
```

Instalando instâncias de criação de log adicionais

Em liberações anteriores do IBM Cloud Private, era possível instalar instâncias adicionais de criação de log no "modo padrão". Esse modo permitia que pilhas ELK separadas fossem configuradas para coletar logs de diferentes nós e namespaces. No entanto, havia vários drawbacks inerentes ao "modo padrão":

- Nenhuma segurança era necessária entre os componentes ELK. Qualquer solicitação para Logstash ou Elasticsearch em si seria permitida, sem qualquer autenticação ou autorização necessária.
- O serviço Kibana era exposto em uma `nodePort` em cada nó do cluster.
- Nenhuma autenticação era necessária para o acesso ao Kibana. Mesmo usuários que não tivessem login efetuado no IBM Cloud Private poderiam acessar, injetar, mudar ou excluir todos os dados.

Nota: o uso de criação de log com a segurança desativada foi descontinuado no IBM Cloud Private Versão 3.2.0. É recomendável reconfigurar qualquer pilha de criação de log para usar a segurança.

Com o IBM Cloud Private Versão 3.2.0, ainda é possível instalar múltiplas instâncias do gráfico de criação de log com a segurança a seguir ativada:

- Permita bloqueio de acesso direto ao Elasticsearch. Somente solicitações de componentes de pilha ELK, como Logstash e Kibana, são permitidas.
- Permita a requisição de login do IBM Cloud Private antes de acessar o Kibana.
- Permita a requisição de acesso ao namespace antes de acessar o Kibana.
- Inclua links de navegação do console do IBM Cloud Private para instâncias de criação de log adicionais, já que a associação da equipe é necessária para que os links sejam exibidos.

Enquanto o valor `mode` não é mais usado, o comportamento de "modo padrão" antigo ainda é possível. É possível configurar os valores de autenticação do Elasticsearch e do Kibana ao instalar o gráfico. Com os novos recursos, é possível configurar a criação de log de uma maneira segura. Proteja a ocupação variada protegendo instâncias de criação de log separadas por locatário.

Considerações para o IBM Cloud Private com o OpenShift

Ao usar o IBM Cloud Private com o OpenShift, há etapas de configuração extras que são necessárias para assegurar que instâncias de criação de log adicionais possam ser criadas.

1. Inclua os namespaces para instâncias de criação de log adicionais para o recurso `icp-scc SecurityContextConstraints` (SCC).

   ```
   oc edit scc icp-scc
   ```

2. Inclua a configuração a seguir no `SecurityContextConstraints` que você edita na etapa 1.

   ```
   - system:serviceaccount:your_namespace_for_additional_logging:default
   ```

3. Permita que instâncias de criação de log adicionais usem o registro de imagem interno

   ```
   oc policy add-role-to-user system:image-puller
   system:serviceaccount:your_namespace_for_additional_logging:default --namespace=ibmcom
   ```

Instâncias de criação de log seguras adicionais sem ocupação variada

Vários cenários podem requerer a instalação de mais instâncias de criação de log, por exemplo:

- O administrador de um cluster grande pode desejar restringir a criação de log que está instalada com o IBM Cloud Private para coletar logs apenas da plataforma (namespace `kube-system` e nós principal/gerenciamento/proxy). Em seguida, use instâncias de criação de log separadas para coletar logs de aplicativos em diferentes namespaces.
- O administrador de um cluster que hospeda aplicativos sensíveis pode querer instâncias de criação de log separadas com configurações de retenção de dados diferentes.

Nesses cenários, instâncias de criação de log seguras separadas são desejáveis. No entanto, a configuração de segurança é diferente em comparação a um cenário de ocupação variada verdadeiro. Use as mudanças na configuração de chave a seguir:

1. Restrinja a criação de log instalada pelo IBM Cloud Private

Na maioria dos casos em que múltiplas instâncias de criação de log são desejadas, restrinja a criação de log que está instalada pelo IBM Cloud Private. Múltiplas instâncias de criação de log não devem coletar os mesmos logs. É possível configurar seletores de nó e filtros de namespace no gráfico de criação de log para restringir onde o Filebeat tem permissão para executar e de quais logs ele coleta. No tempo de instalação do IBM Cloud Private, inclua estes valores no config.yaml:

```
logging:
  filebeat:
    scope:
      namespaces:
        - kube-system
        - istio-system
        - cert-manager
        - icp-system
      nodes:
        systemfilebeat: "true"
```

- `namespaces:`
  - Controla quais namespaces possuem logs coletados.
- `nós:`
  - Controla onde o Filebeat é executado.
  - Deve-se também aplicar o rótulo `systemfilebeat: true` em cada nó do cluster no qual você deseja que a criação de log do sistema colete logs. Para obter mais informações, consulte Detalhes de criação de log e Documentação do Kubernetes. O valor nodes: que é especificado no arquivo `config.yaml` é usado como o `nodeSelector` para o daemonset Filebeat. Essa opção permite mais flexibilidade para controlar onde o Filebeat é executado.

Dependendo do comportamento que você deseja atingir, é possível usar a configuração `node:` sem especificar namespaces individuais ou usar a configuração `namespaces:` sem um seletor de nó.

Também é possível mudar essas configurações após a instalação extraindo os valores de liberação de `logging` atualizando-os e usando o `helm upgrade`. Certifique-se de configurar também os mesmos valores em `config.yaml` em `logging:` para que eles persistam no upgrade.

2. Instale instâncias de criação de log adicionais

Importante: para permitir que instâncias de criação de log que são instaladas em namespaces diferentes do `kube-system` sejam integradas com autenticação e autorização do IBM Cloud Private, a criação de log requer acesso a um segredo do Kubernetes nomeado `platform-oidc-credentials` e às credenciais nela contidas. Esse segredo deve existir no namespace no qual a criação de log está sendo instalada, mas é criado apenas no namespace `kube-system` pela instalação do IBM Cloud Private. Os segredos não estão acessíveis entre namespaces. Deve-se copiar o segredo para o namespace desejado antes que a criação de log seja iniciada com sucesso.

Para cada instância de criação de log adicional que você instala, especifique a configuração a seguir no arquivo `.yaml` que é usado com o `helm install`:

```
filebeat:
  scope:
    namespaces:
      - the-namespace-to-be-collected-from
    nodes:
      sample-node-label: "true"
  tolerations:
    # See https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
```

Assim como a instância de criação de log do sistema, os parâmetros a seguir permitem controlar onde e como o Filebeat coleta logs. É possível customizar os parâmetros para o propósito específico de cada instância de criação de log.

```
kibana:
  nodeSelector:
  tolerations:
  access: ingress
  ingress:
    # No trailing /
    path: "/sample/kibana"
    # Ingress labels to define link in IBM Cloud Private console navigation menu
    labels:
      inmenu: "true"
```

```
      # if provided, the link will open in a new tab with the target value in the <a> tag
      target: "sample"
  service:
    # No trailing /
    path: "/tenantA/kibana"
    # Service labels and annotations to define link in IBM Cloud Private console navigation menu
    labels:
      inmenu: "true"
      # if provided, the link will open in a new tab with the target value in the <a> tag
      target: "sample"
    # additional annotations to facilitate link rendering in icp console
    annotations:
      # Display name that will show in the menu
      name: "Logging - Sample"
      # Location in navigation
      id: "add-ons"
      # List of roles to be able to see the link
      roles: "ClusterAdministrator,Administrator,Operator,Viewer"
  security:
    authc:
      enabled: true
    authz:
      enabled: false
```

As configurações do Kibana definem como essa instância é acessada por usuários. A configuração `security:` no `kibana:` é a chave para controlar se os usuários devem ter o login efetuado com sucesso no IBM Cloud Private antes de acessarem o Kibana. A configuração `authz:` também é útil na restrição de acesso, mesmo em cenários que não sejam de ocupação variada. Para obter mais informações sobre a configuração `authz:`, consulte Ocupação variada.

```
general:
  environment: IBMCloudPrivate
  clusterDomain: cluster.local
  clusterName: mycluster
  ingressPort: 8443
logstash:
  nodeSelector:
  tolerations:
elasticsearch:
  security:
    authc:
      # what it does: mtls authentication between Logging components
      enabled: true
      # accepted values: searchguard-tls, xpack
      provider: searchguard-tls
    authz:
      # what it does: filter queried log content by the namespaces
      # that the current user has access to
      enabled: true
  client:
    tolerations:
    nodeSelector:
  master:
    tolerations:
    nodeSelector:
  data:
    tolerations:
    nodeSelector:
curator:
  # Controls log retention
  app:
    unit: days
    count: 1
  # Runs at 23:30 UTC daily
  schedule: "30 23 * * *"
  nodeSelector:
  tolerations:
```

Na maioria dos cenários, as configurações `tolerations:` e `nodeSelector:` devem ser idênticas para todos os componentes da pilha de criação de log diferentes do Filebeat, que usa as configurações de `scope`. Para obter mais informações, consulte tolerations e nodeSelectors.

Se não for especificado, cada componente poderá ser executado em qualquer nó que não especifique uma contaminação `NoSchedule`.

Ocupação variada

Em um cluster de múltiplos locatários, cada locatário é provisionado com sua própria instância de criação de log. A configuração para instalação de instâncias de criação de log adicionais ainda se aplica, mas há configurações extras para garantir a segurança e o isolamento de dados:

- Cada usuário do locatário pode ver apenas os links de navegação corretos.
- O Kibana bloqueia as tentativas de acesso por usuários de outros locatários, mesmo se o caminho for acessado diretamente sem o link de navegação.
- Usuários locatários, e até mesmo administradores, não podem ver ou mudar a configuração de suas instâncias de criação de log provisionadas. Essa ação assegura a impossibilidade de mudar as configurações de Filebeat para coleta de logs de outros locatários ou da plataforma compartilhada.

Deve-se usar as mudanças na configuração de chave a seguir, além das configurações para instâncias de criação de log que não sejam de múltiplos locatários.

1. Restrinja a criação de log instalada por IBM Cloud Private para

ocupação variada Em um ambiente de múltiplos locatários, o acesso à instância de criação de log do sistema instalada com o IBM Cloud Private deve ser restrito para que os usuários locatários não possam acessá-la. Restrinja o acesso do Kibana aos usuários que possuem acesso ao namespace `kube-system` no `config.yaml`.

Nota: as amostras a seguir indicam apenas as configurações adicionais e mudadas. As amostras devem ser combinadas com as configurações que não sejam de múltiplos locatários na sintaxe do arquivo `.yaml` apropriada.

```
logging:
  kibana:
    security:
      authz:
        enabled: true
        icp:
          # 1. user is allowed to access the kibana ingress
          #    if namespaces granted to user are listed below
          # 2. when the list below is empty, only cluster admin
          #    can access this kibana ingress
          authorizedNamespaces:
            - kube-system
```

Conforme observado, ao ativar o `authz` e configurar os `authorizedNamespaces` para uma lista vazia, o acesso pode ser restrito apenas ao administrador de cluster.

2. Instale instâncias de criação de log adicionais para ocupação variada

Para cada instância de criação de log do locatário, configure esses valores para restringir o acesso ao Kibana.

Nota: as amostras a seguir indicam apenas as configurações adicionais e mudadas. As amostras devem ser combinadas com as configurações que não sejam de múltiplos locatários na sintaxe do arquivo `.yaml` apropriada.

```
kibana:
  service:
    annotations:
      # show link if user is in any of the teams
      ui.icp.ibm.com/tenant: "sample-tenant-one-team"
  security:
    authz:
      enabled: true
      icp:
        authorizedNamespaces:
          - sample-tenant-one
```

Essas configurações instruem o console do IBM Cloud Private a mostrar o link apenas se o usuário for um membro da equipe indicada. Ele bloqueia o acesso ao Kibana, a menos que o usuário tenha acesso aos namespaces indicados.

Importante: ao provisionar uma instância de criação de log para um locatário, é importante instalar a instância em um namespace separado ao qual os usuários locatários não têm acesso concedido. Esta ação evita que administradores de locatário visualizem ou mudem as configurações. A configuração `authorizedNamespaces` lista os namespaces aos quais os usuários do locatário têm acesso.

Importante: a criação de log requer acesso a um segredo do Kubernetes chamado `platform-oidc-credentials` e às credenciais nele contidas para integração com os componentes de autenticação e autorização da plataforma. Por esse motivo, também é fundamental instalar instâncias de múltiplos locatários de criação de log em namespaces aos quais os usuários locatários não têm acesso concedido.

Escalando serviços de criação de log após a instalação do IBM Cloud Private

Por padrão, uma instância do serviço de criação de log é instalada juntamente com o IBM Cloud Private. Se você precisar de mais capacidade, é possível escalar o serviço de criação de log para mais nós de gerenciamento ou do trabalhador.

Conclua as etapas a seguir para escalar o serviço de criação de log para mais nós.

1. Provisione e inclua nós de gerenciamento no cluster do IBM Cloud Private. Para obter mais informações, consulte Incluindo um nó do cluster do IBM Cloud Private. Leva aproximadamente de 5 a 10 minutos para incluir um nó.

2. Extraia os parâmetros do gráfico de serviço de criação de log existentes.

   - Execute o comando a seguir para extrair os parâmetros do gráfico do Helm do serviço de criação de log e prepare o novo arquivo de parâmetros do gráfico.

     ```
     helm get values your_release_name --tls > values-old.yaml
     ```

   - Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recurso do Kubernetes que você faz usando o comando `kubectl` são substituídos por valores que são definidos nos parâmetros do gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se os manifests de recurso do Kubernetes anteriores foram ajustados, certifique-se de que os mesmos ajustes sejam aplicados ao `values-old.yaml`.

   - Prepare o arquivo de parâmetros do gráfico. Crie o arquivo `values-override.yaml` para incluir os parâmetros a seguir:

     ```
     elasticsearch:
     client:
       replicas: "2" # adjust to total number of nodes
     data:
       replicas: "2" # adjust to total number of nodes
     master:
       replicas: "2" # adjust to total number of nodes
     logstash:
     replicas: "2" # adjust to total number of nodes
     ```

     O número de réplicas é o número total de nós que você incluiu na etapa 1.

3. Prepare volumes persistentes (PV) para armazenar dados do serviço de criação de log em nós do trabalhador recém-incluídos. É possível ignorar esta etapa para nós de gerenciamento recém-incluídos.

   - Prepare o manifest de PV. Crie o arquivo, `pv-logging-datanode-aaa.bbb.ccc.ddd.yaml`, em que `aaa.bbb.ccc.ddd` é o endereço IP do nó do trabalhador. Por exemplo:

     ```
     apiVersion: v1
     kind: PersistentVolume
     metadata:
       name: logging-datanode-aaa.bbb.ccc.ddd
     spec:
       accessModes:
     - ReadWriteOnce
     capacity:
       storage: 20Gi # adjust to your need
     local:
       path: /var/lib/icp/logging/elk-data # adjust to your need
     nodeAffinity:
       required:
         nodeSelectorTerms:
         - matchExpressions:
           - key: kubernetes.io/hostname
                operator: In
                values:
     ```

```
        - <worker_node_name> # set to the name of the ICP worker node, can be found using
kubectl get nodes
persistentVolumeReclaimPolicy: Retain
storageClassName: logging-storage-datanode
```

- Mude as permissões do diretório de dados do serviço de criação de log. No nó do trabalhador, conceda pelo menos a permissão `0755` para o diretório de dados do serviço de criação de log. O local padrão é `/var/lib/icp/logging/elk-data`.

- Execute o comando a seguir para criar um PV.

  ```
  kubectl apply -f pv-logging-datanode-aaa.bbb.ccc.ddd.yaml
  ```

  Repita a etapa 3 para cada nó do trabalhador recém-incluído.

4. Faça download do gráfico do Helm de serviço de criação de log.

   - Identifique a versão do gráfico. As versões do gráfico de criação de log variam com base na versão do produto IBM Cloud Private que está instalada. É possível usar o IBM Cloud Privateconsole de gerenciamento para localizar versões de gráfico no catálogo de serviços. O gráfico de criação de log pode ser identificado pelo nome `ibm-icplogging` no repositório `mgmt-repo`. Também é possível selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.

   - Faça download do arquivo .tar do gráfico. Execute o comando a seguir usando o link local localizado na Etapa 4a.

     ```
     curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz > ibm-
     icplogging-x.y.z.tgz
     ```

5. Escale o serviço de criação de log para usar os nós recém-incluídos. Execute o seguinte comando. Substitua o x.y.z pela versão localizada na Etapa 4.

   ```
   helm upgrade your_release_name ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-
   override.yaml --namespace your_namespace --recreate-pods --force --timeout 600 --tls
   ```

   Por exemplo:

   ```
   helm upgrade logging ibm-icplogging-2.3.0.tgz -f values-old.yaml -f values-override.yaml --
   namespace kube-system --recreate-pods --force --timeout 600 --tls
   ```

6. Aguarde aproximadamente 5 minutos antes de verificar o status do serviço de criação de log.

   ```
   helm status --tls your_release_name
   helm history --tls your_release_name
   ```

Modificando a política de retenção de dados para serviços de criação de log

Os dados do serviço de criação de log são persistidos no disco. Ao longo do tempo, o crescimento dos dados não gerenciados preenche seu espaço em disco. Para manter o tamanho dos dados sob controle, são definidas políticas de retenção. O componente `curator` do serviço de criação de log limpa os dados do log com base em sua política de retenção.

Antes de iniciar, considere as seguintes dicas:

- O `curator` é executado em um planejamento diário. Como resultado, é possível reter um dia extra de dados.
- Quando o índice de dados é removido, um novo índice de dados é recriado quando novos dados são enviados para ele.
- Períodos de retenção aumentados requerem mais espaço em disco e outros recursos de cálculo. Você pode ter problemas de estabilidade quando definir períodos de retenção mais longos sem ajustar seus recursos.

Visualize os índices de log e as políticas de retenção padrão para uma instância de criação de log na tabela a seguir.

Índice do log	Descrição	Período de retenção padrão (dia)
`logstash-*`	Logs de contêiner	`1

`audit-*` | Logs de auditoria do ICP | `1 compliance-*` | Dados do ICP Vulnerability Advisor | `90 secconfig1-*` | Dados do ICP Vulnerability Advisor | `90 vulnerabilityscan-*` | Dados do ICP Vulnerability Advisor | `90 .monitoring` | Dados de monitoramento do Elastic | `7 .monitoring-alerts` | Alerta de monitoramento do Elastic | `7 .watcher-history` | Observador do Elastic | `7`

Conclua as seguintes etapas para customizar a política de retenção de dados.

Nota: Se você seguir o procedimento para atualizar os valores do gráfico, inclua as mesmas linhas que foram incluídas no arquivo `values-override.yaml` em seu arquivo `config.yaml`. Esse método permite que o instalador reaplique as mudanças nas configurações durante as operações de upgrade e retrocesso. O upgrade reconfigura o gráfico para os padrões de gráfico, substituídos pelos valores configurados em seu config.yaml.

1. Extraia os parâmetros do gráfico de criação de log existentes.
   - Extraia os parâmetros do Helm executando o seguinte comando: `helm get values logging --tls > values-old.yaml`
   - Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recurso do Kubernetes que são feitos usando o comando `kubectl` são substituídos por valores definidos nos parâmetros de gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se manifests de recurso do Kubernetes anteriores foram ajustados, certifique-se de aplicar os mesmos ajustes a `values-old.yaml`.
2. Prepare os parâmetros do gráfico.
   - Crie um arquivo `values-override.yaml` para incluir as seguintes configurações de `curator`.

     Nota:

     - `app` refere-se ao log do contêiner.
     - É possível configurar o valor `unit` para valores diferentes de `days`.
     - Evite configurar uma política de retenção para menos de um dia.

     ```
     curator:
      # in this example, container log retention period is set to 2 days
      app:
        count: 2
        unit: days
     ```

     Para obter informações detalhadas de parâmetros, consulte o arquivo leia-me do gráfico do Helm.
3. Faça download do gráfico.
   - Identifique a versão do gráfico.

     As versões do gráfico de criação de log variam com base na versão instalada do IBM Cloud Private. É possível usar o IBM Cloud Privateconsole de gerenciamento para localizar versões de gráfico no catálogo de serviços. O gráfico de criação de log pode ser identificado pelo nome `ibm-icplogging` no repositório `mgmt-repo`. Também é possível selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.
   - Faça download do arquivo .tar do gráfico.

     Execute o seguinte comando usando o link local encontrado na Etapa 3:

     ```
      curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz > ibm-icplogging-x.y.z.tgz
     ```
4. Faça upgrade do gráfico do Helm.
   - Execute o seguinte comando. Substitua `x.y.z` pela versão encontrada na Etapa 3.

     ```
     helm upgrade logging ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-override.yaml --force --timeout 600 --tls
     ```
5. O serviço de criação de log se torna disponível em aproximadamente 5 a 10 minutos. Também é possível verificar o status de upgrade do Helm usando o seguinte comando:

   ```
   helm history --tls logging
   ```

Ativando a segurança para serviços de criação de log

Por padrão, os recursos de segurança de serviços de criação de log são ativados quando são instalados junto com o IBM Cloud Private. É possível instalar mais instâncias de criação de log sem nenhuma segurança. Iniciando com o IBM Cloud Private versão

3.2.0, a criação de log de operação nesse modo foi descontinuada. Deve-se ativar a segurança em instâncias de criação de log existentes. Para a instalação customizada de serviços de criação de log com a segurança desativada, conclua as etapas a seguir para ativar a segurança depois de fazer upgrade para a versão mais recente.

Antes de iniciar, considere as seguintes dicas:

- As instruções a seguir ativam a segurança de uma instância de serviço de criação de log.

  Nota: esta configuração permite a autenticação baseada em certificado entre componentes de criação de log, em vez de exigir que os usuários sejam autenticados no IBM Cloud Private antes que eles acessem o Kibana. Outras documentações para autenticação e autorização do usuário estão disponíveis nos tópicos de criação de log principal.

- Os certificados que são usados pelos serviços de criação de log são excluídos e gerados novamente. Faça backup de seus certificados, conforme necessário.
- O Nome da liberação do Helm, `logging`, é usado nessas instruções. Se você usar um nome de liberação diferente, substitua o nome pelo nome da liberação de criação de log.
- Ao mudar instâncias que são instaladas com o IBM Cloud Private, certifique-se de fazer as mesmas mudanças nos valores na seção `Loggin:` de seu arquivo `config.yaml`. Esta ação assegura que os valores sejam reaplicados ao fazer upgrade ou aplicar correções.

1. Extraia os parâmetros do gráfico de criação de log existentes.
   - Extraia os parâmetros do Helm executando o seguinte comando: `helm get values logging --tls > values-old.yaml`
   - Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recurso do Kubernetes que são feitos usando o comando `kubectl` são substituídos por valores definidos nos parâmetros de gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se manifests de recurso do Kubernetes anteriores foram ajustados, certifique-se de aplicar os mesmos ajustes a `values-old.yaml`.
2. Prepare os parâmetros do gráfico.
   - Crie um arquivo `values-override.yaml` para incluir as seguintes configurações de segurança. Essas configurações são os valores para a criação de log normalmente instalados com o IBM Cloud Private.

     ```
     elasticsearch:
       security:
         authc:
           enabled: true
     security:
       ca:
       # set to `external` to use existing CA stored in Kubernetes secret to generate certs
       # set to `internal` to self-signed CA generated by Logging Service
       origin: external
       external:
         # the secret need to be in the same namespace as the chart release
         secretName: cluster-ca-cert
         # the Kubenetes field name (key) within the specified secret that stores CA cert
         certFieldName: tls.crt
         # the Kubenets field name (key) within the specified secret that stores CA private
     key
          keyFieldName: tls.key
     ```

   - Ajuste as configurações de segurança para o modo `padrão`. Por exemplo, se você deseja uma nova autoridade de certificação, deve-se configurar `security.ca.origin` para `internal` e excluir a seção `external`.

     ```
     security:
         ca:
           origin: internal
     ```

     Para obter informações detalhadas de parâmetros, consulte o arquivo leia-me do gráfico do Helm.
3. Faça download do gráfico.
   - Identifique a versão do gráfico.

     As versões do gráfico de criação de log variam com base na versão instalada do IBM Cloud Private. É possível usar o IBM Cloud Privateconsole de gerenciamento para localizar versões de gráfico no catálogo de serviços. O gráfico de criação de log pode ser identificado pelo nome `ibm-icplogging` no repositório `mgmt-repo`. Também é possível

selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.

- Faça download do arquivo .tar do gráfico.

  Execute o seguinte comando usando o link local encontrado na Etapa 3:

  ```
  curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz > ibm-icplogging-x.y.z.tgz
  ```

4. Remova os certificados de criação de log existentes, se aplicável.

   - Para a criação de log que está instalada com o IBM Cloud Private, execute o comando a seguir:

     ```
     kubectl delete secret logging-elk-certs -n kube-system
     ```

   - Para a criação de log que é instalada separadamente, execute o comando a seguir:

     ```
     kubectl delete secret <logging_helm_release_name>-ibm-icplogging-certs -n <logging_name_space>
     ```

5. Faça upgrade do gráfico do Helm.

   Execute o seguinte comando. Substitua `x.y.z` pela versão encontrada na Etapa 3:

   ```
   helm upgrade logging ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-override.yaml --recreate-pods --force --timeout 600 --tls
   ```

6. O serviço de criação de log se torna disponível em aproximadamente 5 a 10 minutos. Também é possível verificar o status de upgrade do Helm usando o seguinte comando:

   ```
   helm history --tls logging
   ```

Gerenciando a alocação de recurso para serviços de criação de log

O desempenho estável de seus serviços de criação de log depende da alocação de recurso adequada. Deve ser dada consideração cuidadosa ao planejamento da capacidade antes da implementação. Também é importante revisar e ajustar periodicamente as alocações de recursos.

É possível ajustar recursos que são alocados para serviços de criação de log. Alguns exemplos são:

- Heap do pod de dados e tamanho da memória do contêiner do Elasticsearch
- Contagem de réplicas do pod de dados do Elasticsearch
- Contagem de réplicas do pod do Logstash

Antes de iniciar, considere as seguintes dicas:

- O monitoramento do Elastic é um recurso gratuito que pode ser ativado para visualizar o funcionamento do cluster.
- Ocorre uma breve interrupção na disponibilidade do Elasticsearch enquanto as configurações de recursos atualizadas são aplicadas.
- Para reduzir a área de cobertura para uma implementação mínima, os valores padrão no gráfico de serviço de criação de log alocam uma quantidade mínima de recursos. Ajuste a alocação para uso de produção, ou quando outros serviços dependentes estiverem ativados. Por exemplo, serviços como o Vulnerability Advisor ou a criação de log de auditoria.
- O ajuste de desempenho é tanto uma ciência quanto um ofício. Os intervalos de alocação de recursos padrão são destinados a fornecer um intervalo no qual iniciar.
- Para evitar perda de dados ou disponibilidade diminuída, o planejamento cuidadoso é necessário quando você reduz os recursos.

Pontos de ajuste

Diferentes parâmetros de gráfico estão disponíveis para ajustar as alocações de recursos de componentes de serviço de criação de log. O pod de dados do Elasticsearch, o pod do cliente do Elasticsearch e o Logstash são alguns exemplos. As tabelas a seguir listam os parâmetros do gráfico relacionados a recursos para cada componente.

Tabela 1. Nó de dados do Elasticsearch

Parâmetro	Descrição	Padrão	Notes
`elasticsearch.data.replicas`	O número de pods iniciais no cluster de dados	`2`	
`elasticsearch.data.heapSize`	O tamanho de heap da JVM a ser alocado para cada pod de dados do Elasticsearch	`1024m`	Intervalo de valores de produção de `4g` a `12g`
`elasticsearch.data.memoryLimit`	O máximo de memória (incluindo o heap da JVM e o cache do sistema de arquivos) a ser alocado para cada pod de dados do Elasticsearch	`2048Mi`	Intervalo de valores de produção de `8Gi` a `24Gi`, aproximadamente o dobro do tamanho de heap
`elasticsearch.data.storage.size`	O tamanho mínimo do volume persistente	`10Gi`	

Para resiliência e balanceamento de carga, somente um pod de dados do Elasticsearch pode ser executado em cada máquina host. Como resultado:

- Certifique-se de que tenha nós de `gerenciamento` ou do `trabalhador` suficientes no cluster antes de aumentar a contagem de pods de dados

Parâmetro	Descrição	Padrão	Notes
`elasticsearch.client.replicas`	O número de pods iniciais no cluster de clientes	`1	

|
|elasticsearch.client.heapSize|`O tamanho de heap da JVM a ser alocado para cada pod do cliente do Elasticsearch`|512m| `Intervalo de valores de produção de`2g`a`8g|
|elasticsearch.client.memoryLimit|`O máximo de memória (incluindo o heap da JVM e o cache do sistema de arquivos) a ser alocado para cada pod do cliente do Elasticsearch`|1536Mi|`Inclua pelo menos`512Mi` no tamanho de heap|

Parâmetro	Descrição	Padrão	Notes
`logstash.replicas`	O tamanho do cluster de pod inicial	`1	

|
|logstash.heapSize|`O tamanho de heap da JVM a ser alocado para Logstash`|512m| `Intervalo de valores de produção de`2g`a`8g|
|logstash.memoryLimit|`O máximo permitido de memória para Logstash. Inclui o heap da JVM e o cache do sistema de arquivos.`|10246Mi|`Inclua pelo menos`512Mi` no tamanho de heap|

Parâmetro	Descrição	Padrão	Notes
`elasticsearch.master.replicas`	O número de pods iniciais no cluster	`1	

|
|elasticsearch.master.heapSize|`O tamanho de heap da JVM a ser alocado para cada pod principal do Elasticsearch`|1024 | `Intervalo de valores de produção de`1g`a`4g|
|elasticsearch.master.memoryLimit|`O máximo de memória (incluindo o heap da JVM e o cache do sistema de arquivos) a ser alocado para cada pod principal do Elasticsearch`|1536Mi|`Inclua pelo menos`512Mi` no tamanho de heap|

Parâmetro	Descrição	Padrão
`kibana.replicas`	O tamanho do cluster de pod inicial	`1

|
|kibana.maxOldSpaceSize|`Tamanho máximo do espaço antigo (em MB) do mecanismo JavaScript V8`|1024 |
|kibana.memoryLimit|`O máximo permitido de memória para Kibana`|1280Mi`|

1. Extraia os parâmetros do gráfico de criação de log existentes
    1. Extraia os parâmetros do Helm executando o seguinte comando: `helm get values logging --tls > values-old.yaml`
    2. Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recurso do Kubernetes que são feitos usando o comando `kubectl` são substituídos por valores definidos nos parâmetros de gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se manifests de recurso do Kubernetes anteriores foram ajustados, certifique-se de aplicar os mesmos ajustes a `values-old.yaml`.

2. Prepare os parâmetros do gráfico.

    1. Crie um arquivo `values-override.yaml` para incluir os seguintes parâmetros:

```
# This example contains all parameters related to resource allocation
# Only include the parameters that need to be adjusted appropriate to your need
logstash:
  replicas: 1
 heapSize: "512m"
  # at least 0.5g more than heap size
  memoryLimit: "1024Mi"

kibana:
replicas: 1
  # maximum old space size (in MB) of the V8 Javascript engine
  maxOldSpaceSize: "1024"
  # at least 0.25g more  than maximum old space size
  memoryLimit: "1280Mi"

elasticsearch:
  client:
    replicas: 1
    heapSize: "1024m"
    # at least 0.5g more than heap size
    memoryLimit: "1536Mi"

  master:
    replicas: 1
    heapSize: "1024m"
    # at least 0.5g more than heap size
    memoryLimit: "1536Mi"

  data:
    replicas: 2
    heapSize: "1024m"
    # about 2 times the heap size
    memoryLimit: "2048M"
```

3. Faça download do gráfico.

    1. Identifique a versão do gráfico.

        As versões do gráfico de criação de log variam com base na versão instalada do IBM Cloud Private. É possível usar a console de gerenciamento do IBM Cloud Private para localizar versões do gráfico no catálogo de serviços. O gráfico de criação de log é identificado pelo nome, `ibm-icplogging`, no repositório `mgmt-repo`. Também é possível selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.

    2. Faça download do arquivo `.tar` do gráfico.

        Execute o seguinte comando usando o link local encontrado na Etapa 3:

        ```
        curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz > ibm-icplogging-x.y.z.tgz
        ```

4. Faça upgrade do gráfico do Helm.

    Execute o seguinte comando. Substitua `x.y.z` pela versão encontrada na Etapa 3. Remova a opção `--recreate-pods` se não estiver ajustando a contagem de réplicas do pod principal do Elasticsearch.

    ```
    helm upgrade your_release_name ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-override.yaml --namespace your_namespace --recreate-pods --force --timeout 600 --tls
    ```

5. O serviço de criação de log se torna disponível em aproximadamente 5 a 10 minutos. Também é possível verificar o status de upgrade do Helm usando o seguinte comando:

    ```
    helm history --tls logging
    ```

Atualizando filtros de coleção de dados do serviço de criação de log

Por padrão, os logs do contêiner de todas as cargas de trabalho do IBM Cloud Private são coletados. Uma filtragem extra pode ser aplicada ao processo de coleta de log.

Dois tipos de filtros são suportados:

- Filtrar por rótulo do host
- Filtrar por namespace

Conclua as etapas a seguir para atualizar os filtros de coleta de dados.

1. Extraia os parâmetros do gráfico de criação de log existentes.
   - Execute o comando a seguir para extrair os parâmetros do Helm:

     ```
     helm get values logging --tls > values-old.yaml
     ```

   - Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recurso do Kubernetes que são feitos usando o comando kubectl são substituídos por valores que estão definidos nos parâmetros do gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se manifests de recurso do Kubernetes anteriores foram ajustados, certifique-se de aplicar os mesmos ajustes ao values-old.yaml.
2. Prepare os parâmetros do gráfico.
   - Crie um arquivo `values-override.yaml` para incluir as configurações a seguir.

     ```
     filebeat:
       scope:
         # logs are only collected from hosts with all matching label key/value pairs
         # no filtering is applied if left blank
         nodes:
           planet: jupiter
           system: solar
         # logs are only collected from listed namespaces
         # no filtering is applied if left blank
         namespaces:
           - europa
           - ganymede
     ```

     Notas:

     - O `filebeat.scope.nodes` usa o formato do seletor de nó do Kubernetes
     - O `filebeat.scope.nodes` e o `filebeat.scope.namespaces` podem ser usados separadamente. Se ambos os valores forem configurados, os logs que atenderem apenas a ambos os critérios serão coletados.
3. Faça download do gráfico.
   - Identifique a versão do gráfico

     As versões do gráfico de criação de log variam com base na versão instalada do IBM Cloud Private. É possível usar o IBM Cloud Privateconsole de gerenciamento para localizar versões de gráfico no catálogo de serviços. O gráfico de criação de log pode ser identificado pelo nome `ibm-icplogging` no repositório `mgmt-repo`. Também é possível selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.

   - Faça download do arquivo .tar do gráfico Execute o comando a seguir usando o link local localizado na Etapa 3:

     ```
      curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz >
     ibm-icplogging-x.y.z.tgz
     ```

4. Faça upgrade do gráfico do Helm.

   Execute o seguinte comando. Substitua `x.y.z` pela versão encontrada na Etapa 3:

   ```
   helm upgrade logging ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-override.yaml --
   recreate-pods --force --timeout 600 --tls
   ```

5. O serviço de criação de log se torna disponível em aproximadamente 5 a 10 minutos. Também é possível verificar o status de upgrade do Helm usando o seguinte comando:

   ```
    helm history --tls logging
   ```

Ativando o monitoramento do Elastic

O monitoramento do Elasticsearch é um recurso X-Pack gratuito do Elastic que fornece visibilidade operacional. Por padrão, esse recurso é desativado durante a instalação do serviço de criação de log nos modos `gerenciado` e `padrão`. É possível usar essas instruções para ativar o recurso de monitoramento do Elastic para os dois tipos de modo.

Antes de iniciar, considere as seguintes dicas:

- As métricas de funcionamento do Elastic começam a ser acumuladas após a ativação do recurso de monitoramento.
- É possível visualizar o painel de funcionamento do cluster do Elastic e as métricas. Na UI da web do Kibana, selecione **Monitoramento** no menu de navegação.
- O nome da liberação do Helm, `logging` é usado nessas instruções. Se você usar um nome de liberação diferente no modo `padrão`, substitua o nome pelo nome da liberação de `criação de log`.

1. Extraia os parâmetros do gráfico de criação de log existentes.
   - Extraia os parâmetros do Helm executando o seguinte comando: `helm get values logging --tls > values-old.yaml`
   - Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recursos do Kubernetes que são feitos usando o comando `kubectl` são substituídos por valores que estão definidos em parâmetros do gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se manifests de recursos do Kubernetes anteriores foram ajustados, certifique-se de aplicar os mesmos ajustes ao `values-old.yaml`.

2. Prepare os parâmetros do gráfico.

   - Crie um arquivo `values-override.yaml` para incluir os seguintes parâmetros:

     ```
     xpack:
       monitoring: true
     ```

3. Faça download do gráfico.

   - Identifique a versão do gráfico.

     As versões de gráfico de criação de log variam com base na versão do IBM Cloud Private instalada. É possível usar a console de gerenciamento do IBM Cloud Private para encontrar versões de gráfico no catálogo de serviços. O gráfico de criação de log pode ser identificado pelo nome `ibm-icplogging` no repositório `mgmt-repo`. Também é possível selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.

   - Faça download do arquivo .tar do gráfico.

     Execute o seguinte comando usando o link local encontrado na Etapa 3:

     ```
     curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz > ibm-icplogging-x.y.z.tgz
     ```

4. Faça upgrade do gráfico do Helm.

   Execute o seguinte comando. Substitua `x.y.z` pela versão encontrada na Etapa 3:

   ```
   helm upgrade logging ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-override.yaml --recreate-pods --force --timeout 600 --tls
   ```

5. O serviço de criação de log se torna disponível em aproximadamente 5 a 10 minutos. Também é possível verificar o status de upgrade do Helm usando o seguinte comando:

   ```
   helm history --tls logging
   ```

Atualizando licenças do Elastic X-Pack

Uma avaliação de 30 dias do Elastic, Licença do X-Pack é incluída durante a instalação do serviço de criação de log do IBM Cloud Private. Esta licença para teste permite acesso a todos os recursos. Ela é gerada pela instância de criação de log em si e é referida como uma licença para teste gerada automaticamente. O serviço de criação de log pode ser atualizado com licenças customizadas obtidas pelo cliente.

Antes de iniciar, considere as seguintes dicas:

- As licenças do X-Pack vêm em níveis diferentes. Para obter mais informações, consulte Assinaturas.
- Quando sua licença para avaliação grátis expirar e quando você não atualiza o serviço de criação de log com as licenças fornecidas pelo usuário, apenas os recursos `Open Source` permanecem operacionais. Para obter mais informações, consulte Assinatura do Open Source.
- É possível adquirir uma licença `Basic` grátis por meio do Registro anual. Solicite uma licença para o Elastic Versão 5.x. Para obter mais informações, consulte Assinaturas básicas.
- É possível configurar o parâmetro de gráfico, `xpack.license.source`, para especificar uma das licenças a seguir:
  - Para uma licença para teste gerada automaticamente, configure o valor de parâmetro como `selfGenerated`.
  - Para um arquivo de licença externo, configure o valor de parâmetro como `secret`.

Conclua as etapas a seguir para atualizar sua licença.

1. Adquira a licença do Elastic. Com base no conjunto de recursos desejado, adquira uma licença no formato JSON. Para obter mais informações, consulte Assinaturas básicas.

2. Verifique sua licença existente.

   - Efetue login no console do Kibana e clique em **Ferramentas de desenvolvimento**.

   - Na janela de navegação esquerda, prepare a solicitação para obter sua licença:

     ```
     GET /_xpack/license
     ```

   - Clique no triângulo verde para fazer a chamada da API. No namespace da implementação de criação de log, crie um Segredo do Kubernetes que contém a licença:

     ```
     {
       "license": {
         "status": "active",
         "uid": "46e1db38-a100-44bc-9a3a-69542a1aaf7e",
         "type": "trial",
         "issue_date": "2019-01-03T20:16:07.591Z",
         "issue_date_in_millis": 1546546567591,
         "expiry_date": "2019-02-02T20:16:07.591Z",
         "expiry_date_in_millis": 1549138567591,
         "max_nodes": 1000,
         "issued_to": "elasticsearch",
         "issuer": "elasticsearch",
         "start_date_in_millis": -1
       }
     }
     ```

3. No namespace da implementação de criação de log, crie um Segredo do Kubernetes que contém a licença.

   ```
    kubectl create secret generic es-license --from-file=./license.json --namespace=<logging
   namespace>
   ```

   Para o criador de logs do sistema, o `<logging namespace>` é `kube-system`. Para todas as outras liberações de criação de log, ele é o namespace que hospeda a implementação de criação de log.

4. Crie um arquivo `xpack.yaml` que se refira ao Segredo e que ativa os serviços desejados.

   ```
   xpack:
     monitoring: true
     license:
       # accepted values: secret, selfGenerated
       # what it does: determines which xpack license will be used
       source: secret
       secret:
         # the secret needs to be in the same namespace as the chart release
         secretName: es-license
         # the Kubenetes field name (key) within the specified secret
         # that stores license json file
         fieldName: license.json
   ```

   Se `xpack.license.source` for configurado como `selfGenerated`, uma nova licença de 30 dias será solicitada a partir do Elasticsearch toda vez que o pod principal do Elasticsearch for reiniciado.

5. Extraia os parâmetros do gráfico de criação de log existentes.

- Extraia os parâmetros do Helm executando o seguinte comando: `helm get values logging --tls > values-old.yaml`
- Opcionalmente, aplique ajustes anteriores. Todos os ajustes de manifest de recurso do Kubernetes que são feitos usando o comando `kubectl` são substituídos por valores definidos nos parâmetros de gráfico. A contagem de réplicas, o tamanho de heap da JVM ou os limites de memória do contêiner são alguns exemplos. Se manifests de recurso do Kubernetes anteriores foram ajustados, certifique-se de aplicar os mesmos ajustes a `values-old.yaml`.

6. Faça download do gráfico.

   - Identifique a versão do gráfico.

     As versões do gráfico de criação de log variam com base na versão instalada do IBM Cloud Private. É possível usar o IBM Cloud Privateconsole de gerenciamento para localizar versões de gráfico no catálogo de serviços. O gráfico de criação de log pode ser identificado pelo nome `ibm-icplogging` no repositório `mgmt-repo`. Também é possível selecionar **SOURCE & TAR FILES** da console de gerenciamento do IBM Cloud Private para encontrar um link local para um gráfico.

   - Faça download do arquivo .tar do gráfico.

     Execute o comando a seguir usando o link local localizado na Etapa 6:

     ```
     curl -k https://<master ip>:8443/mgmt-repo/requiredAssets/ibm-icplogging-x.y.z.tgz > ibm-icplogging-x.y.z.tgz
     ```

7. Faça upgrade do gráfico do Helm

   Execute o seguinte comando. Substitua `x.y.z` pela versão que você localizou na Etapa 6:

   ```
   helm upgrade logging ibm-icplogging-x.y.z.tgz -f values-old.yaml -f values-override.yaml --recreate-pods --force --timeout 600 --tls
   ```

8. O serviço de criação de log se torna disponível em aproximadamente 5 a 10 minutos. Também é possível verificar o status de upgrade do Helm usando o seguinte comando:

   ```
   helm history --tls logging
   ```

9. Valide sua atualização de licença. Aguarde mais 10 minutos e, em seguida, siga a Etapa 2 para verificar se o Elasticsearch retorna a licença esperada.

Customizando nós Filebeat do IBM Cloud Private para o serviço de criação de log

É possível customizar o Filebeat para coletar logs do sistema ou do aplicativo para um subconjunto de nós.

O serviço de criação de log do IBM Cloud Private usa Filebeat como o agente de coleção de logs padrão.

O Filebeat monitora logs que são produzidos por cargas de trabalho, como contêineres, no mesmo nó. Ele extrai e transfere os logs para o servidor para processamento adicional e armazenamento. Se uma instância do Filebeat não for executada em um nó específico, os logs das cargas de trabalho nesse nó não serão transmitidos para o Elasticsearch.

Por padrão, uma instância do Filebeat em cada nó do IBM Cloud Private coleta todos os logs do aplicativo para o nó. É possível usar os rótulos e seletores do nó para customizar quais nós executam o Filebeat.

1. Instale a interface da linha de comandos `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Obtenha uma lista de nós do IBM Cloud Private executando o comando a seguir:

   ```
   kubectl get nodes --show-labels
   ```

   A saída de comando é semelhante ao texto a seguir:

   ```
   NAME          STATUS    AGE       VERSION                    LABELS
   9.42.24.5     Ready     5h        v1.7.3-11+f747daa02c9ffb
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
   9.42.24.5,role=master
   9.42.30.64    Ready     4h        v1.7.3-11+f747daa02c9ffb
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
   9.42.30.64
   ```

```
9.42.41.109   Ready     4h        v1.7.3-11+f747daa02c9ffb
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
9.42.41.109,management=true
```

3. Rotule os nós nos quais executar o Filebeat. Os rótulos são aplicados executando o seguinte comando. `<node_name>` é o nome de um nó que deve executar Filebeat e `myfilebeat=true` é um rótulo que pode ser usado posteriormente para corresponder a esse nó para a implementação de Filebeat. Qualquer rótulo que se adeque aos padrões do Kubernetes funcionará.

   ```
   kubectl label node <node_name> myfilebeat=true
   ```

4. Obtenha uma lista das instâncias atuais do Filebeat para cada arquitetura. Procure os DaemonSets no namespace que geram as instâncias de Filebeat executando o seguinte comando. `<namespace>` é o namespace a ser procurado.

   ```
   kubectl get ds --namespace=<namespace>
   ```

5. Inclua o rótulo no `nodeSelector` para o Daemonset do Filebeat. O bloco `nodeSelector` informa ao Kubernetes como corresponder os nós do cluster que devem executar um pod específico do Daemonset.

   1. Abra uma definição de DaemonSet de Filebeat em um editor. `<filebeat_daemonset>` é o nome de um DaemonSet de Filebeat ativo e `<namespace>` é o namespace que hospeda o DaemonSet.

      ```
      kubectl edit ds <filebeat_daemonset> --namespace=<namespace>
      ```

   2. Inclua o rótulo `myfilebeat=true` no parâmetro `nodeSelector`. O Kubernetes agora implementará os pods para esse Daemonset somente para nós que corresponderem a todos os critérios de `nodeSelector`. Você deve terminar com algo como o texto a seguir:

      ```
      nodeSelector:
        beta.kubernetes.io/arch: amd64
        myfilebeat: "true"
      ```

   3. Salve o arquivo.

   4. Verifique se o DaemonSet de Filebeat está em execução. `<filebeat_daemonset>` é o nome do DaemonSet de Filebeat que foi modificado e `<namespace>` é o namespace que hospeda o DaemonSet.

      ```
      kubectl get ds <filebeat_daemonset> --namespace=<namespace>
      ```

      Se o DaemonSet do Filebeat atualizado estiver sendo executado adequadamente, as contagens de instâncias desejadas e disponíveis serão correspondentes, conforme mostrado.

6. Repita a etapa anterior para cada DaemonSet Filebeat restante.

Requisitos e recomendações de hardware

O Elasticsearch foi projetado para manipular grandes quantias de dados de log. Quanto mais dados você optar por reter, mais recursos isso exigirá. É possível criar um protótipo do cluster e dos aplicativos antes da implementação de produção integral para medir o impacto dos dados do log em seu sistema. Para obter informações detalhadas de planejamento de capacidade, consulte: {: child} Planejamento da capacidade de criação de log e métricas doIBM Cloud Private

Nota: a alocação de memória padrão para a pilha ELK gerenciada não é destinada ao uso de produção. O uso de produção real pode ser muito maior. Os valores padrão fornecem um ponto de início para a a criação de protótipos e outros esforços de demonstração.

Memória

O tamanho mínimo do disco necessário geralmente está correlacionado à quantia de dados do log bruto gerados para um período de retenção de log integral. Também é uma boa prática considerar os bursts inesperados do tráfego de log. Como tal, considere alocar mais 25-50% de armazenamento. Caso não saiba quantos dados do log são gerados, um bom ponto de início é alocar `100Gi` de armazenamento para cada nó de gerenciamento.

Evite armazenamento NAS, pois você pode ter problemas de latência e pode introduzir um ponto único de falha. Para obter mais informações, consulte Discos.

É possível modificar o tamanho de armazenamento padrão, incluindo o seguinte bloco no arquivo `config.yaml`:

```
elasticsearch_storage_size: <new_size>
```

Memória

A quantidade de memória que é exigida por cada pod é diferente, dependendo do volume de logs a serem retidos. É impossível prever as necessidades de memória exata, mas é possível iniciar com as diretrizes a seguir:

- Aloque 16, 32 ou até 64 GBs de memória para cada pod *data*.
- Aloque 8, 16 ou 32 GBs de memória para cada pod *client* e *master*.

A memória insuficiente pode levar ao excesso de coleta de lixo, o que pode incluir consumo significativo de CPU pelo processo Elasticsearch.

As configurações de alocação de memória padrão para a pilha ELK gerenciada podem ser modificadas pela inclusão e customização das linhas a seguir em `config.yaml`. Em geral, o valor `heapSize` é igual a aproximadamente metade do valor `memoryLimit` geral do pod.

Nota: o tamanho de heap é especificado usando as unidades JDK: g|G, m|M, k|K. O limite de memória do pod é especificado em unidades do Kubernetes: G|Gi, M|Mi, K|Ki.

```
logging:
  logstash:
    heapSize: "512m"
    memoryLimit: "1024Mi"
  elasticsearch:
    client:
      heapSize: "1024m"
      memoryLimit: "1536Mi"
    data:
      heapSize: "1536m"
      memoryLimit: "3072Mi"
    master:
      heapSize: "1024m"
      memoryLimit: "1536Mi"
```

CPU

O uso da CPU pode flutuar dependendo de vários fatores. Consultas longas ou complexas tendem a requerer a maior parte da CPU. Planeje-se com antecedência para assegurar que você tenha a capacidade necessária para manipular todas as consultas que sua organização precisa.

Planejamento da capacidade de criação de log e métricas do IBM Cloud Private

Flexibilidade

O planejamento responsável prepara as empresas para maximizar os recursos de hardware para cargas de trabalho, enquanto minimiza os recursos necessários para resolução de problemas e análise de histórico. Alocar recursos suficientes para a captura, o armazenamento e o gerenciamento de criação de log e métricas é crucial, especialmente sob condições estressantes. Os dados geralmente fornecem a chave para a análise de eventos passados e a previsão de requisitos futuros.

Não há nenhuma recomendação universal com custo reduzido para a captura, o armazenamento e o gerenciamento de logs e métricas disponível, mas o guia a seguir fornece alguns insights com base em observações do comportamento da carga de trabalho no IBM Cloud Private. Recomenda-se testar as cargas de trabalho sob condições de inatividade e estresse e usar essas informações para prever os recursos de hardware necessários para o gerenciamento de curto e longo prazo.

Serviços gerenciados

O IBM Cloud Private fornece um conjunto de serviços gerenciados que podem ser implementados nos nós de gerenciamento. Os recursos que são alocados para esses nós precisam refletir se esses serviços gerenciados devem manipular todo o tráfego de log e monitoramento para todo o cluster doIBM Cloud Private. No entanto, o gerenciamento central nem sempre é obrigatório.

Vários serviços gerenciados também incluem um gráfico Helm configurável semelhante no catálogo. Por meio de uma combinação de rótulos de nó e opções do gráfico Helm, os usuários podem implementar serviços que focam em cargas de trabalho e

namespaces específicos. Essa abordagem requer uma consideração mais detalhada de como a nuvem é usada, mas tem o benefício potencial de reduzir o trabalho geral de um serviço central.

Criação de log e monitoramento em Kubernetes

As cargas de trabalho são registradas e monitoradas em dois níveis. O nível padrão, e mais comuns, manipula cargas de trabalho como caixas pretas. Os serviços de criação de log e monitoramento manipulam as cargas de trabalho como caixas pretas porque elas leem e medem somente os dados que são visíveis de fora do contêiner do Docker em si. Nenhum conhecimento da carga de trabalho em si é necessário para que os serviços de criação de log e monitoramento funcionem. O segundo nível é uma integração de carga de trabalho mais profunda.

Cargas de trabalho como caixas pretas

Alguns metadados para pods, contêineres, namespaces e outras cargas de trabalho não estão disponíveis no Elasticsearch ou no Prometheus. A maioria dos metadados que está visível para usuários do Kubernetes não está visível para os serviços gerenciados de criação de log e monitoramento. No caso da criação de log, por exemplo, as consultas do Elasticsearch retornam esses campos como `kubernetes.pod`, `kubernetes.container_name` e `kubernetes.namespace`. Esses valores de campo não são recuperados por meio de uma API, mas em vez disso são extraídos dos nomes de arquivos de log em si. O Kubernetes cria convenientemente symlinks para os logs do Docker subjacente e o nome de cada symlink é estruturado com o nome do pod, o nome do contêiner, o ID do contêiner e o namespace. Sem armazenar essas informações no nome do arquivo, o serviço de criação de log não seria capaz de preencher esses valores.

O serviço de monitoramento gerenciado possui ainda mais restrições. Os coletores de métrica extraem informações do nó em execução sobre quais processos (incluindo contêineres do Docker) estão usando quais recursos e em qual grau. Na maioria dos casos, os coletores não têm uma percepção mais profunda no Kubernetes ou na carga de trabalho para detectar metadados sobre a origem dos dados coletados, que limita o grau no qual a filtragem e a correlação podem ser executadas. Como resultado, o menor escopo que pode ser configurado para coleta pela pilha de monitoramento gerenciado é o nó IBM Cloud Private.

Integração de carga de trabalho mais profunda

Apesar da limitação de caixa preta para muitos aplicativos, algumas cargas de trabalho integram as APIs de coleção e os recursos de compartilhamento de log. Por exemplo, elas podem usar um sidecar Filebeat para enviar arquivos de log de um contêiner ou o middleware pode implementar uma API para coletar métricas detalhadas. Também é possível encontrar maneiras de combinar essas técnicas com contêineres de caixa preta para expressar a coleção de metadados mais ricos.

Resumo de impacto do hardware

Pilha ELK

O IBM Cloud Private implementa a pilha ELK conforme a seguir:

1. Um daemonset Filebeat que é executado em cada nó
2. Um único pod do Logstash, que pode ser escalado
3. Um pod principal do Elasticsearch, que coordena o gerenciamento do cluster do Elasticsearch
4. Um pod do cliente do Elasticsearch que implementa a interface REST para todos os logs recebidos do Logstash e consultas do Kibana
5. Dois pods de dados do Elasticsearch para processar e armazenar todos os dados do log
6. Um pod Kibana opcional

Em geral, as partes da pilha que requerem a maioria dos recursos são Logstash e os nós de dados Elasticsearch. Os nós Elasticsearch principal e cliente são capazes de manipular grandes quantidades de tráfego com uso mínimo de recursos. O Filebeat também é muito eficiente, usando recursos triviais.

A configuração Logstash de única instância padrão pode manipular centenas de entradas de log por segundo, com uso da CPU que cresce em uma taxa de cerca de um núcleo por 150 - 200 registros por segundo. No entanto, em um certo ponto, dependendo da capacidade de rede, que possivelmente é cerca de 700 registros por segundo, o volume do tráfego de log começa a degradar o desempenho da rede. Esse aumento do volume tem um efeito de correlação em aplicativos que são executados nos nós afetados. Em geral, se você espera altas taxas de tráfego de log, distribua-o pelo maior número possível de nós ou divida as cargas de trabalho em múltiplos clusters do IBM Cloud Private. Felizmente, o Filebeat e o Logstash são excelentes para rastreamento e recuperação de erros de conectividade com perda mínima de dados quando as taxas de tráfego normais são continuadas.

Os nós de dados Elasticsearch geralmente usam menos CPU do que Logstash, mas requerem mais atenção ao disco e RAM. De acordo com a companhia Elastic, logs armazenados com Elasticsearch geralmente requerem armazenamento semelhante aos

arquivos de log brutos em si. O consumo de memória pode aumentar tanto quanto 15 - 20% dos logs armazenados. Ajustes na configuração do Elasticsearch afeta potencialmente esses números, mas é importante enfatizar que o heap da JVM representa apenas um aspecto da memória total que o nó de dados usa.

A criptografia coloca naturalmente uma carga mais pesada na CPU, particularmente à medida que o tráfego de logs e de consultas aumenta. Os modelos mais recentes de CPU são capazes de manipular a criptografia de forma mais eficiente e podem compensar a necessidade de mais hardware, mas mais hardware será necessário. Incluir memória extra também é recomendado, pois os plug-ins que fornecem a criptografia TLS podem impor alguma sobrecarga.

Prometheus

Por várias razões, o Prometheus retém todas as métricas coletadas na memória por um período de 2 horas. A quantidade de RAM que Prometheus requer depende de inúmeros fatores, incluindo:

1. O número de nós no cluster do IBM Cloud Private
2. O número de cargas de trabalho durante condições de operação de pico
3. A frequência com a qual as métricas são coletadas

O terceiro fator é o único que precisa de consideração cuidadosa. Dobrar o tempo entre a coleta de métricas (por exemplo, aumentando de cada 15 segundos para cada 30 segundos) diminui o uso de memória do Prometheus pela metade, mas também reduz a granularidade dessas métricas. Um elemento chave do processo de planejamento é uma avaliação dos requisitos para a coleta de métricas, tanto para resolução de problemas quanto para análise preditiva.

Impacto detalhado

Os planejadores precisam considerar os fatores a seguir ao estimar os recursos para gerenciar dados de criação de log e monitoramento:

1. Se a criptografia de dados em movimento é necessário
2. Se deve coletar centralmente os logs e métricas para os serviços gerenciados de criação de log e monitoramento
3. O número de logs de fluxo de instâncias Filebeat (daemonset ou sidecar do nó) para o cluster Logstash
4. O volume de logs que são gerados pelas cargas de trabalho
5. Bursts antecipados de carga, resultando em maior volume de log
6. A granularidade das métricas a serem coletadas
7. Requisitos de criação de log e retenção de medição
8. Desempenho da consulta do Elasticsearch

Criptografia

Embora o TLS não seja suportado para muitos dos coletores usados pelo Prometheus, todos os outros tráfegos de dados em movimento para monitoramento e criação de log podem ser criptografados. Embora a CPU no geral seja afetada, as CPUs recentes são mais eficientes, mas também trabalham mais arduamente. A pilha ELK, em particular, pode incorrer em sobrecarga de memória como resultado do plug-in ativar a criptografia. Os nós que têm restrições mais rígidas de RAM podem encontrar problemas de estabilidade e desempenho.

Em geral, se você planeja ativar a criptografia, considere aumentar o número de CPUs alocadas por uma quantia proporcional ao volume de log geral.

Coleta centralizada

O serviço de criação de log gerenciado está devidamente configurado para manipular cargas relativamente pequenas, embora ele possa manipular cargas de trabalho muito maiores. No entanto, à medida que a carga aumenta, o uso de CPU, disco e RAM também aumenta.

O monitoramento de recursos geralmente requer mais RAM do que CPU. O Prometheus retém todas as métricas na memória por um período de 2 horas não configurável, por razões que incluem responsividade para consultas sensíveis ao tempo e operações de disco em massa mais eficientes. O resultado é que mais cargas de trabalho requerem mais recursos para o medidor, gerando mais métricas nessas 2 horas. Se a disponibilidade de memória for uma preocupação nos nós de gerenciamento e o IBM Cloud Private estiver executando muitas cargas de trabalho, será possível restringir os nós nos quais a pilha de monitoramento gerenciado coleta métricas.

Número de instâncias Filebeat

Esse fator impacta enormemente o Logstash. Por padrão, o IBM Cloud Private implementa um daemonset Filebeat para cada nó e cada daemonset é transmitido de volta para o serviço de criação de log gerenciado. Você também cria uma instância Filebeat para cada pod que usa um sidecar Filebeat para transmitir logs que estão armazenados no contêiner. À medida que o número de instâncias Filebeat cresce e conforme o tráfego de log aumenta, considere revisar a taxa de uso de CPU Logstash no Grafana. Quando a instância Logstash começa a usar um núcleo de CPU total, é uma boa hora para considerar incluir outra réplica para o cluster Logstash.

Volume de log

Alto volume de log geralmente impacta o desempenho de RAM e de rede mais do que outros fatores. Para alguns ambientes, isso pode ser dezenas de entradas de log por segundo. Em outros, ele pode subir para milhares de entradas por segundo. Algumas medições indicam que o desempenho da rede pode começar a diminuir quando o volume de log atinge a taxa de 1.000 entradas por segundo, mas os resultados individuais variam.

O volume de log normalmente aumenta por meio de um aumento na contagem de cargas de trabalho ou de um aumento na taxa de saída de cargas de trabalho. Conforme mencionado em outros lugares, o Prometheus requer mais RAM para armazenamento temporário de métricas à medida que o número de recursos, inclusive de cargas de trabalho, cresce.

Bursts de Tráfego

Embora os recursos necessários para monitorar a coleta de dados estejam relativamente estáveis por meio de bursts de tráfego, o volume de saída de log pode crescer significativamente. Deve-se considerar cuidadosamente o núcleo da CPU, a memória e a alocação de disco para manipular bursts inesperados no tráfego.

Granularidade das métricas

Consulte a seção Prometheus.

Retenção de dados

A configuração padrão para as pilhas gerenciadas de criação de log e monitoramento retém dados por apenas um dia. Toda noite por volta da 0h, os dados antigos são excluídos. É possível modificar essas configurações. Consulte IBM Cloud Private log. Mas ao modificar essas configurações, deve-se considerar algumas implicações importantes ao optar por reter dados por períodos de tempo mais longos.

O Elasticsearch quebra os dados em chunks, que são conhecidos como *índices*. Cada índice é composto por três partes: primeiro, os dados no disco; segundo, um cache na memória do Elasticsearch; e terceiro, um cache Lucene (mecanismo de procura). A pilha ELK gerenciada define cada índice como um dia. Para os logs retidos de cada dia, os requisitos de disco e cache são acumulados. O tamanho de cada cache se correlaciona ao volume de logs que são gerados para esse cache, às vezes até 15%. Em outras palavras, é possível que para cada 100 GB de logs que são armazenados, o Grafana possa relatar nós de dados Elasticsearch que estão usando o máximo de 15 GB de RAM. Essa proporção de uso não é uma regra universal, mas isso demonstra a necessidade de teste para determinar a carga de recursos que é criada pelas cargas de trabalho que você executa no IBM Cloud Private.

Desempenho da consulta

Alguns cenários de planejamento podem colocar restrições sobre o prazo no qual as consultas Elasticsearch (sejam elas executadas por meio do Kibana ou diretamente por meio da API de REST Elasticsearch) devem ser concluídas. Algumas consultas são complexas e outras são sensíveis ao tempo, portanto, os resultados devem estar disponíveis dentro de um limite específico. Nesses casos, é ainda mais importante planejar mais memória, mas também discos mais rápidos. Os discos de estado sólido (SSDs) geralmente têm um custo mais alto, mas eles fornecem o desempenho de E/S que permite que os sistemas estejam em conformidade com os limites de consulta rígidos.

A combinação de mais RAM, facilitando caches maiores na memória e velocidades de disco mais rápidas pode ajudar, mas pode não ser suficiente. Conforme descrito em outras seções, o tráfego de log extremamente alto pode afetar a qualidade de rede, o que pode afetar a responsividade da consulta. Outros fatores que não estão relacionados à pilha ELK gerenciada ou até mesmo não relacionados ao IBM Cloud Private, podem afetar a responsividade da consulta. O teste antecipado ajuda a identificar quaisquer gargalos que possam surgir.

Plano para falha

Em muitos casos, os dados de criação de log e monitoramento são arquivados para auditoria, mas raramente para revisão ativa. É tentador alocar recursos menos caros para gerenciar esses dados, em vez de focar esse hardware nas cargas de trabalho. Mas uma das ideias fundamentais por trás do Kubernetes é que os desenvolvedores de aplicativos e administradores de sistemas devem

projetar seus softwares para que falhem sem danos e se recuperem rapidamente da falha. É por essa razão que não há nenhum comando para reiniciar os pods: é possível apenas excluir o pod e esperar que o Kubernetes o recrie.

Assim, o planejamento adequado considera não só o comportamento padrão diário das cargas de trabalho que são executadas no IBM Cloud Private, mas também o que é necessário quando um ou mais sistemas ou cargas de trabalho falham catastroficamente. É nesses momentos que o acesso aos dados de métrica e criação de log é mais crucial.

Guia de ferramentas da CLI

O IBM Cloud Private inclui várias opções de interface da linha de comandos (CLI).

O IBM Cloud Private inclui uma CLI para gerenciamento de seu cluster e execução de várias operações. Também é possível usar a CLI do Helm, a CLI do Kubectl e outras ferramentas de CLI com o IBM Cloud Private.

- Gerenciando seu cluster com a CLI do IBM® Cloud Private (cloudctl)
- Instalando a CLI do Kubernetes (kubectl)
- Instalando a CLI do Helm (leme)
- Instalando a CLI do Istio (istioctl)
- Instalando a CLI do Calico (calicoctl)

Gerenciando seu cluster com a CLI do IBM Cloud Private (cloudctl)

É possível usar a interface da linha de comandos (CLI) do IBM Cloud Private para visualizar informações sobre seu cluster, gerenciar seu cluster, instalar gráficos e cargas de trabalho Helm e muito mais.

- Instalando a CLI do IBM Cloud Private
- IBM Cloud Private Comandos do Catálogo CLI (Catálogo)
- IBM Cloud Private comandos gerais da CLI (cloudctl)
- IBM Cloud Private Comandos de Gerenciamento de Cluster CLI (cm)
- IBM Cloud Private Comandos de gerenciamento de chaves da API de serviço da CLI (iam)
- Comandos multicluster da CLI do IBM Cloud Private
- Comandos de medição da CLI do IBM Cloud Private (metering)
- IBM Cloud Private Comandos de Gerenciamento de Senha da CLI (pm)

Instalando a CLI do IBM® Cloud Private

É possível instalar e usar a interface da linha de comandos (CLI) do IBM Cloud Private para gerenciar um ou vários clusters.

Depois de instalar o IBM Cloud Private, é possível instalar a CLI no Windows™, Linux® ou macOS.

Para configurar a CLI do IBM Cloud Private}, conclua as etapas a seguir:

1. Na página *Introdução* da console de gerenciamento do IBM Cloud Private, clique em **Instalar ferramentas da CLI**.
2. Expanda **Instalar a CLI do IBM Cloud Private**. Leia o texto e, em seguida, copie e execute o comando curl para seu sistema operacional. Continue o procedimento de instalação na documentação do produto.

   Escolha o comando curl para o sistema operacional aplicável. Por exemplo, é possível executar o comando a seguir para macOS:

   ```
   curl -kLo <install_file> https://<Cluster Master Host>:<Cluster Master API
   Port>/api/cli/cloudctl-darwin-amd64
   ```

3. Depois de executar o comando curl para seu sistema operacional, continue a instalar a CLI do IBM Cloud Private.

   Para instalar a CLI do IBM Cloud Private, execute o comando que corresponde ao sistema operacional de seu computador cliente, em que `<path_to_installer>` é o caminho para o diretório no qual o arquivo da CLI foi transferido por download e `<install_file>` e o nome do arquivo transferido por download.

   - Por exemplo, para Linux e macOS, execute os comandos a seguir para mudar e mover o arquivo. Lembre-se de que o comando curl para seu cluster está localizado na console de gerenciamento:

     ```
     chmod 755 <path_to_installer>/<install_file>
     ```

```
sudo mv <path_to_installer>/<install_file> /usr/local/bin/cloudctl
```

- Para o Windows, renomeie o arquivo transferido por download para `cloudctl` e coloque o arquivo na variável de ambiente PATH.

4. Confirme se a CLI do IBM Cloud Private está instalada:

   ```
   cloudctl -- help
   ```

   O uso da CLI do IBM Cloud Private é exibido.

5. Configure a CLI do `kubectl`. Consulte Instalando a CLI do Kubernetes (kubectl) para obter instruções de instalação.

6. Efetue login em seu cluster com o comando a seguir, em que `<Cluster Master Host>` é o nome do host ou o endereço IP externo para o nó principal ou principal líder.

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

Agora é possível usar a CLI do IBM Cloud Private para visualizar informações sobre seu cluster e gerenciar seus clusters.

IBM Cloud Private Comandos do Catálogo CLI (Catálogo)

Aprenda sobre os comandos `cloudctl catalog` que podem ser executados para gerenciar seus gráficos Helm.

Catálogo cloudctl

add-repo do catálogo cloudctl

Incluir um repositório do Helm.

Exemplo

```
cloudctl catalog add-repo --name <repo-name> --url <repo-url>

OPTIONS:
   --name value  Name of the Helm repository
   --url value   URL of the Helm repository
```

Gráficos de catálogo cloudctl

Liste gráficos Helm por meio dos repositórios do Helm do cluster.

Exemplo

```
Gráficos de catálogo cloudctl [ -- repo HELM_REPO_NAME ]

OPTIONS:
   --json                 Display output in JSON format
   --repo value, -r value  The name of the target Helm repository. Execute 'cloudctl catalog repos'
para listar os repositórios.
   -s                     Do not show the column headers in the output
```

arquivo create-archive do cloudctl

Cria um archive que contém imagens do Docker e gráficos do Helm para distribuição para clusters sem acesso à Internet.

Exemplo

```
cloudctl catalog create-archive --archive ARCHIVE_TO_CREATE [--chart CHART_ARCHIVE_OR_DIR | --spec
SPEC_FILE]] [--values VALUES_YAML_FILE]
    Before you create the archive, you must set up Docker. See
https://www.ibm.com/support/knowledgecenter/SSBS6K_3.2.0/manage_images/using_docker_cli.html

OPTIONS:
   --architectures value            The result only includes images for the architectures specified
[amd64, ppc64le, s390x, etc].
   --archive value, -a value        The path to write the archive tgz.
   --batch-images, -b               Save images in batch. All images are saved together, which saves
space through the de-duplication of shared layers of images. Por padrão, as imagens do Docker são
salvas em arquivos separados. WARNING: This flag cannot be used if the version of the cluster to
load the archive into is version 3.1.0 or earlier.
   --chart value, -c value          The path to the Helm chart tgz archive or chart directory.
   --pak-extension value, -p value  The path to the Pak extension tgz archive or directory.
   --skip-cleanup                   Do not delete images pulled to the local Docker repository
during archive creation.
   --skip-pull                      Do not pull images. As imagens devem estar no repositório local
do Docker.
   --spec value, -s value           The path to a spec file to create an archive of charts.
   --values value, -f value         The path to an optional values.yaml file containing values to
override in the chart values.yaml file.
```

delete-chart do catálogo cloudctl

Exclui um gráfico Helm do registro interno do IBM Cloud Private.

Exemplo

```
cloudctl catalog delete-chart --name HELM_CHART_NAME [--repo HELM_REPO_NAME] [--version
HELM_CHART_VERSION]

OPTIONS:
   --name value, -n value     The name of the Helm chart to delete
   --repo value, -r value     The name of the target Helm repository. Execute 'cloudctl catalog
repos' para listar os repositórios.
                              If not specified, the 'local-charts' repository is used.
   --version value, -v value  The version of the Helm chart to delete
```

delete-repo do catálogo cloudctl

Excluir um repositório do Helm.

Exemplo

```
cloudctl catalog delete-repo NAME [-f]

OPTIONS:
   -f  Delete a Helm repository with force without any user prompts
```

load-archive do catálogo cloudctl

Carregue as imagens do Docker e os gráficos Helm a partir de um arquivo archive de catálogo ou de um caminho de archive expandido.

Exemplo

```
cloudctl catalog load-archive --archive ARCHIVE [--registry REGISTRY] [--repo HELM_REPO_NAME]
    Before you load the archive, you must set up Docker. See
https://www.ibm.com/support/knowledgecenter/SSBS6K_3.2.0/manage_images/using_docker_cli.html

OPTIONS:
   --archive value, -a value  The path to a catalog archive tgz file or a path to an expanded
archive.
   --password value           Docker password. Isso é necessário se você está usando macOS e também
pode ser configurado no ambiente como DOCKER_PWD.
   --registry value           The registry that the docker image will be pushed to. Exemplo:
'mycluster.icp:8500/namespace'.
                              If not specified, images are uploaded to the registry associated with
the default cluster CA domain and the current targeted namespace.
```

```
                              Run 'cloudctl target' to see the current targeted namespace.
   --repo value, -r value     The name of the target Helm repository. Execute 'cloudctl catalog
repos' para listar os repositórios.
                              If not specified, the 'local-charts' repository is used.
   --username value           Docker username. Isso é necessário se você está usando macOS e também
pode ser configurado no ambiente como DOCKER_USER.
```

cloudctl catalog load-chart

Carrega um archive do gráfico Helm para um cluster do IBM Cloud Private.

Exemplo

```
cloudctl catalog load-chart --archive HELM_CHART_ARCHIVE [--repo HELM_REPO_NAME]
    Before you load a Helm chart, you may need to push your images to the private registry. See
https://www.ibm.com/support/knowledgecenter/SSBS6K_3.2.0/manage_images/using_docker_cli.html

OPTIONS:
   --archive value, -a value  The path to the Helm chart tgz archive or chart directory
   --registry value           If the chart images are loaded into a private registry, use this flag
to specify the registry and the load will modify the values.yaml of the chart to reference the
private registry, example: 'mycluster.icp:8500/namespace'
   --repo value, -r value     The name of the target Helm repository; run 'cloudctl catalog repos'
to list the repositories
                              If not specified, the 'local-charts' repository is used
   --trim-images              If using --registry, this option will first remove any registry or
namespaces from images in the chart; with a --registry of 'mycluster.icp:8500/default', an image
value of 'ibmcom/icp-image:latest' becomes 'mycluster.icp:8500/default/icp-image:latest'
```

Carregamento do Catálogo cloudctl-imagens

Carrega imagens do Docker em um registro do Docker interno do IBM Cloud Private.

Exemplo

```
cloudctl catalog load-images --registry REGISTRY [--repo IMAGE_REPO_NAME] [--repo-pattern
IMAGE_REPO_REGEXP] [--tag IMAGE_TAG_NAME] [--tag-pattern IMAGE_TAG_REGEXP] [-f]

OPTIONS:
   -f                    Force the load of images with no user prompts
   --registry value      The host name of the registry to load the images too
   --repo value          The exact name of the repository of the images to load
   --repo-pattern value  The pattern to match the repository of the images to load
   --tag value           The exact tag name of the images to load
   --tag-pattern value   The pattern to match the tag of the images to load
```

repos do catálogo cloudctl

Liste repositórios do Helm.

Exemplo

```
repos do catálogo cloudctl

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

sincronização do catálogo cloudctl

Sincronize gráficos em todos os repositórios do Helm em um cluster do IBM Cloud Private.

Exemplo

```
sincronização do catálogo cloudctl

OPTIONS:
   --repo value, -r value  The name of the target Helm repository. Execute 'cloudctl catalog repos'
para listar os repositórios.
```

IBM Cloud Private comandos gerais da CLI (cloudctl)

Saiba mais sobre os comandos `cloudctl` gerais que podem ser executados para acessar o cluster do IBM® Cloud Private.

cloudctl api

Visualize o terminal da API e a versão da API para o serviço.

Exemplo

```
cloudctl api

OPTIONS:
   --ca  Output the cluster CA certificate in PEM format
```

conclusão do cloudctl

Generate an auto-completion script for the specified shell (bash or zsh).

Exemplo

```
Generate an auto-completion script for the specified shell (bash or zsh).

Esse comando pode gerar autocompleções de shell. Por exemplo:

  $ cloudctl completion bash

Origem da saída para conclusão.

  $ cloudctl completion bash > cloudctl_complete.sh; source cloudctl_complete.sh
```

configuração cloudctl

Grave valores padrão para a configuração.

Exemplo

```
cloudctl config [--http-timeout TIMEOUT_IN_SECONDS] [--trace true | false | path/to/file] [--color
true | false] [--locale (LOCALE | CLEAR)] | --list

OPTIONS:
   --color value         Enable or disable color
   --http-timeout value  Timeout for HTTP requests (default: 60)
   --list                List all configurations
   --locale value        Set default locale; if LOCALE is CLEAR, previous locale is deleted
   --trace value         Trace HTTP requests
```

cloudctl helm-init

Imprime a configuração da definição de HELM_HOST para o Helm.

Exemplo

```
cloudctl helm-init

      EXAMPLE (Linux):
      eval "$(cloudctl helm-init)"
```

```
      EXAMPLE (Windows):
      cloudctl helm-init > helm_host.cmd && call helm_host.cmd && del helm_host.cmd
```

cloudctl login

Efetuar login do usuário.

Exemplo

```
cloudctl login [-a CLUSTER_URL] [-u USERNAME] [-p PASSWORD] [-c ACCOUNT_ID or ACCOUNT_NAME] [-n
namespace] [--skip-ssl-validation]

      WARNING: It is best practice to avoid providing your password in the command line option. Sua
senha pode ficar visível para outras pessoas e pode ser registrada em seu histórico de shell.

      EXAMPLE:
      cloudctl login
          To interactively provide your user name and password, omit the user name and password
options.
      cloudctl login -u name@example.com -p pa55woRD
          Specify your username and password as arguments.
      cloudctl login -u name@example.com -p "my password"
          Use quotation marks (") around passwords that have spaces.
      cloudctl login -u name@example.com -p "\"password"\"
          If your password contains quotation mark characters (“), use backslash characters (\) to
escape them.


OPTIONS:
   -a value               The URL that you use to access the management console, such as
https://<ip_address>:8443
   -c value               Account ID or name
   -n value               Name of a namespace
   -p value               Password
   --skip-helm-config     Bypass helm configuration
   --skip-kubectl-config  Bypass kubectl configuration if kubectl is not installed
   --skip-ssl-validation  Bypass SSL validation of HTTP requests; this option is not recommended
   -u value               Username
```

logout cloudctl

Efetuar logout do usuário.

Exemplo

```
logout cloudctl
```

destino cloudctl

Configure ou visualize o namespace de destino.

Exemplo

```
cloudctl target [-n NAMESPACE] [--list]

OPTIONS:
   --list                       List all available namespaces
   --namespace value, -n value  Name of the namespace to target
```

cloudctl tokens

Exiba os tokens OAuth para a sessão atual. Execute `cloudctl login` para recuperar os tokens.

Exemplo

```
cloudctl tokens [ --access | --id ]

OPTIONS:
   --access, -a  Only print the access token value
   --id, -i      Only print the ID token value
```

versão cloudctl

Verifique a compatibilidade da versão da CLI e da API.

Exemplo

```
cloudctl version
```

IBM Cloud Private Comandos da CLI cm (cm)

Aprenda sobre os comandos `cloudctl cm` que podem ser executados para gerenciar seu cluster.

cloudctl cm

- cloudctl cm psp-default-get
- cloudctl cm psp-default-set

cloudctl cm psp-default-get

Obtenha a política de segurança de pod do Kubernetes padrão.

Exemplo

```
cloudctl cm psp-default-get
```

cloudctl cm psp-default-set

Configure a política de segurança de pod do Kubernetes padrão.

Exemplo

```
cloudctl cm psp-default-set restricted|unrestricted
```

IBM Cloud Private Comandos do CLI iam (iam)

Aprenda sobre os comandos `cloudctl iam` que podem ser executados para gerenciar as chaves API, IDs e políticas de serviço.

Cloudctl iam

- cloudctl iam contas
- cloudctl iam api-key
- cloudctl iam api-key-create
- cloudctl iam api-key-delete
- cloudctl iam api-key-update
- cloudctl iam api-keys
- cloudctl iam group-import
- cloudctl iam group-remove
- cloudctl iam grupos
- cloudctl iam ldap-create
- cloudctl iam ldap-delete
- cloudctl iam ldap-get
- cloudctl iam ldaps
- cloudctl iam oauth-client
- cloudctl iam oauth-client-delete
- cloudctl iam oauth-client-register
- cloudctl iam oauth-client-update
- cloudctl iam oauth-clients
- cloudctl iam resource-add
- recurso cloudctl iam resource-rm
- cloudctl iam recursos
- cloudctl iam funções

- cloudctl iam saml-disable
- cloudctl iam saml-enable
- cloudctl iam saml-export-metadata
- cloudctl iam saml-status
- cloudctl iam saml-upload-metadata
- cloudctl iam service-api-key
- cloudctl iam service-api-key-create
- cloudctl iam service-api-key-delete
- Cloudctl iam service-api-key-update
- cloudctl iam service-api-keys
- cloudctl iam service-id
- cloudctl iam service-id-create
- cloudctl iam service-id-delete
- cloudctl iam service-id-update
- cloudctl iam service-ids
- cloudctl iam service-policies
- cloudctl iam service-policy
- cloudctl iam service-policy-create
- cloudctl iam service-policy-delete
- cloudctl iam service-policy-update
- cloudctl iam equipe-add-groups
- cloudctl iam team-add-service-ids
- cloudctl iam team-add-users
- cloudctl iam team-create
- cloudctl iam team-delete
- cloudctl iam equipe-get
- cloudctl iam team-remove-groups
- cloudctl iam team-remove-service-ids
- cloudctl iam team-remove-users
- cloudctl iam equipes
- cloudctl iam user-import
- cloudctl iam user-remove
- usuários do cloudctl iam

cloudctl iam contas

Liste todas as contas.

Exemplo

```
cloudctl iam contas
```

cloudctl iam api-key

Liste os detalhes de uma chave de API.

Exemplo

```
cloudctl iam api-key NAME  [--uuid]

OPTIONS:
   --uuid  Display only uuid
```

cloudctl iam api-key-create

Criar uma chave API.

Exemplo

```
cloudctl iam api-key-create NAME  [-d, --description DESCRIPTION] [-f, --file FILE]

OPTIONS:
   -d value, --description value  Description of the API key
   -f value, --file value         Save API key information to specified file, if not set, the JSON
content will be displayed
```

cloudctl iam api-key-delete

Exclua uma chave de API.

Exemplo

```
cloudctl iam api-key-delete NAME [-f, --force]

OPTIONS:
   -f, --force  Delete without confirmation
```

cloudctl iam api-key-update

Atualize uma chave de API.

Exemplo

```
cloudctl iam api-key-update NAME [-n, --name NEW_NAME] [-d, --description DESCRIPTION] [-f, --force]

OPTIONS:
   -d value, --description value  New description of the API key
   -f, --force                    Update without confirmation
   -n value, --name value         New name of the API key
```

cloudctl iam api-keys

Liste todas as chaves de API.

Exemplo

```
cloudctl iam api-keys

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

cloudctl iam group-import

Importe um grupo de uma conexão LDAP.

Exemplo

```
cloudctl iam group-import -g searchFilter

OPTIONS:
   -c value, --connection value  The ID of the LDAP connection
   -f, --force                   Import without confirmation
   -g value, --group value       A LDAP search filter for the groups to import
```

cloudctl iam group-remove

Remova um ou mais grupos.

Exemplo

```
cloudctl iam group-remove groupID1,groupID2,...

OPTIONS:
   -f, --force  Remove without confirmation
```

cloudctl iam grupos

Liste todos os grupos importados.

Exemplo

```
cloudctl iam grupos

OPTIONS:
```

```
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

cloudctl iam ldap-create

Crie uma nova conexão LDAP.

Exemplo

```
cloudctl iam ldap-create NAME --basedn BASEDN --server SERVER --group-filter GROUP-FILTER --group-
id-map GROUP-ID-MAP --group-member-id-map GROUP-MEMBER-ID-MAP --user-filter USER-FILTER --user-id-
map USER-ID-MAP [--binddn BINDDN] [--binddn-password BINDDN-PASSWORD] [-t TYPE]

OPTIONS:
   --basedn value               The distinguished name of the search base
   --binddn value               The user who is allowed to search the base DN, if not given, the
LDAP connection is established without authentication
   --binddn-password value      The password of the user who is mentioned in the binddn
   --group-filter value         The filter clause for searching groups
   --group-id-map value         The filter to map a group name to an LDAP entry
   --group-member-id-map value  The filter to map a user to a group
   --server value               The LDAP directory URL
   -t value, --type value       Type of the LDAP server being used, default value is Custom
   --user-filter value          The filter clause for searching users
   --user-id-map value          The filter to map a user name to an LDAP entry
```

cloudctl iam ldap-delete

Exclua uma conexão LDAP.

Exemplo

```
cloudctl iam ldap-delete

OPTIONS:
   -c value, --connection value  The ID of the LDAP connection
   -f, --force                   Delete without confirmation
```

cloudctl iam ldap-get

Obter detalhes da conexão LDAP.

Exemplo

```
cloudctl iam ldap-get

OPTIONS:
   -c value, --connection value  The ID of the LDAP connection
```

cloudctl iam ldaps

Liste todas as conexões LDAP.

Exemplo

```
cloudctl iam ldaps

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

cloudctl iam oauth-client

Exibir detalhes do um registro de um cliente no formato JSON.

Exemplo

```
cloudctl iam oauth-client CLIENT_ID
```

cloudctl iam oauth-client-delete

Exclua um ou mais registros de cliente

Exemplo

```
cloudctl iam oauth-client-delete [-f] <CLIENT_ID> [CLIENT_ID-2..]

OPTIONS:
   -f  Force the removal of a registered client with no user prompts
```

cloudctl iam oauth-client-register

Registre um cliente com um serviço de autorização.

Exemplo

```
cloudctl iam oauth-client-register --file REGISTRATION_JSON_FILE

OPTIONS:
   -f value, --file value  Path to a file containing the client registration JSON data
```

cloudctl iam oauth-client-update

Atualizar um registro do cliente.

Exemplo

```
cloudctl iam oauth-client-update CLIENT_ID --file REGISTRATION_JSON_FILE

OPTIONS:
   -f value, --file value  Path to a file containing the client registration JSON data
```

cloudctl iam oauth-clients

Liste todos os clientes registrados.

Exemplo

```
cloudctl iam oauth-clients

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

cloudctl iam resource-add

Inclua um recurso em uma equipe.

Exemplo

```
cloudctl iam resource-add <TEAM_ID> -r <RESOURCE_CRN>

OPTIONS:
   -r value, --resources value  Cloud Resource Name of resource to add, can be a comma separated
list
```

recurso cloudctl iam resource-rm

Remover um recurso de uma equipe.

Exemplo

```
cloudctl iam resource-rm <TEAM_ID> -r <RESOURCE_CRN>

OPTIONS:
   -r value, --resources value  Cloud Resource Name of resource to remove; can be a comma-separated
list
```

recursos cloudctl iam

Liste os recursos para as equipes às quais você está designado.

Exemplo

```
cloudctl iam resources [-t, --team TEAM_ID | -r, --resource-type RESOURCE_TYPE]

OPTIONS:
   --json                           Display output in JSON format
   -r value, --resource-type value  Only return resources of this type. A opção 'resource-type' e a
opção 'team' não podem ser especificadas juntas.
   -s                               Do not show the column headers in the output
   -t value, --team value           Only return resources assigned to this team. Opção 'team' e a
opção 'resource-type' não podem ser especificadas juntas.
```

funções cloudctl iam

Liste as funções.

Exemplo

```
funções cloudctl iam

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

cloudctl iam saml-disable

Desative a autenticação SAML.

Exemplo

```
cloudctl iam saml-disable
```

cloudctl iam saml-enable

Ative a autenticação SAML.

Exemplo

```
cloudctl iam saml-enable
```

cloudctl iam saml-export-metadata

Exporte o conteúdo de metadados SAML para criar uma integração SAML. Requer que SAML seja ativado com 'cloudctl iam saml-enable'.

Exemplo

```
cloudctl iam saml-export-metadata [ -- file SAML_XML_FILE ]

OPTIONS:
   --file value, -f value  Write the SAML metadata content to file
```

cloudctl iam saml-status

Obtenha o status de configuração SAML.

Exemplo

```
cloudctl iam saml-status
```

cloudctl iam saml-upload-metadados

Faça upload do conteúdo de metadados SAML para concluir a integração do SAML.

Exemplo

```
cloudctl iam saml-upload-metadata -- file SAML_XML_FILE

OPTIONS:
   --file value, -f value  Read the SAML metadata content from file
```

Cloudctl iam service-api-key

Liste os detalhes de uma chave de API de serviço.

Exemplo

```
cloudctl iam service-api-key NAME SERVICE_ID_NAME [--uuid]

OPTIONS:
   --uuid  Display only uuid
```

cloudctl iam service-api-key-create

Crie uma chave de API de serviço.

Exemplo

```
cloudctl iam service-api-key-create NAME SERVICE_ID_NAME [-d, --description DESCRIPTION] [-f, --file
FILE]

OPTIONS:
   -d value, --description value  Description of the API key
   -f value, --file value         Save API key information to specified file, if not set, the JSON
content will be displayed
```

Cloudctl iam service-api-key-delete

Exclua uma chave de API de serviço.

Exemplo

```
cloudctl iam service-api-key-delete NAME SERVICE_ID_NAME [-f, --force]

OPTIONS:
   -f, --force  Delete without confirmation
```

Cloudctl iam service-api-key-update

Atualize uma chave de API de serviço.

Exemplo

```
cloudctl iam service-api-key-update NAME SERVICE_ID_NAME  [-n, --name NEW_NAME] [-d, --description
DESCRIPTION] [-f, --force]

OPTIONS:
   -d value, --description value  New description of the service API key
   -f, --force                    Update without confirmation
   -n value, --name value         New name of the service API key
```

cloudctl iam service-api-keys

Liste todas as chaves de API de um serviço.

Exemplo

```
cloudctl iam service-api-keys SERVICE_ID_NAME

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

Cloudctl iam service-id

Exibir detalhes de um ID de serviço.

Exemplo

```
cloudctl iam service-id NAME [--uuid]

OPTIONS:
   --uuid  Display the UUID of the service ID
```

Cloudctl iam service-id-create

Crie um ID de serviço.

Exemplo

```
cloudctl iam service-id-create NAME [-d, --description DESCRIPTION]

OPTIONS:
   -d value, --description value  Description of the service ID
```

Cloudctl iam service-id-delete

Exclua um ID de serviço.

Exemplo

```
cloudctl iam service-id-delete NAME [-f, --force]

OPTIONS:
   -f, --force  Delete without confirmation
```

Cloudctl iam service-id-update

Atualize um ID de serviço.

Exemplo

```
cloudctl iam service-id-update NAME [-n, --name NEW_NAME] [-d, --description DESCRIPTION] [-f, --
force]

OPTIONS:
   -d value, --description value  New description of the service ID
   -f, --force                    Update without confirmation
   -n value, --name value         New name of the service ID
```

cloudctl iam service-ids

Listar todos os IDs de serviço.

Exemplo

```
cloudctl iam service-ids -- uuid

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
   --uuid  Show UUID of service IDs only
```

Cloudctl iam service-policies

Liste todas as políticas de serviço do serviço especificado.

Exemplo

```
cloudctl iam service-policies SERVICE_ID_NAME [ -- json ]

OPTIONS:
   --json  Display policy in JSON format
```

Cloudctl iam service-policy

Exiba detalhes de uma política de serviço.

Exemplo

```
cloudctl iam service-policy SERVICE_ID_NAME POLICY_ID [ -- json ]

OPTIONS:
   --json  Display policy in JSON format
```

Cloudctl iam service-policy-create

Crie uma política de serviço.

Exemplo

```
cloudctl iam service-policy-create SERVICE_ID_NAME {-r, --roles ROLE_NAME1,ROLE_NAME2... [--service-
name SERVICE_NAME]} [-f, --force]

OPTIONS:
   -f, --force              Create service policy without confirmation
   -r value, --roles value  Role names of the policy definition; for supported roles, run 'cloudctl
iam roles'
   --service-name value     Service name of the policy definition
```

cloudctl iam service-policy-delete

Exclua uma política de serviço.

Exemplo

```
cloudctl iam service-policy-delete SERVICE_ID_NAME POLICY_ID [-f, --force]

OPTIONS:
   -f, --force  Delete without confirmation
```

Cloudctl iam service-policy-update

Atualize uma política de serviço.

Exemplo

```
cloudctl iam service-policy-update SERVICE_ID_NAME POLICY_ID --roles ROLE_NAME1,ROLE_NAME2... --
service-name SERVICE_NAME [ -f, -- force ]

OPTIONS:
   -f, --force              Update service policy without confirmation
   -r value, --roles value  Role names of the policy definition; for supported roles, run 'cloudctl
iam roles'
   --service-name value     Service name of the policy definition
```

cloudctl iam team-add-groups

Inclua grupos em uma equipe com a função definida.

Exemplo

```
cloudctl iam team-add-groups TEAM_ID ROLE -g group1ID,group2ID,...]

OPTIONS:
   -g value, --groups value  Groups to add to the team
```

cloudctl iam team-add-service-ids

Inclua IDs de serviço em uma equipe.

Exemplo

```
cloudctl iam team-add-service-ids TEAMID -s Service-ID-Name1,Service-ID-Name2,...

OPTIONS:
   -s value, --service-id-names value  Names of service IDs to add to the team
```

cloudctl iam team-add-users

Inclua usuários em uma equipe com a função definida.

Exemplo

```
cloudctl iam team-add-users TEAM_ID ROLE -u user1ID,user2ID,...

OPTIONS:
   -u value, --users value  Users to add to the team
```

cloudctl iam team-create

Criar uma equipe.

Exemplo

```
cloudctl iam team-create NAME
```

cloudctl iam team-delete

Exclua uma equipe.

Exemplo

```
cloudctl iam equipe-delete TEAM_ID [ -f, -- force ]

OPTIONS:
   -f, --force  Delete without confirmation
```

cloudctl iam team-get

Visualize usuários e grupos para uma equipe.

Exemplo

```
cloudctl iam team-get TEAM_ID

OPTIONS:
   --TEAM_ID value  ID of team
   --json           Display output in JSON format
   -s               Do not show the column headers in the output
```

cloudctl iam team-remove-groups

Remova grupos de uma equipe.

Exemplo

```
cloudctl iam team-remove-groups TEAM_ID -g group1ID,group2ID,...

OPTIONS:
   -f, --force               Remove without confirmation
   -g value, --groups value  Groups to remove from the team
```

cloudctl iam team-remove-service-ids

Remova IDs de serviço de uma equipe.

Exemplo

```
cloudctl iam team-remove-service-ids TEAMID -s Service-ID-Name1, Service-ID-Name2,...

OPTIONS:
```

```
   -f, --force                          Remove without confirmation
   -s value, --service-id-names value   Names of service IDs to be removed from the team
```

cloudctl iam team-remove-users

Remova usuários de uma equipe.

Exemplo

```
cloudctl iam team-remove-users TEAM_ID -u user1ID,user2ID,...

OPTIONS:
   -f, --force              Remove without confirmation
   -u value, --users value  Users to remove from the team
```

cloudctl iam equipes

Liste todas as equipes.

Exemplo

```
cloudctl iam equipes

OPTIONS:
   --json                 Display output in JSON format
   -s                     Do not show the column headers in the output
   -u value, --user value  Return only the teams that contain this user
```

cloudctl iam user-import

Importe um usuário de uma conexão LDAP.

Exemplo

```
cloudctl iam user-import -u searchFilter

OPTIONS:
   -c value, --connection value  The ID of the LDAP connection
   -f, --force                   Import without confirmation
   -u value, --user value        A LDAP search filter for the users to import
```

cloudctl iam user-remove

Remova um ou mais usuários.

Exemplo

```
cloudctl iam user-remove user1ID,user2ID,...

OPTIONS:
   -f, --force  Remove without confirmation
```

usuários do cloudctl iam

Liste todos os usuários importados.

Exemplo

```
usuários do cloudctl iam

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

Comandos multicluster (mc) da CLI do

IBM Cloud Private

Aprenda sobre os comandos `cloudctl mc` que podem ser executados para acessar seu cluster do IBM Multicloud Manager.

cloudctl mc

cloudctl mc apply

Aplique uma configuração a um recurso por `filename` ou `stdin`.

Exemplo

```
cloudctl mc apply -f FILENAME [options]

OPTIONS:
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --filename, -f       Filename, directory, or URL to files create -f filename
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --recursive, -R      Process the directory used in -f, --filename recursively; useful when you
want to manage related manifests organized within the same directory
   --stderrthreshold    Logs at or above this threshold go to stderr
   -v, -v               Log level for V logs
   --vmodule            Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc cluster import

Importar um cluster

Exemplo

```
cloudctl mc cluster import -f {config.yaml} [-C|--cluster-context {context}] [-K|--cluster-
kubeconfig {path}] [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]

OPTIONS:
   --alsologtostderr          Log to standard error, as well as files
   --bootstrap-namespace, -b  The namespace that the bootstrap will run in to import the cluster
   --cluster-context, -C      The name of the Kubernetes context of the cluster to import within the
default configuration path, or within the path set with --cluster-kubeconfig, run 'kubectl config
get-contexts' to list contexts
   --cluster-kubeconfig, -K   The path to the alternate Kubernetes config file containing the
cluster to import configuration from. Use --cluster-context if the cluster is not the current
context in the configuration
   --dry-run                  If true, only print the YAML that would be used, but do not apply it
   --filename, -f             Filename, directory, or URL to files import -f config.yaml file
   --kube-host                Override the Kubernetes host and port that the cluster to import will
connect to
   --kubeconfig               Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at         When logging hits line file:N, emit a stack trace
```

```
   --log-dir                 If non-empty, write log files in this directory
   --logtostderr             Log to standard error instead of files
   --recursive, -R           Process the directory used in -f, --filename recursively; useful when
you want to manage related manifests organized within the same directory
   --stderrthreshold         Logs at or above this threshold go to stderr
   --timeout, -t             The length of time to wait for the cluster to join the hub-cluster
   -v, -v                    Log level for V logs
   --vmodule                 Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc cluster list

Listar os clusters importados

Exemplo

```
cloudctl mc cluster list

OPTIONS:
   --alsologtostderr   Log to standard error, as well as files
   --kubeconfig        Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at  When logging hits line file:N, emit a stack trace
   --log-dir           If non-empty, write log files in this directory
   --logtostderr       Log to standard error instead of files
   --stderrthreshold   Logs at or above this threshold go to stderr
   -v, -v              Log level for V logs
   --vmodule           Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc cluster remove

Remover um cluster importado

Exemplo

```
cloudctl mc cluster remove {name} [-n|--namespace {namespace}] [-C|--cluster-context {context}] [-
K|--cluster-kubeconfig {path}] [-b|--bootstrap-namespace {namespace}]

OPTIONS:
   --alsologtostderr         Log to standard error, as well as files
   --boostrap-namespace, -b  The namespace that the bootstrap will run in to import the cluster
   --cluster-context, -C     The name of the Kubernetes context of the cluster to remove within the
default configuration path or within the path set with --cluster-kubeconfig; run 'kubectl config
get-contexts' to list contexts
   --cluster-kubeconfig, -K  The path to the alternate kubeconfig file containing the cluster to be
removed, use --cluster-context if the cluster is not the current context in the configuration
   --dry-run                 If true, only print the YAML that would be used, but do not apply it
   --kubeconfig              Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at        When logging hits line file:N, emit a stack trace
   --log-dir                 If non-empty, write log files in this directory
   --logtostderr             Log to standard error instead of files
   --namespace, -n           The namespace in the hub-cluster to manage the imported cluster. Um
namespace correspondente ao nome do cluster será usado se não estiver configurado.
   --stderrthreshold         Logs at or above this threshold go to stderr
   -v, -v                    Log level for V logs
   --vmodule                 Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc cluster template

Saída de um arquivo config.yaml de modelo usado para importação do cluster

Exemplo

```
cloudctl mc cluster template {name} [-n|--namespace {namespace}]

OPTIONS:
   --alsologtostderr   Log to standard error, as well as files
   --kubeconfig        Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at  When logging hits line file:N, emit a stack trace
   --log-dir           If non-empty, write log files in this directory
   --logtostderr       Log to standard error instead of files
   --namespace, -n     The namespace in the hub-cluster for the target cluster
   --stderrthreshold   Logs at or above this threshold go to stderr
```

```
   -v, -v              Log level for V logs
   --vmodule           Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc create

Criar um recurso usando um arquivo ou uma stdin

Exemplo

```
cloudctl mc create -f FILENAME [options]

OPTIONS:
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --filename, -f       Filename, directory, or URL to files create -f filename
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --recursive, -R      Process the directory used in -f, --filename recursively; useful when you
want to manage related manifests organized within the same directory
   --stderrthreshold    Logs at or above this threshold go to stderr
   -v, -v               Log level for V logs
   --vmodule            Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc create helmrepo

Criar um repositório Helm

Exemplo

```
cloudctl mc create helmrepo --repo-name <repo_name> --repo-url <repo_url>

OPTIONS:
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --repo-name          Name of Repository
   --repo-url, -u       URL of Repository
   --stderrthreshold    Logs at or above this threshold go to stderr
   -v, -v               Log level for V logs
   --vmodule            Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc delete

Excluir recursos por nomes de arquivos, stdin, recursos e nomes ou por recursos e seletor de rótulo

Exemplo

```
cloudctl mc delete ([-f FILENAME] | TYPE [(NAME | -l label | --all)]) [options]

OPTIONS:
   --all                Delete all resources, including uninitialized ones, in the namespace of the
specified resource types.
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --field-selector     Selector (field query) to filter on, supports '=', '==', and '!='.(e.g. --
field-selector key1=value1,key2=value2); the server only supports a limited number of field queries
```

```
per type
   --filename, -f       Filename, directory, or URL to files delete -f filename
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --output, -o         Output mode; use "-o name" for shorter output (resource/name)
   --recursive, -R      Process the directory used in -f, --filename recursively; useful when you
want to manage related manifests organized within the same directory
   --selector, -l       Selector (label query) to filter on, not including uninitialized ones.
   --stderrthreshold    Logs at or above this threshold go to stderr
   -v, -v               Log level for V logs
   --vmodule            Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc deploy application

Implementar um aplicativo

Exemplo

```
cloudctl mc deploy application <app_name> --cluster-replica <number_of_clusters> --cluster-selector
<key=value>

OPTIONS:
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-replica    Number of clusters to deploy the application to
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --resource-selector  Resource selector
   --stderrthreshold    Logs at or above this threshold go to stderr
   -v, -v               Log level for V logs
   --vmodule            Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc describe

Mostrar detalhes de um recurso ou grupo de recursos específico

Exemplo

```
cloudctl mc describe (-f FILENAME | TYPE [NAME_PREFIX | -l label] | TYPE/NAME) [options]

OPTIONS:
   --all-namespaces         If present, list the requested object(s) across all namespaces;
namespace in current context is ignored even if specified with --namespace
   --alsologtostderr        Log to standard error, as well as files
   --cluster, -c            Name of the cluster
   --cluster-namespace      Namespace of the cluster
   --cluster-selector       Selector (label query) that is used to filter clusters; supports '=',
'==', and '!=' (-l key1=value1,key2=value2)
   --filename, -f           Filename, directory, or URL to files containing the resource to describe
   --include-uninitialized  If true, the kubectl command applies to uninitialized objects; if
explicitly set to false, this flag overrides other flags that make the kubectl commands apply to
uninitialized objects, such as, "--all"; objects with empty metadata.initializers are regarded as
initialized
   --kubeconfig             Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at       When logging hits line file:N, emit a stack trace
   --log-dir                If non-empty, write log files in this directory
   --logtostderr            Log to standard error instead of files
   --namespace, -n          Namespace of the object
   --recursive, -R          Process the directory used in -f, --filename recursively; useful when
you want to manage related manifests organized within the same directory
   --selector, -l           Selector (label query) to filter on, supports '=', '==', and '!='.(por
exemplo, -l key1=value1,key2=value2)
   --show-events            If true, display events related to the described object.
   --stderrthreshold        Logs at or above this threshold go to stderr
```

```
   -v, -v                   Log level for V logs
   --vmodule                Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc edit

Editar um recurso usando o editor padrão

Exemplo

```
cloudctl mc edit (<resource_type/resource_name> | -f <filename>)

OPTIONS:
   --alsologtostderr       Log to standard error, as well as files
   --cluster, -c           Name of the cluster
   --cluster-namespace     Namespace of the cluster
   --cluster-selector      Selector (label query) that is used to filter clusters; supports '=',
'==', and '!=' (-l key1=value1,key2=value2)
   --kubeconfig            Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at      When logging hits line file:N, emit a stack trace
   --log-dir               If non-empty, write log files in this directory
   --logtostderr           Log to standard error instead of files
   --namespace, -n         Namespace of the object
   --output, -o            Output format; one of: json|yaml|wide|custom-columns=...|custom-columns-
file=...|go-template=...|go-template-file=...|jsonpath=...|jsonpath-file=... See custom columns
[http://kubernetes.io/docs/user-guide/kubectl-overview/#custom-columns], golang template
[http://golang.org/pkg/text/template/#pkg-overview] and jsonpath template
[http://kubernetes.io/docs/user-guide/jsonpath]
   --output-patch          Output the patch if the resource is edited
   --stderrthreshold       Logs at or above this threshold go to stderr
   -v, -v                  Log level for V logs
   --vmodule               Comma-separated list of pattern=N settings for file-filtered logging
   --windows-line-endings  Defaults to the line ending native to your platform.
```

cloudctl mc get

Exibir um ou muitos recursos

Exemplo

```
cloudctl mc get [(-o|--output=)json|yaml|wide|go-template=...|go-template-
file=...|jsonpath=...|jsonpath-file=...] (TYPE[.VERSION][.GROUP] [NAME | -l label] | TYPE[.VERSION]
[.GROUP]/NAME ...) [flags]

OPTIONS:
   --all-namespaces              If present, list the requested object(s) across all namespaces;
namespace in current context is ignored even if specified with --namespace
   --allow-missing-template-keys  If true, ignore any errors in templates when a field or map key is
missing in the template; only applies to golang and jsonpath output formats
   --alsologtostderr             Log to standard error, as well as files
   --cluster, -c                 Name of the cluster
   --cluster-namespace           Namespace of the cluster
   --cluster-selector            Selector (label query) that is used to filter clusters; supports
'=', '==', and '!=' (-l key1=value1,key2=value2)
   --field-selector              Selector (field query) to filter on, supports '=', '==', and '!='.
(por exemplo, -l key1=value1,key2=value2)
   --kubeconfig                  Path to a kubeconfig file, which overrides $KUBECONFIG
   --label-columns, -L           Accepts a comma separated list of labels that are going to be
presented as columns; names are case-sensitive; you can also use multiple flag options such as -L
label1 -L label2...
   --label-selector, -l          Selector (label query) to filter on, supports '=', '==', and '!='.
(por exemplo, -l key1=value1,key2=value2)
   --log-backtrace-at            When logging hits line file:N, emit a stack trace
   --log-dir                     If non-empty, write log files in this directory
   --logtostderr                 Log to standard error instead of files
   --namespace, -n               Namespace of the object
   --no-headers                  When using the default or custom-column output format, do not
print headers (default: print headers)
   --output, -o                  Output format. Um de: json|yaml|wide|name|custom-
columns=...|custom-columns-file=...|go-template=...|go-template-file=...|jsonpath=...|jsonpath-
file=... Consulte as colunas customizadas [http://kubernetes.io/docs/user-guide/kubectl-
overview/#custom-columns], modelo golang [http://golang.org/pkg/text/template/#pkg-overview] e
modelo jsonpath [http://kubernetes.io/docs/user-guide/jsonpath].
   --server-print                Enable server print
```

```
   --show-labels                  When printing, show all labels as the last column (default hide
labels column)
   --sort-by                      If non-empty, sort list types using this field specification; the
field specification is expressed as a JSONPath expression (e.g. '{.metadata.name}'); the field in
the API resource specified by this JSONPath expression must be an integer or a string
   --stderrthreshold              Logs at or above this threshold go to stderr
   --template                     Template string or path to template file to use when -o=go-
template, -o=go-template-file. O formato de modelo é modelos de goldang
[http://golang.org/pkg/text/template/#pkg-overview].
   -v, -v                         Log level for V logs
   --vmodule                      Comma-separated list of pattern=N settings for file-filtered
logging
   --watch, -w                    After listing/getting the requested object, watch for changes;
uninitialized objects are excluded if no object name is provided
```

cloudctl mc label

Atualizar os rótulos em um recurso

Exemplo

```
cloudctl mc label [--overwrite] (-f FILENAME | TYPE NAME) KEY_1=VAL_1 ... KEY_N=VAL_N [--resource-
version=version] [options]

OPTIONS:
   --all                Select all resources, including uninitialized ones, in the namespace of the
specified resource types
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --field-selector     Selector (field query) to filter on, supports '=', '==', and '!='.(e.g. --
field-selector key1=value1,key2=value2); the server only supports a limited number of field queries
per type
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --selector, -l       Selector (label query) to filter on, not including uninitialized ones,
supports '=', '==', and '!='.(por exemplo, -l key1=value1,key2=value2).
   --stderrthreshold    Logs at or above this threshold go to stderr
   -v, -v               Log level for V logs
   --vmodule            Comma-separated list of pattern=N settings for file-filtered logging
```

cloudctl mc logs

Imprimir os logs para um contêiner em um pod

Exemplo

```
cloudctl mc logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER] [options]

OPTIONS:
   --alsologtostderr    Log to standard error, as well as files
   --cluster, -c        Name of the cluster
   --cluster-namespace  Namespace of the cluster
   --cluster-selector   Selector (label query) that is used to filter clusters; supports '=', '==',
and '!=' (-l key1=value1,key2=value2)
   --container          Print the logs of this container
   --follow, -f         Specify if the logs should be streamed
   --kubeconfig         Path to a kubeconfig file, which overrides $KUBECONFIG
   --log-backtrace-at   When logging hits line file:N, emit a stack trace
   --log-dir            If non-empty, write log files in this directory
   --logtostderr        Log to standard error instead of files
   --namespace, -n      Namespace of the object
   --previous, -p       If true, print the logs for the previous instance of the container in a pod
if it exists
   --since              Only return logs newer than a relative duration, such as 5s, 2m, or 3h.;
defaults to all logs
   --stderrthreshold    Logs at or above this threshold go to stderr
   --tail               Lines of recent log file to display, defaults to -1 (show all lines)
```

```
--timestamps          Include timestamps on each line in the log output
-v, -v                Log level for V logs
--vmodule             Comma-separated list of pattern=N settings for file-filtered logging
```

IBM Cloud Private Comandos de medição de CLI

Aprenda sobre os comandos `cloudctl metering` que podem ser executados para obter relatórios de medição.

medição do cloudctl

- cloudctl metering full-report
- cloudctl metering mcm-report
- relatório de carga de trabalho cloudctl-relatório

cloudctl metering full-report

Fazer download do relatório completo

Exemplo

```
cloudctl metering full-report
```

mcm-mcm-relatório cloudctl

Faça download do relatório de vários clusters

Exemplo

```
mcm-mcm-relatório cloudctl
```

relatório de carga de trabalho cloudctl-relatório

Fazer download do relatório de carga de

Exemplo

```
relatório de carga de trabalho cloudctl-relatório
```

Comandos pm da CLI do IBM Cloud Private (pm)

Saiba mais sobre os comandos `cloudctl pm` que podem ser executados para gerenciar senhas para os serviços em um namespace de cluster.

cloudctl pm

- verificação do cloudctl pm
- cloudctl pm password-rule-rm
- cloudctl pm password-rule-set
- cloudctl pm password-rules
- cloudctl pm update-secret

verificação de cloudctl pm

Verifique um valor de senha usando regras ou as regras atuais de um namespace.

Exemplo

```
cloudctl pm check --rule <rule_regex> [--rule <rule_regex>] [--password <password>]

OPTIONS:
   --password value, -p value  Password value to check. Se não configurado, a senha será lida da
entrada padrão.
```

```
   --rule value, -r value      Regular expression for password validation. Uma ou mais regras são
necessárias.
```

cloudctl pm password-rule-rm

Remova uma regra de senha para um namespace.

Exemplo

```
cloudctl pm password-rule-rm <namespace> <rule_name>
```

cloudctl pm password-rule-set

Configure uma regra de senha para um namespace.

Exemplo

```
cloudctl pm password-rule-set <namespace> <rule_name> <rule_regex> <rule_desc>
```

cloudctl pm password-rules

Liste as regras de senha para um namespace.

Exemplo

```
cloudctl pm password-rules <namespace>

OPTIONS:
   --json  Display output in JSON format
   -s      Do not show the column headers in the output
```

cloudctl pm update-secret

Atualizar um segredo e reiniciar as implementações que usam o segredo.

Exemplo

```
cloudctl pm update-secret <namespace> <secret_name> [-f] [-d <data_key>=<data_value>]

OPTIONS:
   -d value  The secret data key to update. O valor da chave de dados de segredo será solicitado se
não configurado.
   -f        Update the secret data with no user prompts
```

Instalando a CLI do Kubernetes (kubectl)

Para acessar seu cluster usando a interface da linha de comandos (CLI), deve-se instalar e configurar o `kubectl`, a ferramenta de linha de comandos do Kubernetes.

1. Sincronize os clocks entre o computador cliente e os nós no cluster do IBM® Cloud Private. Para sincronizar os seus clocks, é possível usar o protocolo de tempo de rede (NTP). Para obter mais informações sobre como configurar o NTP, consulte a documentação do usuário para o seu sistema operacional.

2. Na página *Introdução* da console de gerenciamento do IBM Cloud Private, clique em **Instalar ferramentas da CLI**.

3. Expanda **Instalar a CLI do Kubernetes** para fazer download do instalador usando um comando `curl`. Copie e execute o comando curl para seu sistema operacional e, em seguida, continue o procedimento de instalação na documentação do produto:

   Escolha o comando curl para o sistema operacional aplicável. Por exemplo, é possível executar o comando a seguir para macOS:

   ```
   curl -kLo <install_file> https://<Cluster Master Host>:<Cluster Master API
   Port>/api/cli/kubectl-darwin-amd64
   ```

4. Mude o arquivo para um executável e, em seguida, mova o arquivo para seu diretório. Consulte os comandos a seguir, em que `<path_to_installer>` é o local do arquivo transferido por download e `<install_file>` é o nome do arquivo:

- Para Linux® e macOS, execute os seguintes comandos para mudar e mover o arquivo:

  ```
  chmod 755 <path_to_installer>/<install_file>

  sudo mv < path_to_installer > / < install_file> /usr/local/bin/kubectl
  ```

- Para o Windows™, renomeie o arquivo transferido por download para `kubectl` e coloque o arquivo na variável de ambiente PATH.

Nota: também é possível fazer download do Kubernetes. Consulte Instalar e configurar o kubectl.

5. Obtenha os detalhes da configuração de cluster. É possível obter os detalhes da configuração de cluster usando a CLI ou a console de gerenciamento do IBM Cloud Private.

   - Para obter os detalhes de configuração da console de gerenciamento:
     1. Efetue login em seu console de gerenciamento do cluster. Consulte Acessando o seu cluster do IBM Cloud Private usando o console de gerenciamento.
     2. Selecione o ícone do usuário e, em seguida, clique em **Configurar cliente**. Os

detalhes da configuração de cluster são exibidos e se assemelham ao código a seguir, em que o `<Cluster Master Host>` está definido no Terminal principal:

```
    ```
    kubectl config set-cluster {cluster_name} --server=https://<Cluster Master Host>:8001 --
insecure-skip-tls-verify=true
    kubectl config set-context {cluster_name}-context --cluster={cluster_name}
    kubectl config set-credentials {username} --token={token}
    kubectl config set-context {cluster_name}-context --user={username} --namespace=default
    kubectl config use-context {cluster_name}-context

    ```

    **Nota:** essa configuração expira em 12 horas. Para continuar usando a CLI, deve-se efetuar
login e reconfigurar o `kubectl` a cada 12 horas. Para evitar essa limitação, é possível configurar
sua CLI usando contas do serviço. Consulte
```

Configurando a CLI do Kubernetes usando tokens de conta de serviço.

```
3. Copie e cole as informações de configuração para sua linha de comandos e pressione **Enter**.
```

- Para obter os detalhes de configuração da CLI do IBM Cloud Private:
  1. Instale a interface da linha de comandos (CLI) do IBM Cloud Private e efetue login em seu cluster. Consulte Instalando a CLI do IBM Cloud Private.

Instalando a CLI do Helm (helm)

É possível usar a interface da linha de comandos (CLI) do Helm para gerenciar liberações no cluster.

Para obter mais informações sobre o Helm, consulte Docs do Helm no GitHub.

Como o IBM Cloud Private oferece controle de acesso baseado na função, deve-se instalar uma versão específica do cliente CLI do Helm e fornecer certificados que contenham o token de acesso do IBM Cloud Private para uma conta específica.

Importante: depois de configurar uma conexão, deve-se incluir a opção `--tls` para comandos Helm que acessam o servidor por meio do Tiller.

Antes de configurar a CLI do Helm, deve-se concluir as etapas a seguir:

- Instale a ferramenta de linha de comandos Kubernetes, `kubectl`, e configure o acesso ao seu cluster. Consulte Instalando a CLI do Kubernetes (kubectl) .
- Instale a CLI do IBM Cloud Private e efetue login em seu cluster. Consulte Instalando a CLI do IBM Cloud Private.
- Obtenha acesso ao nó de inicialização e à conta do administrador de cluster, ou solicite que alguém que tem esse nível de acesso crie seu certificado. Se não for possível acessar a conta do administrador de cluster, será necessário ter uma conta do IBM Cloud Private que esteja designada à função de administrador para uma equipe e que possa acessar o namespace `kube-system`.

Instalando a CLI do Helm

É possível instalar a CLI do Helm a partir da CLI do IBM Cloud Private.

Conclua as etapas a seguir para instalar a CLI do Helm usando a CLI do IBM Cloud Private:

1. Na página *Introdução* da console de gerenciamento do IBM Cloud Private, clique em **Instalar ferramentas da CLI**.
2. Expanda **Instalar a CLI do Helm**. Leia o texto e, em seguida, copie e execute o comando curl para seu sistema operacional. Continue o procedimento de instalação na documentação do produto.

   Escolha o comando curl para o sistema operacional aplicável. Por exemplo, é possível executar o comando a seguir para macOS:

   ```
   curl -kLo <install_file> https://<Cluster Master Host>:<Cluster Master API Port>/api/cli/helm-darwin-amd64.tar.gz
   ```

3. Depois de executar o comando curl para seu sistema operacional, crie um diretório `helm-unpacked` e descompacte o arquivo de instalação nesse diretório com os comandos a seguir:

   ```
   mkdir helm-unpacked
   ```

   ```
   tar -xvzf ./<path_to_installer> -C helm-unpacked
   <!-- doc test blocked here-->
   ```

4. Mude o arquivo para um executável, em seguida, mova o arquivo para seu diretório:

   - Para Linux® e macOS, execute os seguintes comandos para mudar e mover o arquivo:

     ```
     chmod 755 ./helm-unpacked/ < unpacked_dir> /helm
     ```

     ```
     sudo mv ./helm-unpacked/<unpacked_dir>/helm /usr/local/bin/helm
     ```

   - Para o Windows™, renomeie o arquivo transferido por download para `helm` e coloque o arquivo na variável de ambiente PATH.

5. Exclua o instalador e archives adicionais descompactados:

   ```
   rm -rf ./helm-unpacked ./<path_to_installer>
   ```

Nota: também é possível fazer download da CLI do Helm por meio da comunidade do Helm. Consulte Helm v2.9.1-Bug Fix Release para o procedimento de instalação.

Verificando a instalação

1. Se você estiver usando o Helm 2.9.1, deverá configurar o HELM_HOME:

   ```
   export HELM_HOME= ~ /.helm
   ```

2. Inicialize a CLI do Helm. **Importante:** não use a sinalização `--upgrade` com o comando `helm init`. A inclusão da sinalização `--upgrade` substitui a versão do servidor do Helm Tiller que está instalada com o IBM Cloud Private.

   - Para ambientes com acesso à Internet, execute o seguinte comando:

     ```
     helm init --client-only
     ```

   - Para ambientes de airgap, execute o seguinte comando:

     ```
     helm init --client-only --skip-refresh
     ```

3. Verifique se a CLI do Helm foi inicializada. Execute o comando a seguir:

   ```
   helm version --tls
   ```

   A saída assemelha-se ao conteúdo a seguir:

   ```
   Cliente: &version.Version{SemVer:"v2.9.1",
   GitCommit:"20adb27c7c5868466912eebdf6664e7390ebe710", GitTreeState:"clean"}
   ```

   ```
   Servidor: &version.Version{SemVer:"v2.9.1+icp",
   GitCommit:"843201eceab24e7102ebb87cb00d82bc973d84a7", GitTreeState:"clean"}
   ```

4. Siga as etapas para revisar uma lista de pacotes disponíveis ou instalados:

1. Incluir um repositório do Helm. Para incluir o repositório Incubador do Kubernetes, execute o comando a seguir:

   ```
   helm repo add incubator https://kubernetes-charts-incubator.storage.googleapis.com/
   ```

2. Visualize os gráficos disponíveis ao executar o comando a seguir:

   ```
   helm search -l
   ```

3. Instale um gráfico. Execute o comando a seguir:

   ```
   helm install -- name = release_name estável / chart_in_repo -- tls
   ```

   Nesse comando, `release_name` é o nome da liberação a ser criada a partir do gráfico e `chart_in_repo` é o nome do gráfico disponível a ser instalado. Por exemplo, para instalar o gráfico WordPress, execute o comando a seguir:

   ```
   helm install --name=my-wordpress stable/wordpress --tls
   ```

4. Liste as liberações executando o comando a seguir:

   ```
   helm list --tls
   ```

   A saída assemelha-se ao conteúdo a seguir:

   ```
   NAME                REVISION    UPDATED                     STATUS      CHART
   NAMESPACE
   my-wordpress        1           Wed Jun 28 22:15:13 2017    DEPLOYED    wordpress-0.6.5
   default
   ```

5. Para remover uma liberação, execute o comando a seguir:

   ```
   helm delete release_name -- purge -- tls
   ```

   Neste comando, o `release_name` é o nome da liberação a ser removida. Por exemplo, para remover a liberação do WordPress, execute o comando a seguir:

   ```
   helm delete my-wordpress --purge --tls
   ```

Instalando a CLI do Istio (istioctl)

Deve-se instalar e configurar a ferramenta de linha de comandos do Istio (istioctl) para usar a interface da linha de comandos (CLI) para gerenciar sua malha de serviço dentro do cluster.

Para obter mais informações sobre o `istioctl`, consulte istioctl.

Antes de configurar a CLI do Istio, as ferramentas de linha de comandos a seguir devem ser instaladas e configuradas para acessar seu cluster:

- A ferramenta de linha de comandos do Kubernetes (kubectl): consulte Instalando a CLI do Kubernetes (kubectl) para obter instruções de instalação.
- CLI do IBM Cloud Private: consulte Instalando a CLI do IBM Cloud Private para obter mais informações.

É possível instalar a CLI do Istio a partir da CLI do IBM Cloud Private.

Conclua as etapas a seguir para instalar a CLI do Istio usando a CLI do IBM Cloud Private:

1. Sincronize os clocks entre o computador cliente e os nós no cluster do IBM Cloud Private. Para sincronizar os seus clocks, é possível usar o protocolo de tempo de rede (NTP). Para obter mais informações sobre como configurar o NTP, consulte a documentação do usuário para o seu sistema operacional.

2. Na página *Introdução* da console de gerenciamento do IBM Cloud Private, clique em **Instalar ferramentas da CLI**.

3. Expanda **Instalar a CLI do Istio**. Leia o texto e, em seguida, copie e execute o comando curl para seu sistema operacional. Continue o procedimento de instalação na documentação do produto.

   Escolha o comando curl para o sistema operacional aplicável. Por exemplo, é possível executar o comando a seguir para macOS:

   ```
   curl -kLo <install_file> https://<Cluster Master Host>:<Cluster Master API
   Port>/api/cli/istioctl-darwin-amd64
   ```

4. Depois de executar o comando curl para seu sistema operacional, continue instalando a CLI do Istio.

   Para instalar a CLI do Istio, execute o comando que corresponde à arquitetura do nó, em que `<path_to_installer>` é o caminho para o diretório onde o arquivo da CLI foi transferido por download e `<install_file>` é o nome do arquivo transferido por download.

   - Para Linux® e macOS, execute os seguintes comandos para mudar e mover o arquivo:

     ```
     chmod 755 <path_to_installer>/<install_file>
     ```

     ```
     sudo mv < path_to_installer > / < install_file> /usr/local/bin/istioctl
     ```

5. Execute o comando a seguir para confirmar que a CLI do Istio está instalada:

   ```
   istioctl -- help
   ```

   O uso da CLI do Istio é exibido.

Para fazer download do arquivo a partir do website do Istio, consulte Istio na página do *Kubernetes* ↗.

Instalando a CLI do Calico (calicoctl)

É possível usar a interface da linha de comandos (CLI) do Calico, calicoctl, para gerenciar redes e políticas de segurança do Calico.

À medida que você instala a CLI do Calico, certifique-se de que ela esteja instalada em seu cluster do IBM® Cloud Private em um nó principal, do trabalhador ou proxy.

Também é possível configurar o `calicoctl` a partir de uma estação de trabalho remota que esteja fora do ambiente do IBM Cloud Private.

Para configurar a linha de comandos Calico, conclua as etapas a seguir:

1. Na página *Introdução* da console de gerenciamento do IBM Cloud Private, clique em **Instalar ferramentas da CLI**.
2. Expanda **Instalar a CLI do Calico**. Leia o texto e, em seguida, faça download do instalador usando o comando `curl`.

   Escolha o comando curl para o sistema operacional aplicável. Por exemplo, é possível executar o comando a seguir para macOS, em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal:

   ```
   curl -kLo <install_file> https://<Cluster Master Host>:<Cluster Master API
   Port>/api/cli/calicoctl-darwin-amd64
   ```

   Lembre-se de que o comando curl para seu cluster está localizado na console de gerenciamento.

3. Depois de executar o comando curl para seu sistema operacional, é possível instalar a CLI do Calico. Para configurar a CLI do Calico, execute os comandos a seguir que correspondem à sua arquitetura de nó, em que `<path_to_installer>` é o caminho para o diretório no qual você transferiu por download o arquivo CLI e `<install_file>` é o nome do arquivo transferido por download.

   - Por exemplo, para Linux® e macOS, execute os comandos a seguir para mudar e mover o arquivo.

     ```
     chmod 755 <path_to_installer>/<install_file>
     ```

     ```
     sudo mv < path_to_installer > / < install_file> /usr/local/bin/calicoctl
     ```

   - Para o Windows™, renomeie o arquivo transferido por download para `calicoctl` e inclua o arquivo em sua variável de ambiente PATH.

4. Confirme se a CLI do Calico está instalada.

   ```
   calicoctl -- help
   ```

5. Se você estiver configurando o `calicoctl` em uma estação de trabalho remota, copie os arquivos a seguir do nó principal para sua estação de trabalho:

   - `/etc/cfc/conf/etcd/ca.pem`
   - `/etc/cfc/conf/etcd/client-key.pem`
   - `/etc/cfc/conf/etcd/client.pem`

6. Configure `calicoctl` para usar o armazenamento de dados etcdv3. Use o mesmo `cluster_name` que está no arquivo `config.yaml` no nó de inicialização.

   - Exporte o arquivo de certificado com o comando a seguir:

     ```
     export ETCD_CERT_FILE=/etc/cfc/conf/etcd/client.pem
     ```

   - Exporte o arquivo de certificado de CA:

     ```
     export ETCD_CA_CERT_FILE=/etc/cfc/conf/etcd/ca.pem
     ```

   - Exporte o arquivo-chave:

     ```
     export ETCD_KEY_FILE=/etc/cfc/conf/etcd/client-key.pem
     ```

   - Exporte o domínio de CA com o comando a seguir, em que `<Cluster Master Host>` está definido no Terminal principal:

     ```
     export ETCD_ENDPOINTS=https://<Cluster Master Host>:4001
     ```

   Nota: para reter os valores da variável de ambiente entre sessões, é possível incluí-los em um script, tal como `.bashrc`. Consulte o exemplo a seguir. Deve-se copiar o script para todos os nós nos quais você deseja executar os comandos da CLI do Calico:

   ```
   #!/bin/sh
      export ETCD_CERT_FILE=/etc/cfc/conf/etcd/client.pem
      export ETCD_CA_CERT_FILE=/etc/cfc/conf/etcd/ca.pem
      export ETCD_KEY_FILE=/etc/cfc/conf/etcd/client-key.pem
      export ETCD_ENDPOINTS=https://<Cluster Master Host>:4001
   ```

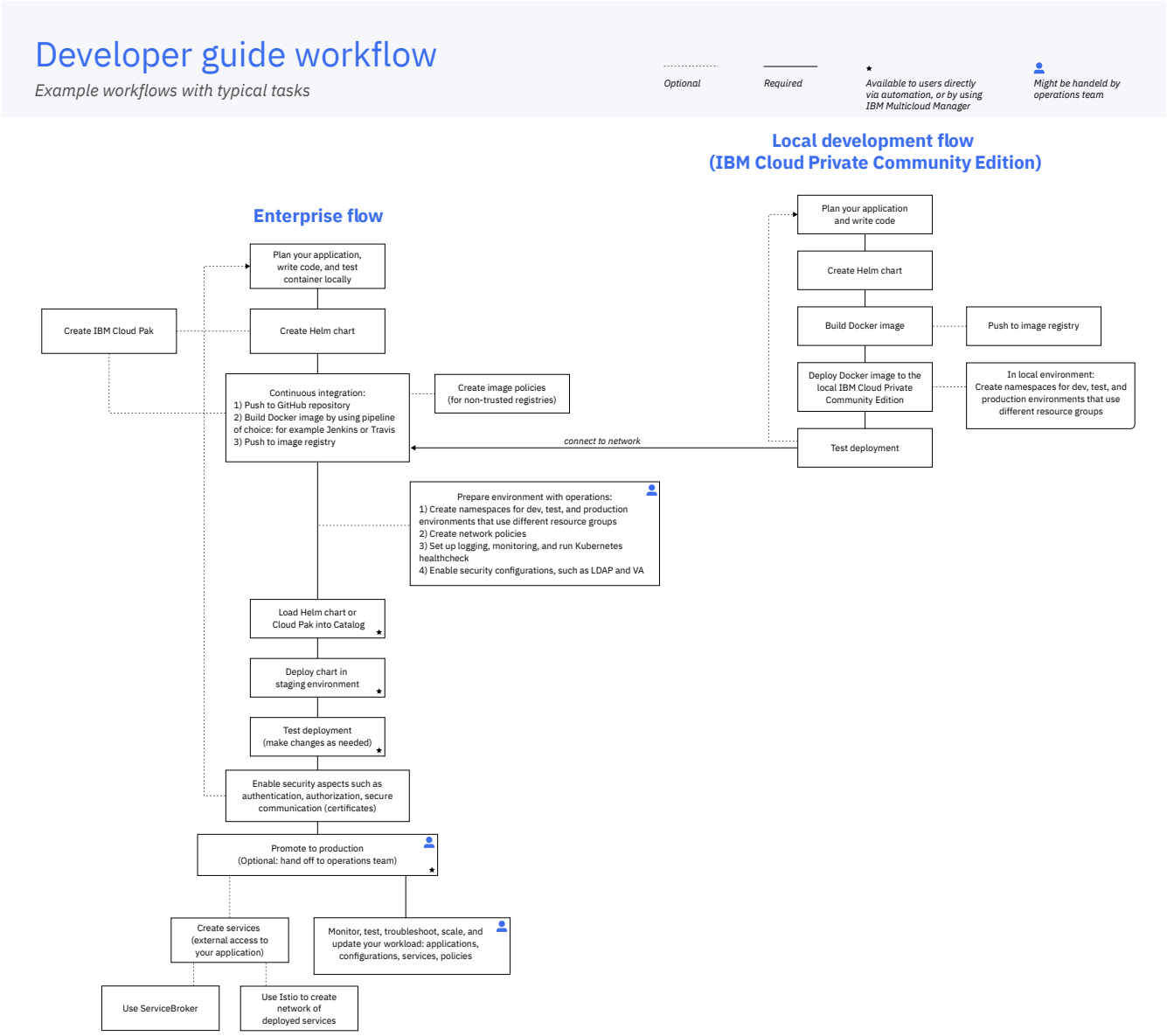
   Para obter mais informações sobre como configurar o calicoctl com o armazenamento de dados etcdv3, consulte Configurando o calicoctl para se conectar a um armazenamento de dados etcd.

7. Use a linha de comandos do Calico. Para iniciar com a linha de comandos do Calico, consulte Referência de comando.

Guia do desenvolvedor

Este guia contém os detalhes do uso do IBM Cloud Private, como o gerenciamento de IBM Cloud Paks, gráficos, aplicativos, serviços, imagens e cargas de trabalho.

O diagrama a seguir ilustra um fluxo de trabalho de exemplo com tarefas típicas que um desenvolvedor pode executar enquanto usa o IBM Cloud Private.



- Identificando os IBM Certified Containers
- Identificando o IBM Cloud Paks
- Gerenciando gráficos e apps
- Gerenciando serviços
- Catalog de Serviço
- Gerenciando imagens
- Gerenciando cargas de trabalho

Gerenciando gráficos e aplicativos

Usando o Catalog, é possível procurar e instalar pacotes em seu cluster do IBM® Cloud Private por meio de gráficos Helm.

O Catalog exibe gráficos Helm, que contêm pacotes de aplicativos que podem ser executados como serviços Kubernetes. Os pacotes estão armazenados em repositórios. O Catalog no IBM Cloud Private contém uma conexão do repositório por padrão, mas é possível se conectar a outros repositórios. Depois de se conectar a um repositório, é possível acessar seus gráficos no Catalog. Os desenvolvedores de aplicativos também podem desenvolver aplicativos e publicá-los no Catalog para que outros usuários possam acessar e instalá-los facilmente.

As ações que estão disponíveis para você no Catalog são determinadas pelo controle de acesso baseado na função. Para obter mais informações, consulte Controle de acesso baseado na função.

Localizando o Catalog

1. Efetue login no console de gerenciamento IBM Cloud Private.
2. A partir do menu de navegação horizontal, clique em **Catalog**

Procurando no Catalog

Na página Catalog, é possível visualizar os gráficos Helm por categoria. Clique em uma das categorias a seguir para visualizar os gráficos nessa categoria:

- Blockchain
- Automação de Negócios
- Dados
- Ciência de dados e Análise de dados
- DevOps
- Integração
- IoT
- Rede
- Operações
- Tempos de execução e estrutura
- Segurança
- Armazenamento
- Ferramentas
- Outras

Para uma navegação mais fácil, é possível procurar por itens e filtrar os gráficos do Helm com base nas classificações a seguir:

- Tipo de Release
  - Beta - gráficos de liberação inicial que são oficialmente anunciados.
  - Visualização técnica - gráficos que ainda não foram anunciados, mas estão disponíveis para revisão inicial.
  - Uso limitado - gráficos que podem ser usados para testes de desenvolvimento e avaliações (gráficos sem encargo)
  - Uso comercial - gráficos que podem ser usados para produção (gráficos pagos)
- Arquitetura
  - Linux®
  - Linux® on Power® (ppc64le)
  - Linux® on IBM® Z and LinuxONE
- Nome de Repositório

Certificações do gráfico Catalog

Alguns dos gráficos no Catalog possuem um badge de certificação associado a eles. Esse badge de certificação é incluído em entradas do IBM que foram testadas e que atendem aos critérios do nível de certificação indicado. Consulte Identificando IBM Cloud Paks para obter mais informações sobre o IBM Cloud Paks e suas certificações.

Acessando o Catalog usando servidores proxy

Para acessar o Catalog usando servidores proxy, é possível ativar o acesso ao proxy durante a instalação do cluster do IBM Cloud Private ou editando a implementação do Helm depois de instalar seu cluster.

- Para ativar o acesso ao proxy, durante a instalação, configure os parâmetros `tiller_http_proxy` e `tiller_https_proxy` no arquivo `config.yaml`. Veja Customizando o cluster com o arquivo config.yaml.
- Para ativar o acesso depois de instalar seu cluster, conclua as etapas a seguir:
  1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
  2. Para a implementação `helm-api`, selecione **Ação** > **Editar**. O arquivo JSON da implementação é exibido.
  3. Atualize as variáveis `HTTP_PROXY` e `HTTPS_PROXY` com os valores para suas URLs de proxy. Assegure-se de incluir protocolo e porta.
  4. Clique em **Enviar**. A implementação é reiniciada e o acesso ao catálogo por servidores proxy é configurado.

Revisando a coleção de gráficos Helm

Para obter mais informações sobre os aplicativos padrão que estão disponíveis no Catalog, consulte Serviços de destaque.

- Implementando gráficos Helm no Catalog

- Implementando gráficos Helm que requerem privilégios elevados em um namespace não padrão
- Trabalhando com gráficos

Implementando gráficos Helm no Catalog

O Catalog exibe os gráficos Helm que estão disponíveis para implementação.

Tipo de usuário ou nível de acesso necessário: administrador de cluster, administrador de equipe ou operador

Nota: consulte o arquivo leia-me ou a documentação do gráfico para determinar a função mínima que é necessária para instalar o gráfico.

1. Sincronize seus repositórios Helm, caso você ainda não tenha feito isso. Consulte Gerenciando repositórios do Helm para o procedimento.
2. Clique em **Catalog** .
3. Selecione o Gráfico Helm que deseja implementar.

   Um arquivo leia-me que inclui informações sobre instalação, desinstalação e configuração é fornecido para cada Gráfico Helm.
4. Se uma lista de gráficos Helm não for exibida no Catalog, verifique se você especificou o repositório do Helm e o namespace para sua equipe concluindo as etapas a seguir:

   1. No menu de navegação, selecione **Gerenciar** > **Identidade e acesso** > **Equipes**.
   2. Selecione a equipe que você deseja atualizar.
   3. Selecione **Recursos** e verifique se há um repositório do Helm e um namespace listado.
   4. Se for necessário incluir os recursos, selecione **Gerenciar recursos**.
   5. Marque a caixa de seleção para o repositório do Helm e o namespace que você deseja incluir na equipe.
   6. Selecione **Salvar** para salvar suas mudanças.
   7. Conclua o procedimento novamente, iniciando com a etapa 1.
5. Clique em **Configurar**.
6. Na página Configuração, nomeie sua liberação. O nome pode consistir apenas em caracteres alfanuméricos minúsculos ou caracteres de traço (-) e deve iniciar e terminar com um caractere alfanumérico.
7. Selecione o namespace para a implementação do gráfico, que preenche as políticas de segurança de pod que estão associadas ao namespace no campo *Políticas de Namespace de Destino* na seção Segurança de Pod. Isso fornece orientação sobre qual namespace selecionar e filtra as políticas para aquela que corresponde à política de segurança de pod que é definida pelo gráfico. Selecione um namespace com base no requisito de segurança de pod para o gráfico. As mensagens de aviso serão mostradas se você selecionar um namespace que não tenha uma das políticas compatíveis. Por exemplo, o sistema está em execução no modo restrito e você está instalando uma política irrestrita. **Nota:** ao implementar um gráfico em um managed-cluster, a lista de namespaces para seleção não se limita aos namespaces aos quais você tem permissão de acesso. Todos os namespaces possíveis são exibidos. Se selecionar um namespace ao qual você não tem acesso, a implementação do gráfico falhará e o pod não será iniciado. Ao implementar um gráfico em um único cluster local, a lista de namespaces é filtrada para exibir os namespaces que podem ser acessados.
8. Selecione os nomes dos clusters nos quais você deseja implementar o gráfico. É possível usar a função de procura para localizar um nome de cluster em uma longa lista de clusters.
9. Verifique a política de segurança de pod necessária para o gráfico na seção de segurança de Pod. Se o nome da política de segurança de pod for definido pelo gráfico na especificação do IBM Certified Container, os detalhes do nome da política de segurança de pod serão mostrados. Se o nome da política de segurança de pod não estiver definido, os detalhes necessários para selecionar um namespace com a política de segurança de pod menos restritiva serão exibidos. Um link para detalhes adicionais sobre a Política de Segurança de Pod é fornecido para obter mais informações.
10. (Opcional) Customize campos de acordo com sua preferência.
11. Confirme que você leu e concordou com o contrato de licença.
12. Clique em **Instalar** para implementar seu gráfico Helm e criar uma liberação. Uma liberação é uma instância de um gráfico que é executado em um cluster Kubernetes.

Implementando um IBM Cloud Pak

É possível localizar e implementar um IBM Cloud Pak a partir do IBM Cloud Private Catalog.

Um IBM Cloud Pak é listado no IBM Cloud Private Catalog sob o título *Solution Paks* para a versão 3.2.0. Conclua as etapas a seguir para instalar um IBM Cloud Pak a partir da console de gerenciamento:

1. Efetue login em seu cluster do IBM Cloud Private com um ID que tenha pelo menos as permissões de *Operador* para o cluster.
2. Na visualização da barra de título, clique em **Catalog** .
3. Na seção *Solution Paks*, selecione o IBM Cloud Pak que você deseja instalar.
4. Na guia **Configuração**, forneça os detalhes de configuração necessários.
5. Clique em **Instalar** . Isso cria um painel que é referido como o *Painel do Solution Cloud Pak*. É possível usar o painel do Solution Cloud Pak para instalar os IBM Certified Containers que estão incluídos no IBM Cloud Pak.
6. Clique em **Cargas de Trabalho** > **Liberações do Helm** para localizar o painel do Solution Cloud Pak em execução.
7. Procure pelo nome do IBM Cloud Pak.
8. Clique em **Ativar** para o IBM Cloud Pak, que inicia o painel do Solution Cloud Pak.
9. Na interface do IBM Cloud Pak, selecione as ofertas no IBM Cloud Pak que você deseja instalar.
10. Conclua os campos que são necessários para as ofertas. Os campos obrigatórios são indicados por um asterisco (*).

As ofertas selecionadas no IBM Cloud Pak são instaladas.

Implementando gráficos Helm que requerem privilégios elevados em um namespace não padrão

Implemente gráficos Helm que requerem privilégios elevados do Catalog do IBM® Cloud Private para namespaces designados que contêm políticas de segurança de pod definidas.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

1. Configure a CLI do `kubectl`. Consulte Acessando o cluster do IBM Cloud Private usando a CLI kubectl.
2. Crie uma política de segurança de pod. Por exemplo, para o gráfico de criação de log `ibm-icplogging`, é possível criar um pod `privileged` com o recurso `IPC_LOCK`.

   a. Crie um arquivo YAML com as definições de política.

   ```
   vim icplogging-psp.yaml

   apiVersion: extensions/v1beta1
   kind: PodSecurityPolicy
   metadata:
     name: icplogging
   spec:
     privileged: true
     allowedCapabilities:
     - IPC_LOCK
     fsGroup:
       rule: RunAsAny
     runAsUser:
       rule: RunAsAny
     seLinux:
       rule: RunAsAny
     supplementalGroups:
       rule: RunAsAny
     volumes:
   - "*"
   ```

   b. Crie a política.

   ```
   kubectl create -f icplogging-psp.yaml
   ```

   A saída se assemelha ao código a seguir:

```
podsecuritypolicy "icplogging" created
```

3. Crie uma função de cluster para o recurso de política de segurança de pod. O `resourceNames` para essa função deve ser o nome da política de segurança de pod que foi criada na etapa anterior.

   a. Crie um arquivo YAML para a função de cluster.

   ```
   vim icplogging-clusterrole.yaml
   ```

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRole
   metadata:
    name: icplogging-role
   rules:
    -
      apiGroups:
        - extensions
      resourceNames:
        - icplogging
      resources:
        - podsecuritypolicies
      verbs:
        - use
   ```

   b. Crie a função.

   ```
   kubectl create -f icplogging-clusterrole.yaml
   ```

   A saída se assemelha ao código a seguir:

   ```
    clusterrole "icplogging-role" created
   ```

4. Crie um namespace e configure a ligação de função de cluster para a conta do serviço nesse namespace. Usando essa ligação de função, é possível configurar as contas de serviço no namespace para usar a política de segurança de pod que você criou.

   Por exemplo, para criar um namespace `elk` com o serviço de ligação de função.

   a. Crie o namespace.

   ```
   kubectl create namespace elk
   ```

   b. Crie o serviço de ligação de função

   ```
   vim logging-clusterrolebinding.yaml
   ```

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
    name: logging-psp-users
   roleRef:
    apiGroup: rbac.authorization.k8s.io
    kind: ClusterRole
    name: icplogging-role
   subjects:
    -
      apiGroup: rbac.authorization.k8s.io
      kind: Group
      name: "system:serviceaccounts:elk"
   ```

   ```
   kubectl create -f logging-clusterrolebinding.yaml
   ```

   Nota: `roleRef` deve ser especificado antes do recurso `clusterrole` no arquivo YAML. Os `subjects` devem ser as contas do serviço para o namespace.

   A saída se assemelha ao código a seguir:

   ```
    clusterrolebinding "logging-psp-users" created
   ```

5. Implemente seu gráfico Helm. Consulte Implementando gráficos Helm no Catalog. Assegure-se de especificar o namespace que você criou na parte 4 como o namespace para seu gráfico.

Estendendo os parâmetros do gráfico Helm com metadados

Quando um gráfico do Helm é implementado na seção na console de gerenciamento do IBM® Cloud Private, uma seção *Parâmetros* é exibida na guia *Configuração*. A guia *Configuração* exibe todos os parâmetros que são declarados no arquivo `values.yaml` que é empacotado no arquivo `chart.tgz` com base na inferência de tipo. Os desenvolvedores de gráficos podem, opcionalmente, empacotar um arquivo `values-metadata.yaml` adicional no arquivo `chart.tgz`, que é usado pela console de gerenciamento do IBM® Cloud Private para aprimorar a experiência de implementação.

O arquivo values.yaml

Os gráficos Helm devem incluir um arquivo `values.yaml`, que declara os parâmetros para o gráfico. O formato desse arquivo não permite que os parâmetros sejam formalmente descritos.

O arquivo values-metadata.yaml

Os desenvolvedores de gráficos podem, opcionalmente, empacotar um arquivo `values-metadata.yaml` adicional no arquivo `chart.tgz`, que é usado pela console de gerenciamento do IBM® Cloud Private para aprimorar a apresentação dos parâmetros durante a implementação. Por exemplo, em vez de uma caixa de texto genérica para um parâmetro de sequência, o arquivo de metadados pode indicar que o parâmetro está limitado a um conjunto de sequências. Esse conjunto de sequências pode ser apresentado como uma lista de seleção suspensa para o usuário.

Os metadados definidos no arquivo `values-metadata.yaml` estendem as informações que estão no arquivo `values.yaml`. A estrutura dos valores `values-metadata.yaml` deve ser a mesma que a do arquivo `values.yaml`, exceto que cada propriedade é identificada como `__metadata`.

Quando o arquivo `values-metadata.yaml` ou um parâmetro específico está ausente, a console de gerenciamento do IBM® Cloud Private exibe todos os parâmetros padrão que são declarados no arquivo `values.yaml` com base no tipo de inferência de tipo.

Especificações de values-metadata.yaml

É possível definir os metadados para cada parâmetro no arquivo `values-metadata.yaml` iniciando-o com a chave `__metadata`. Cada parâmetro pode incluir um ou mais dos atributos a seguir:

Propriedade	Tipo de Dados	Valores Possíveis	Valor Padrão	Descrição	Aplicável ao agrupamento ___metadata
description	Sequência	Sequência (146 caracteres, ou menos)		Descrição do parâmetro. Isso deve aparecer em uma dica de ferramenta ou, se houver uma descrição do grupo, como um subcabeçalho.	true	hidden	Booleano	true, false	false	Se hidden = true, o elemento será ocultado (faz a mesma coisa que imutável, mas o campo de formulário nunca é exposto na IU).	false	immutable	Booleano	true, false	false	Se immutable = true, o usuário *não* terá permissão para modificar o parâmetro (field = disabled).	false	label	Sequência			Título do parâmetro. Se o rótulo não for especificado, então a chave do arquivo `values.yaml` será usada.	true	multiline	Booleano	true, false	false	Se type = string, exiba um campo de área de texto.| false	options	Matriz de Objetos, em que cada objeto tem o formato `label: any, value: any`.			Descreve um parâmetro que se transformaria em um menu suspenso.	false	required	Booleano	true, false	false	Descreve se o parâmetro é obrigatório. Se sim, um `*` será exibido ao lado do nome na console de gerenciamento.	false	type	Sequência	string, boolean, number, password		Tipo do parâmetro (nota: a matriz e o objeto também são `type: string` para resolver problemas de compatibilidade histórica de versão anterior)	false	validation	Sequência (regex)	qualquer regex Javascript válida.		Expressão regular para validar o valor do parâmetro.	false

Exemplo de estrutura de values-metadata.yaml

O conteúdo a seguir mostra um exemplo de como criar seus próprios parâmetros de metadados no arquivo `values-metadata.yaml`:

```
demonstration:
  __metadata:
    label: Demonstration
    description: I am an h2
  stringField:
    __metadata:
      label: String field
      type: string
      required: true
```

```
  numberField:
    __metadata:
      label: Number field (with validation)
      type: number
      required: true
  checkboxField:
    __metadata:
      label: Checkbox field
      type: boolean
      required: true
  selectField:
    __metadata:
      label: Select field
      type: string
      required: true
      options:
      - label: myOpt1
        value: myNotSelectedValue
      - label: myOpt2
        value: mySelectedValue
  multilineField:
    __metadata:
      label: Multiline field
      type: string
      multiline: true
      required: true
  immutableField:
    __metadata:
      label: Immutable field
      type: string
      required: true
      immutable: true
  arrayField:
    __metadata: ### arrayField: "[]" or [] must be set in values.yaml
      label: Array field
      type: string   ### do to backwards-compat bug, type must be string
      description: I am an array.
      required: true
  objectField: ### objectField: "{}" or {} must be set in values.yaml
    __metadata:
      label: Object field (new)
      type: string   ### do to backwards-compat bug, type must be string
      description: I am an object.
      required: true
```

Características de parâmetro

- `required`: o campo não poderá ficar vazio se o parâmetro `required` estiver configurado como true. Se estiver vazio, a instalação não será concluída e uma mensagem de erro será exibida.

Exemplo:

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: string
  immutable: false
  required: true
```

- `string`: elemento de entrada. Não há validação padrão.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: string
  immutable: false
  required: false
```

- `number`: elemento de entrada com type='number'. A validação do número é concluída.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: number
```

```
  immutable: false
  required: false
```

- `password`: elemento de entrada com type="password". Não há validação padrão.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: password
  required: true
```

- `options`: cria um menu suspenso com as opções.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: string
  required: false
  options:
  -
   label: foo
   value: 123
  -
   label: bar
   value: bar
```

- `boolean`: cria uma caixa de seleção.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: boolean
  required: true
```

- `multiline`: cria uma área de texto editável. Caracteres de nova linha são preservados.

```
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string
  multiline: true
```

- `immutable`: esta entrada deve permanecer como está e não pode ser editada.

```
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string
  immutable: true
```

- `hidden`: o valor necessário é fornecido para o arquivo `values.yaml`, mas não é exibido na console de gerenciamento.

```
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string
  hidden: true
```

- Widget yaml (array): o widget de matriz espera a entrada fornecida no formato YAML. A validação é executada no formato da matriz e uma mensagem será mostrada se o formato da matriz não for válido. **Nota:** devido a uma compatibilidade com um erro de versão anterior, o tipo *deve* ser `string`.

```
/* values.yaml */
loggingOptions: []

// ou

loggingOptions: "[]"

// ...to set a default for the field:

loggingOptions:
  - 1
  - 2
```

```
  - 3

/* values-metadata.yaml */
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string  ### because of a backwards compatibility bug, the type *must* be string.
  ...
```

- Widget yaml (objecto): o widget de objeto espera a entrada fornecida no formato YAML. A validação é executada no formato do objeto e uma mensagem será mostrada se o formato do objeto não for válido. **Nota:** devido a uma compatibilidade com um erro de versão anterior, o tipo *deve* ser `string`.

```
/* values.yaml */
loggingOptions: "{}"

// ou

loggingOptions: {}

// (there is currently no way to set a default)

/* values-metadata.yaml */
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string  ### because of a backwards compatibility bug, the type *must* be string.
  ...
```

Exemplo de usuário

O exemplo a seguir mostra como é possível usar o exemplo anterior para exibir seu conteúdo:

```
database:
  readinessProbePeriodSeconds:
    __metadata:
      label: I am a label
      description: I am a custom description
      type: number
      immutable: false
      required: false
  readinessProbeInitialDelaySeconds:
    __metadata:
      label: I am a label
      description: I am a custom description
      type: number
      required: false
  livenessProbePeriodSeconds:
    __metadata:
      label: I am a label
      type: number
      required: false
  livenessProbeInitialDelaySeconds:
    __metadata:
      label: I am a label
      description: I am a custom description
      type: number
      required: false
```

Trabalhando com gráficos

Para gerenciar gráficos do Kubernetes ou pacotes, em seu cluster, deve-se usar a interface da linha de comandos (CLI) Helm.

Para saber como instalar a CLI do Helm, consulte Instalando a CLI do Helm (helm). Depois de configurar a CLI do Helm, é possível criar ou atualizar pacotes em seu cluster.

- Incluindo o repositório interno do Helm na CLI do Helm
- Incluindo aplicativos customizados
- Incluindo aplicativos de destaque em clusters em um ambiente de airgap

Incluindo o repositório interno do Helm na CLI do Helm

O repositório interno do Helm denominado `local-charts` agora pode ser incluído na CLI do Helm como um repositório externo. `local-charts` pode ser usado como uma origem para instalar gráficos no cluster por meio da CLI do Helm.

1. (Opcional) Se o servidor principal de cluster não for resolvível por meio do servidor de nomes de domínio (DNS), será possível incluir `<Cluster Master Host>` como um alias no arquivo `/etc/hosts`. O alias aponta para o endereço ou endereços IP do nó principal.
   Nota: `<Cluster Master Host>` está definido em Terminais principais.
   Por exemplo, se o `<Cluster Master Host>` estiver definido como `mycluster.icp` e for resolvido para o IP `1.2.3.4`, inclua o seguinte alias:

   ```
   vi /etc/hosts
   # Add a line like the following:
   1.2.3.4 mycluster.icp
   ```

2. Importe e confie nos certificados para o cluster.

   1. Certifique-se de que HELM_HOME esteja definido para apontar para o diretório de trabalho do Helm (geralmente, `~/.helm`) inserindo o comando a seguir:

      ```
      $ export HELM_HOME=~/.helm
      ```

   2. Inicialize a CLI do Helm inserindo o comando a seguir:

      ```
      $ helm init --client-only
      ```

   3. Efetue logon no cluster do IBM® Cloud Private inserindo o comando a seguir:

      ```
      $ cloudctl login -a https://mycluster.icp:8443 --skip-ssl-validation
      ```

      Quando a última instrução de saída do comando `cloudctl login` lê que ele configurou o Helm com sucesso, os certificados foram copiados para o diretório *HELM_HOME*. **Nota:** Se não for possível efetuar logon no cluster, consulte Acessando o cluster a partir da CLI do Kubernetes (kubectl) para obter informações sobre como configurar a CLI.

   4. Insira o comando a seguir para definir o repositório `local-charts` para a CLI do Helm e especificar os certificados que você copiou na etapa anterior:

      ```
      $ sudo helm repo add local-charts https://mycluster.icp:8443/helm-repo/charts --ca-file
      $HELM_HOME/ca.pem --cert-file $HELM_HOME/cert.pem --key-file $HELM_HOME/key.pem
      ```

      Esses certificados são usados ao incluir o repositório Helm `local-charts` do IBM Cloud Private interno que é gerenciado pela implementação `helm-repo`.

   5. É possível inserir o comando a seguir para incluir o repositório Helm do IBM Cloud Private interno chamado `mgmt-charts` que é gerenciado pela implementação `mgmt-repo`:

      ```
      $ sudo helm repo add mgmt-charts https://mycluster.icp:8443/mgmt-repo/charts --ca-file
      $HELM_HOME/ca.pem --cert-file $HELM_HOME/cert.pem --key-file $HELM_HOME/key.pem
      ```

      Dica: em geral, não é necessário incluir o repositório `mgmt-charts`. Ele contém os gráficos Helm para serviços e recursos internos do IBM Cloud Private.

   6. Verifique se os repositórios estão acessíveis para a CLI do Helm inserindo o comando a seguir:

      ```
      $ helm repo list
      ```

Incluindo aplicativos customizados

Para incluir aplicativos customizados no Catalog do IBM® Cloud Private, crie um gráfico Helm.

Os pacotes Kubernetes que são gerenciados pelo Helm também são conhecidos como gráficos. Os termos *pacotes* e *gráficos* são referenciados de forma intercambiável neste guia do usuário.

É possível incluir gráficos criados por você ou transferidos por download a partir da Internet em um repositório externo do Helm ou no repositório interno do IBM Cloud Private.

- Incluindo aplicativos customizados
  - Empacote um gráfico Helm
  - Inclua o gráfico em um repositório externo
  - Inclua o gráfico no repositório interno

Empacote um gráfico Helm

1. Configure a interface da linha de comandos (CLI) do Helm. Consulte Instalando a CLI do Helm (helm) .
2. Crie um gráfico Helm. Por exemplo, para criar o gráfico do Helm `demoapp`, execute o comando a seguir:

   ```
   helm create demoapp
   ```

3. Visualize os conteúdos do gráfico Helm. Para visualizar o conteúdo do gráfico do Helm `demoapp`, execute o seguinte comando:

   ```
   tree demoapp
   ```

   O conteúdo do gráfico é exibido:

   ```
   demoapp
   |-- charts
   |-- Chart.yaml
   |-- templates
   |   |-- deployment.yaml
   |   |-- _helpers.tpl
   |   |-- NOTES.txt
   |   `-- service.yaml
   `-- values.yaml
   ```

4. Configure o gráfico Helm. Veja Introdução a um modelo de diagrama. É possível incluir informações do mantenedor, a versão do software, o código-fonte para o arquivo `Chart.yaml`. Também é possível modificar os arquivos `deployment.yaml` e `service.yaml` ou incluir mais diretórios de modelo.
5. Verifique se o gráfico está bem formatado. Execute o comando a seguir:

   ```
   helm lint --strict demoapp
   ```

   A saída se assemelha ao código a seguir:

   ```
   ==> Linting demoapp
   Lint OK

   1 chart(s) linted, no failures
   ```

6. Empacote seu gráfico e confirme que ele foi incluído.

   ```
   helm package demoapp ; ls -l
   ```

   Se o gráfico foi incluído, o pacote do gráfico estará no repositório local. Na saída a seguir, é mostrado o pacote `demoapp-0.1.0.tgz`:

   ```
   drwxr-xr-x 4 root root     4096 Jan  8 10:23 demoapp
   -rw-r--r-- 1 root root     1768 Jan  8 10:37 demoapp-0.1.0.tgz
   ```

7. Faça upload das imagens do Docker para o gráfico no registro de origem que ele especifica. Se você usar o registro de imagem privado que é fornecido com o IBM Cloud Private, consulte Enviando por push e efetuando pull de imagens.
8. Inclua o gráfico em um repositório externo ou no repositório interno do IBM Cloud Private.

   - Inclua o gráfico em um repositório externo
   - Inclua o gráfico no repositório interno

Inclua o gráfico em um repositório externo

Depois de empacotar o gráfico Helm, é possível incluí-lo em um repositório externo para torná-lo disponível no IBM Cloud Private.

1. Atualize o índice do repositório de gráfico remoto.

   1. Obtenha o arquivo `index.yaml` para o repositório de gráfico remoto.

```
wget http://<remote_host>/charts/index.yaml
```

2. Atualize o arquivo `index.yaml` com base em seu diagrama.

   ```
   ./helm repo index --merge index.yaml --url http://<remote_host>/charts/ ./
   ```

   A opção `--url` especifica o local do repositório do gráfico. Esse caminho é o local no qual a CLI do Helm obtém o novo pacote do gráfico.

2. Faça upload de seu gráfico para o repositório remoto.

   Copie o `index.yaml` atualizado e os novos arquivos do pacote do gráfico de seu host local para o diretório correspondente no host remoto.

3. Atualize o repositório de gráfico usando a console de gerenciamento do cluster do IBM Cloud Private

Tipo de usuário ou nível de acesso necessário para a sincronização, inclusão ou remoção de repositórios: Administrador de cluster

1. No IBM Cloud Private console de gerenciamento, clique em **Menu** > **Gerenciar** > **Repositórios de Helm**.
2. Se você não vir o repositório na lista, inclua-o. Consulte Incluindo um repositório de Helm.
3. Clique em **Sincronizar repositórios**. **Dica:** Também é possível sincronizar um único repositório, selecionando o menu de ação (**...**), em seguida, selecionando **Sincronizar este repositório**.
4. Clique em **Catalog** . Os novos gráficos de Helm são carregados no Catalog e é possível instalá-los em seu cluster.

Inclua o gráfico no repositório interno

Depois de empacotar o gráfico Helm, é possível incluí-lo no repositório interno que é fornecido com o IBM Cloud Private.

Tipo de usuário ou nível de acesso necessário: administrador de cluster

Antes de carregar o gráfico, conclua os pré-requisitos a seguir:

- Instale a CLI do IBM Cloud Private e efetue login em seu cluster. Consulte Instalando a CLI do IBM Cloud Private.
- Inclua o endereço IP do cluster e o nome de domínio da autoridade de certificação do cluster nos arquivos host conforme mostrado na etapa 1 de Configurando a autenticação para a CLI do Docker.
- Empacote o gráfico Helm.
- Caso não tenha feito isso, efetue login no cluster a partir da CLI do IBM Cloud Private e efetue login no registro de imagem privado do Docker.

  ```
  cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
  ```

  Em que `<Cluster Master Host>:<Cluster Master API Port>` está definido em Terminal principal.

- Instale o gráfico Helm:

  ```
  cloudctl catalog load-chart --archive <helm_chart_archive>
  ```

  Aqui, `helm_chart_archive` é o nome do arquivo de gráfico do Helm compactado.

Estendendo os parâmetros do gráfico Helm com metadados

Quando um gráfico do Helm é implementado na seção na console de gerenciamento do IBM® Cloud Private, uma seção *Parâmetros* é exibida na guia *Configuração*. A guia *Configuração* exibe todos os parâmetros que são declarados no arquivo `values.yaml` que é empacotado no arquivo `chart.tgz` com base na inferência de tipo. Os desenvolvedores de gráficos podem, opcionalmente, empacotar um arquivo `values-metadata.yaml` adicional no arquivo `chart.tgz`, que é usado pela console de gerenciamento do IBM® Cloud Private para aprimorar a experiência de implementação.

O arquivo values.yaml

Os gráficos Helm devem incluir um arquivo `values.yaml`, que declara os parâmetros para o gráfico. O formato desse arquivo não permite que os parâmetros sejam formalmente descritos.

O arquivo values-metadata.yaml

Os desenvolvedores de gráficos podem, opcionalmente, empacotar um arquivo `values-metadata.yaml` adicional no arquivo `chart.tgz`, que é usado pela console de gerenciamento do IBM® Cloud Private para aprimorar a apresentação dos parâmetros durante a implementação. Por exemplo, em vez de uma caixa de texto genérica para um parâmetro de sequência, o arquivo de metadados pode indicar que o parâmetro está limitado a um conjunto de sequências. Esse conjunto de sequências pode ser apresentado como uma lista de seleção suspensa para o usuário.

Os metadados definidos no arquivo `values-metadata.yaml` estendem as informações que estão no arquivo `values.yaml`. A estrutura dos valores `values-metadata.yaml` deve ser a mesma que a do arquivo `values.yaml`, exceto que cada propriedade é identificada como `__metadata`.

Quando o arquivo `values-metadata.yaml` ou um parâmetro específico está ausente, a console de gerenciamento do IBM® Cloud Private exibe todos os parâmetros padrão que são declarados no arquivo `values.yaml` com base no tipo de inferência de tipo.

Especificações de values-metadata.yaml

É possível definir os metadados para cada parâmetro no arquivo `values-metadata.yaml` iniciando-o com a chave `__metadata`. Cada parâmetro pode incluir um ou mais dos atributos a seguir:

Propriedade	Tipo de Dados	Valores Possíveis	Valor Padrão	Descrição	Aplicável ao agrupamento ___metadata
description	Sequência	Sequência (146 caracteres, ou menos)		Descrição do parâmetro. Isso deve aparecer em uma dica de ferramenta ou, se houver uma descrição do grupo, como um subcabeçalho.	true	hidden	Booleano	true, false	false	Se hidden = true, o elemento será ocultado (faz a mesma coisa que imutável, mas o campo de formulário nunca é exposto na IU).	false	immutable	Booleano	true, false	false	Se immutable = true, o usuário *não* terá permissão para modificar o parâmetro (field = disabled).	false	label	Sequência			Título do parâmetro. Se o rótulo não for especificado, então a chave do arquivo `values.yaml` será usada.	true	multiline	Booleano	true, false	false	Se type = string, exiba um campo de área de texto.| false	options	Matriz de Objetos, em que cada objeto tem o formato `label: any, value: any`.			Descreve um parâmetro que se transformaria em um menu suspenso.	false	required	Booleano	true, false	false	Descreve se o parâmetro é obrigatório. Se sim, um `*` será exibido ao lado do nome na console de gerenciamento.	false	type	Sequência	string, boolean, number, password		Tipo do parâmetro (nota: a matriz e o objeto também são `type: string` para resolver problemas de compatibilidade histórica de versão anterior)	false	validation	Sequência (regex)	qualquer regex Javascript válida.		Expressão regular para validar o valor do parâmetro.	false	

Exemplo de estrutura de values-metadata.yaml

O conteúdo a seguir mostra um exemplo de como criar seus próprios parâmetros de metadados no arquivo `values-metadata.yaml`:

```
demonstration:
  __metadata:
    label: Demonstration
    description: I am an h2
  stringField:
    __metadata:
      label: String field
      type: string
      required: true
  numberField:
    __metadata:
      label: Number field (with validation)
      type: number
      required: true
  checkboxField:
    __metadata:
      label: Checkbox field
      type: boolean
      required: true
  selectField:
    __metadata:
      label: Select field
      type: string
      required: true
      options:
```

```
      - label: myOpt1
        value: myNotSelectedValue
      - label: myOpt2
        value: mySelectedValue
  multilineField:
    __metadata:
      label: Multiline field
      type: string
      multiline: true
      required: true
  immutableField:
    __metadata:
      label: Immutable field
      type: string
      required: true
      immutable: true
  arrayField:
    __metadata: ### arrayField: "[]" or [] must be set in values.yaml
      label: Array field
      type: string   ### do to backwards-compat bug, type must be string
      description: I am an array.
      required: true
  objectField: ### objectField: "{}" or {} must be set in values.yaml
    __metadata:
      label: Object field (new)
      type: string   ### do to backwards-compat bug, type must be string
      description: I am an object.
      required: true
```

Características de parâmetro

- `required`: o campo não poderá ficar vazio se o parâmetro `required` estiver configurado como true. Se estiver vazio, a instalação não será concluída e uma mensagem de erro será exibida.

Exemplo:

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: string
  immutable: false
  required: true
```

- `string`: elemento de entrada. Não há validação padrão.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: string
  immutable: false
  required: false
```

- `number`: elemento de entrada com type='number'. A validação do número é concluída.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: number
  immutable: false
  required: false
```

- `password`: elemento de entrada com type="password". Não há validação padrão.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: password
  required: true
```

- `options`: cria um menu suspenso com as opções.

```
__metadata:
  label: Datacenter
  description: Datacenter description
```

```
  type: string
  required: false
  options:
  -
   label: foo
   value: 123
  -
   label: bar
   value: bar
```

- `boolean`: cria uma caixa de seleção.

```
__metadata:
  label: Datacenter
  description: Datacenter description
  type: boolean
  required: true
```

- `multiline`: cria uma área de texto editável. Caracteres de nova linha são preservados.

```
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string
  multiline: true
```

- `immutable`: esta entrada deve permanecer como está e não pode ser editada.

```
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string
  immutable: true
```

- `hidden`: o valor necessário é fornecido para o arquivo `values.yaml`, mas não é exibido na console de gerenciamento.

```
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string
  hidden: true
```

- Widget yaml (array): o widget de matriz espera a entrada fornecida no formato YAML. A validação é executada no formato da matriz e uma mensagem será mostrada se o formato da matriz não for válido. **Nota:** devido a uma compatibilidade com um erro de versão anterior, o tipo *deve* ser `string`.

```
/* values.yaml */
loggingOptions: []

// ou

loggingOptions: "[]"

// ...to set a default for the field:

loggingOptions:
  - 1
  - 2
  - 3

/* values-metadata.yaml */
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string  ### because of a backwards compatibility bug, the type *must* be string.
  ...
```

- Widget yaml (objecto): o widget de objeto espera a entrada fornecida no formato YAML. A validação é executada no formato do objeto e uma mensagem será mostrada se o formato do objeto não for válido. **Nota:** devido a uma compatibilidade com um erro de versão anterior, o tipo *deve* ser `string`.

```
/* values.yaml */
loggingOptions: "{}"
```

```
// ou

loggingOptions: {}

// (there is currently no way to set a default)

/* values-metadata.yaml */
__metadata:
  label: loggingOptions
  description: loggingOptions
  type: string  ### because of a backwards compatibility bug, the type *must* be string.
  ...
```

Exemplo de usuário

O exemplo a seguir mostra como é possível usar o exemplo anterior para exibir seu conteúdo:

```
database:
  readinessProbePeriodSeconds:
    __metadata:
      label: I am a label
      description: I am a custom description
      type: number
      immutable: false
      required: false
  readinessProbeInitialDelaySeconds:
    __metadata:
      label: I am a label
      description: I am a custom description
      type: number
      required: false
  livenessProbePeriodSeconds:
    __metadata:
      label: I am a label
      type: number
      required: false
  livenessProbeInitialDelaySeconds:
    __metadata:
      label: I am a label
      description: I am a custom description
      type: number
      required: false
```

Incluindo aplicativos de destaque em clusters em um ambiente de airgap

É possível incluir aplicativos de destaque no IBM® Cloud Private Catalog em clusters que não têm conectividade com a Internet.

Para incluir aplicativos que fazem parte de pacotes configuráveis do IBM Cloud Private, consulte Instalando o software IBM no IBM Cloud Private.

Os aplicativos que têm gráficos do Helm podem ser compactados em archives que contêm o gráfico do Helm e as imagens de contêiner com a CLI do IBM Cloud Private em um sistema operacional conectado à Internet. A partir da CLI do IBM Cloud Private, é possível carregar seu archive criado para o cluster em um ambiente de airgap.

Antes de incluir um aplicativo, conclua os seguintes pré-requisitos:

- Instale a interface da linha de comandos (CLI) do IBM Cloud Private em seu sistema operacional para criar o archive. Consulte Instalando a CLI do IBM Cloud Private.
- Instale o certificado de registro para que seja possível usar o registro do Docker para o IBM Cloud Private em seu sistema operacional para criar o archive. Consulte API do Docker Registry V2 para obter informações adicionais.
- Instale a CLI do Helm no sistema operacional que está sendo usado para criar o archive. Para obter mais informações, consulte Instalando a CLI do Helm (helm).

Nível de acesso necessário: Administrador de cluster

Importante: Certifique-se de que tenha espaço suficiente disponível no repositório local do Docker e no disco para criar o archive. Os gráficos podem requerer várias imagens dependentes.

Incluindo aplicativos de destaque a partir do repositório *IBM/charts*

O repositório de gráficos do Helm do IBM Cloud contém gráficos do Helm para muitos aplicativos.

- A partir da CLI do IBM Cloud Private, forneça o caminho para o gráfico do Helm que você deseja usar para criar o archive. Muitos dos gráficos do Helm no repositório *IBM/charts* contêm um arquivo `manifest.yaml` no diretório `ibm_cloud_pak`. Se o arquivo `manifest.yaml` existir, o aplicativo suportará diretamente esse processo. Se o arquivo não existir, o aplicativo poderá funcionar se atender a determinados critérios. Visualize a seção *Incluindo aplicativos por gráfico do Helm* para obter os critérios de inclusão de um aplicativo usando somente o gráfico do Helm.
- Os archives suportam uma ou mais arquiteturas de CPU. Se o arquivo `ibm_cloud_pak/manifest.yaml` existir, as arquiteturas da CPU para as quais o archive foi criado serão definidas. Consulte o arquivo README.md do gráfico do Helm para arquiteturas de CPU suportadas. Se o arquivo não existir, o archive será criado para uma arquitetura de CPU do sistema operacional no qual o archive é criado. Se o arquivo não existir e for necessário um archive multiarquitetura, deve-se criar um arquivo `manifest.yaml`. Siga as instruções na seção *Criando um archive a partir de um arquivo YAML* para criar um arquivo `manifest.yaml`.

Para criar um archive a partir de um aplicativo que está no repositório *IBM/charts*, conclua as seguintes etapas:

1. Clone o repositório executando o seguinte comando:

   ```
   git clone https://github.com/IBM/charts.git
   ```

2. Efetue login em seu cluster executando o comando a seguir:

   ```
   cloudctl login -a <cluster_ip> --skip-ssl-validation
   ```

3. Efetue login em um registro do Docker executando o seguinte comando:

   ```
   docker login <cluster_hostname>:<registry_port>
   ```

   Para efetuar login em um registro do Docker denominado `mycluster.icp:8500`, execute o comando a seguir:

   ```
   docker login mycluster.icp:8500
   ```

4. Crie o archive executando o seguinte comando:

   ```
   cloudctl catalog create-archive -c charts/stable/<helm-chart-name> -a <path-to-archive-file-to-create>
   ```

   - *helm-chart-name* é o nome do diretório do gráfico do Helm.
   - *path-to-archive-file-to-create* é o caminho relativo ou absoluto para o diretório no qual você deseja salvar o archive que está sendo criado.

     Consulte o seguinte comando de exemplo:

     ```
     cloudctl catalog create-archive -c charts/stable/ibm-transadv-dev/ -a ibm-transadv-dev.tgz
     ```

     Para obter informações adicionais sobre o comando `cloudctl catalog create-archive`, consulte Comandos do catálogo da CLI do IBM Cloud Private (catálogo).

5. Carregue o archive executando o seguinte comando:

   ```
   cloudctl catalog load-archive --archive ibm-transadv-dev.tgz --registry <cluster_hostname>:<registry_port>/<namespace>
   ```

   Para fazer upload do archive para um registro do Docker chamado `mycluster.icp:8500` usando o namespace `kube-system`, execute o seguinte comando:

   ```
   cloudctl catalog load-archive --archive ibm-transadv-dev.tgz --registry mycluster.icp:8500/kube-system
   ```

   Para macOS, é possível carregar o archive configurando as variáveis de ambiente `DOCKER_USER` e `DOCKER_PWD` ou executar o seguinte comando:

   ```
   cloudctl catalog load-archive --archive ibm-transadv-dev.tgz --registry mycluster.icp:8500/kube-system --username <username> --password <password>
   ```

Scripts de bash de exemplo para carregar múltiplos aplicativos a partir de *IBM/charts*

Depois de concluir as três primeiras etapas da seção anterior, é possível carregar vários aplicativos a partir do repositório de gráficos do Helm do IBM Cloud. Conclua as seguintes etapas para carregar vários aplicativos:

1. Conclua as etapas 1, 2 e 3 da seção anterior.

2. Crie um arquivo `create-archives.sh` e copie o seguinte conteúdo para seu arquivo. Edite as variáveis de matriz `_white_list` e `_black_list` no diretório atual com o seguinte conteúdo:

```
#!/bin/bash

set -e

# CONFIGURATION - modify these variables to control which Helm charts to create archives for
# _white_list is a space-separated set of directories or files to create archives for
#   - leave empty to create for everything in the charts directory
#   - values should be relative to the charts directory
_white_list=$(cat <<- EOM
EOM
)

# _black_list is a space-separated set of directories or files to not create archives for
#   - leave empty to create for everything in the charts directory
#   - values should be relative to the charts directory
_black_list=$(cat <<- EOM
   ibm-ace-dev ibm-cam ibm-cam-prod ibm-cem ibm-db2oltp-dev ibm-db2warehouse-dev ibm-dsm-dev
ibm-dsx-dev
   ibm-eventstreams-dev ibm-f5bigip-controller ibm-iisee-eval ibm-spectrum-conductor ibm-was-
vm-quickstarter
EOM
)

# ARGUMENTS -
# $1 (required) - the directory that contains Helm chart archive files or Helm chart
directories
_charts_dir=${1%/} # remove a trailing slash
if [[ "$_charts_dir" == "" ]]; then
    echo "Provide the path to the directory containing Helm chart source directories or
archive files.   Example: ~/charts/stable"
   exit 1
fi

# $2 (optional) - a destination path for the created archives
_dest_dir=${2:-$(pwd)}
_dest_dir=${_dest_dir%/} # remove a trailing slash
mkdir -p _dest_dir

_charts=$(ls $_charts_dir)
for _chart in ${_charts[@]}
do
    _helm_chart="$_charts_dir/$_chart"
    if [ "$_white_list" != "" ]; then
        if [[ $_white_list = *"$_chart"* ]]; then
            cloudctl catalog create-archive -c $_helm_chart -a $_dest_dir/$_chart.tgz
            echo ""
        fi
    elif [[ $_black_list != "" && $_black_list = *"$_chart"* ]]; then
        echo "Skipping $_chart"
    elif [ -d "$_helm_chart" -o "${_helm_chart: -4}" == ".tgz" -o "${_helm_chart: -7}" ==
".tar.gz" ]; then
        cloudctl catalog create-archive -c $_helm_chart -a $_dest_dir/$_chart.tgz
        echo ""
    fi
done
```

3. Crie os archives executando o seguinte comando:

```
sh create-archives.sh charts/stable archives
```

4. Crie um arquivo `load-archives.sh` no diretório atual com o seguinte conteúdo:

```
#!/bin/bash

set -e # keep going even if one chart fails
```

```
 # ARGUMENTS -
# $1 (required) - the directory that contains the archives to load
_archives_dir=${1%/} # remove a trailing slash
if [[ "$_archives_dir" == "" ]]; then
   echo "Provide the path to the directory containing the archives to load."
     saída 1 fi

 _archives=$(ls $_archives_dir)
 for _archive in ${_archives[@]}
 do
     if [ "${_archive: -4}" == ".tgz" -o "${_archive: -7}" == ".tar.gz" ]; then
         echo "Loading archive $_archive ..."
         cloudctl catalog load-archive -a "$_archives_dir/$_archive"
         echo ""
     fi
 done
```

- Para macOS, é possível carregar os archives configurando as variáveis de ambiente `DOCKER_USER` e `DOCKER_PWD` ou executar o seguinte comando:

  ```
  cloudctl catalog load-archive --archive ibm-transadv-dev.tgz --registry
  mycluster.icp:8500/kube-system --username <username> --password <password>
  ```

5. Carregue os archives executando o seguinte comando:

   ```
   sh load-archives.sh archives
   ```

Incluindo aplicativos com um gráfico do Helm

Um archive pode ser criado somente com a entrada de um gráfico do Helm. Esse método pode ser usado para gráficos de comunidade de software livre e gráficos privados se o conteúdo do gráfico atender às seguintes condições:

- Todos os valores de repositório de imagem devem ser definidos no arquivo `values.yaml` do gráfico. Não é possível incluir valores de repositório de imagem codificados permanentemente nos arquivos YAML de modelo.
- O gráfico do Helm não deve usar subgráficos.
- O archive criado define somente a arquitetura de CPU única.

Importante: O método para incluir aplicativos com um gráfico do Helm pode não funcionar para todos os gráficos do Helm. Se tiver problema com este método, siga as etapas da seção *Criando um archive com um arquivo YAML manifest*.

Criando um archive com um arquivo YAML manifest

É possível criar um archive criando um arquivo YAML manifest que define o gráfico e as imagens que vão para o archive. Conclua as seguintes etapas para criar um archive com um arquivo YAML manifest:

1. Crie o arquivo YAML manifest chamado `manifest.yaml`.

   Nota: O atributo `archive` deve ser um URI em um dos seguintes formatos: `http://`, `https://` ou `file://`. Para URIs que usam `file://`, use um caminho absoluto ou um caminho relativo para o arquivo `manifest.yaml`.

   - Crie um arquivo YAML manifest para um archive de arquitetura de CPU única. O arquivo `manifest.yaml` pode ser semelhante ao seguinte conteúdo:

     ```
     charts:
     - archive: <URI to the Helm chart archive file>
       repository-keys:
       - <values.yaml image key in dot notation to modify at load time>
       registry-keys:
       - <values.yaml image registry key in dot notation to replace at load time>
     images:
     - image: <the image name and tag to be loaded into the image registry>
       references:
       - repository: <the same value as the previous image>
         pull-repository: <the image and tag to pull in to the archive>
     ```

   - Visualize o arquivo `manifest.yaml` a partir do gráfico `redis-ha` no repositório do Helm, considerando que o gráfico está compactado como `redis-ha-3.1.5.tgz` no mesmo diretório que o arquivo `manifest.yaml`:

     ```
     charts:
     - archive: file://redis-ha-3.1.5.tgz
       repository-keys:
     ```

```
  - imagens image.repository:
- image: redis:5.0.3
  references:
  - repository: redis:5.0.3
    pull-repository: redis:5.0.3
```

- Crie um arquivo YAML manifest para vários archives de arquitetura da CPU:

```
charts:
- archive: <URI to the Helm chart archive file>
  repository-keys:
  - <values.yaml image key in dot notation to modify at load time>
  registry-keys:
  - <values.yaml image registry key in dot notation to replace at load time>
images:
- image: <the image name and tag to be loaded into the image registry>
  references:
  - repository: <a different value than the image value above so it is architecture
specific>
    pull-repository: <the image and tag to pull in to the archive for the architecture>
    platform:
      os: linux
      architecture: <amd64|ppc64le|s390x>
```

- Visualize o arquivo `manifest.yaml` de vários archives de arquitetura da CPU para o gráfico `redis-ha` no repositório do Helm, considerando que o gráfico esteja compactado como `redis-ha-3.1.5.tgz` no mesmo diretório que o arquivo `manifest.yaml`:

```
charts:
- archive: file://redis-ha-3.1.5.tgz
  repository-keys:
  - imagens image.repository:
- image: redis:5.0.3
  references:
  - repository: redis-amd64:5.0.3
    pull-repository: amd64/redis:5.0.3
    platform:
      os: linux
      architecture: amd64
  - repository: redis-ppc64le:5.0.3
    pull-repository: ppc64le/redis:5.0.3
    platform:
      os: linux
      architecture: ppc64le
  - repository: redis-s390x:5.0.3
    pull-repository: s390x/redis:5.0.3
    platform:
      os: linux
      architecture: s390x
```

2. Crie o archive executando o seguinte comando:

```
cloudctl catalog create-archive -s <path-to-yaml-file> -a <path-to-archive-file>
```

- *path-to-yaml-file* é o caminho relativo ou absoluto para o local do arquivo YAML.
- *path-to-archive-file* é o caminho relativo ou absoluto para o diretório no qual você deseja salvar o archive que está sendo criado.
  - Crie um archive para seu gráfico do Helm. Seu comando pode ser semelhante à seguinte linha:

```
cloudctl catalog create-archive -s manifest.yaml -a <helm-chart-name>redis-ha-3.1.5-
archive.tgz
```

  - Crie um archive para o gráfico `redis-ha` executando o seguinte comando:

```
cloudctl catalog create-archive -s manifest.yaml -a redis-ha-3.1.5-archive.tgz
```

3. Efetue login no registro do Docker executando o seguinte comando:

```
docker login <cluster_hostname>:<registry_port>
```

Para efetuar login em um registro do Docker denominado `mycluster.icp:8500`, execute o comando a seguir:

```
docker login mycluster.icp:8500
```

4. Carregue o archive a partir do gráfico do Helm executando o seguinte comando:

```
cloudctl catalog load-archive --archive <archive_helm_chart_name> --registry
<cluster_hostname>:<registry_port>/<namespace>
```

   - Para fazer upload do archive do gráfico `redis-ha` para um registro do Docker chamado `mycluster.icp:8500` usando o namespace `kube-public`, execute o seguinte comando:

     ```
     cloudctl catalog load-archive --archive redis-ha-3.1.5-archive.tgz --registry
     mycluster.icp:8500/kube-public
     ```

   - Para macOS, é possível carregar o archive, configurando as variáveis de ambiente `DOCKER_USER` e `DOCKER_PWD` ou executar o seguinte comando:

     ```
     cloudctl catalog load-archive --archive redis-ha-3.1.5-archive.tgz --registry
     mycluster.icp:8500/kube-public --username <username> --password <password>
     ```

Os aplicativos de destaque do IBM Cloud Private Catalog são incluídos no cluster em um ambiente de airgap.

Serviços

Forneça acesso externo ao seu aplicativo expondo seu aplicativo como um serviço.

Criando Serviços

Para permitir o acesso à sua implementação de fora de sua rede, exponha sua implementação como um serviço.

Dois formatos estão disponíveis para você criar um serviço na console de gerenciamento.

É possível criar serviços inserindo os valores de parâmetro na janela Criar serviço ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando um serviço usando a janela Criar serviço

1. No menu de navegação, clique em **Acesso à rede** > **Serviços**.
2. Clique em **Criar Serviço**.

3. Forneça os detalhes do serviço. Forneça valores individuais na janela Criar serviço.

   Deve-se fornecer valores para vários parâmetros:

   - Na guia Geral, forneça esses valores:
     - **Nome** - Um nome para seu serviço
     - **Método de exposição** - Selecione o método para expor a implementação.
       - **ClusterIP**
       - **NodePort**
     - **IP do cluster** - Se você não especificar o endereço IP do cluster, um será gerado automaticamente.
     - **Afinidade de sessão**
   - Na guia Portas, forneça esses valores:
     Defina os detalhes da conexão.
     - Na lista **protocolo**, selecione a conexão para o protocolo.
     - Insira o nome da conexão.
     - No campo **porta**, especifique o ponto de acesso a serviço, como 80.

        - No campo **targetPort**, especifique o número da porta que o serviço que está em execução dentro do contêiner usa, como 8080.
        - Se você usar o método NodePort, no campo **NodePort**, especifique a porta de nó, como `31888`. O intervalo de portas do nó padrão é 31000 - 32000.
    - Na guia Seletores, forneça estes valores: como pares chave-valor, especifique a implementação para a qual você deseja criar um serviço.
        - seletor - app
        - valor - forneça o nome da implementação
4. Clique em **Criar**.

Criando um serviço usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar um serviço do Kubernetes usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service.
3. Clique em **Criar**.

Alterando um Serviço

Atualizar um serviço.

1. No menu de navegação, clique em **Acesso à rede** > **Serviços**.
2. Selecione **Ação** > **Editar**. O arquivo JSON do serviço é exibido.
3. Atualize as propriedades do serviço.
4. Clique em **Enviar**.

Removendo serviços

Remover o serviço que concede acesso externo por meio de uma implementação.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. No menu de navegação, clique em **Acesso à rede** > **Serviços**.
2. Para o serviço que você deseja remover, selecione **Ação** > **Remover**. A janela "Remover implementação" é exibida.
3. Clique em **Remover**.

O serviço selecionado é removido da lista de serviços.

Criando um ID de serviço usando a CLI do IBM Cloud Private

Pré-requisitos

- Instale e configure a ferramenta de linha de comandos do Kubernetes, kubectl. Veja Instalar e configurar o kubectl. **Nota:** deve-se instalar a versão 1.10.1.
- Instale e configure a CLI do IBM Cloud. Consulte Visão geral de CLI e de ferramentas para começar.
- Instale o plug-in CLI do IBM Cloud Private. Consulte Instalando a CLI do IBM Cloud Private para iniciar.

Criando um ID de serviço

1. Efetue login no IBM Cloud Private e configure o namespace para gerar tokens. O comando solicita uma senha e uma conta.

   ```
   cloudctl login -a https://<cluster-domain-name>:8443 -u <username> -n kube-system --skip-ssl-validation
   ```

2. Execute os comandos a seguir para listar comandos e dados do IAM:

    - `Cloudctl iam`

```
Saída
------
NAME:
 cloudctl iam - Manage identities and access to resources
USAGE:
 cloudctl iam command [arguments...][command options]

COMMANDS:
 accounts                  List all accounts
 api-key                   List details of an API key
 api-key-create            Create an API key
 api-key-delete            Delete an API key
 api-key-update            Update an API key
 api-keys                  List all API keys
 group-import              Import a group from an LDAP connection
 group-remove              Remove one or more group(s)
 groups                    List all imported groups
 ldap-create               Create new LDAP connection
 ldap-delete               Delete a LDAP connection
 ldap-get                  Get LDAP connection details
 ldaps                     List all LDAP connections
 resource-add              Add a resource to a team
 resource-rm               Remove a resource from a team
 resources                 List resources
 roles                     List roles
 saml-disable              Disable SAML authentication
 saml-enable               Enable SAML authentication
 saml-export-metadata      Export the SAML metadata content to create a SAML integration.
Requer que SAML seja ativado com 'cloudctl iam saml-enable'.
 saml-status              Get the SAML configuration status.
 saml-upload-metadata     Upload SAML metadata content to complete the SAML integration.
 service-api-key           List details of a service API key
 service-api-key-create    Create a service API key
 service-api-key-delete    Delete a service API key
 service-api-key-update    Update a service API key
 service-api-keys          List all API keys of a service
 service-id                Display details of a service ID
 service-id-create         Create a service ID
 service-id-delete         Delete a service ID
 service-id-update         Update a service ID
 service-ids, services     List all service IDs.
 service-policies          List all service policies of specified service
 service-policy            Display details of a service policy
 service-policy-create     Create a service policy
 service-policy-delete     Delete a service policy
 service-policy-update     Update a service policy
 team-add-groups           Add groups to a team with the defined role
 team-add-service-ids      Add service ID(s) to a team
 team-add-users            Add users to a team with the defined role
 team-create               Create a team
 team-delete               Delete a team
 team-get                  View users and groups for a team
 team-remove-groups        Remove groups from a team
 team-remove-service-ids   Remove service ID(s) from a team
 team-remove-users         Remove users from a team
 teams                     List all teams
 user-import               Import a user from an LDAP connection
 user-remove               Remove one or more users
 users                     List all imported users
 help

Insira 'cloudctl iam help [command]' para obter mais informações sobre um comando.
```

- funções cloudctl iam

```
Saída
------
Obtendo funções definidas pelo sistema como administrador...
OK

Name                  ID                                                    Description
Viewer                crn:v1:icp:private:iam::::role:Viewer                 Viewers can
take actions that do not change state (i.e. read only).
Administrador de cluster crn:v1:icp:private:iam::::role:ClusterAdministrator   Os
Administradores de cluster podem tomar todas as ações, incluindo a capacidade de
```

```
gerenciar o controle de acesso.
Administrador          crn:v1:icp:private:iam::::role:Administrator          Os
Administradores podem tomar todas as ações, incluindo a capacidade para gerenciar o
controle de acesso.
Editor                 crn:v1:icp:private:iam::::role:Editor                 Os Editores
podem tomar as ações que podem modificar o estado e criar/excluir sub-recursos.
Operador               crn:v1:icp:private:iam::::role:Operator               Os
Operadores podem tomar as ações necessárias para configurar e operar recursos.
```

- serviços do Cloudctl iam

```
Saída
------
Obtendo serviços definidos pelo sistema como administrador...
OK

ID                          Name                          Display Name
Supported Roles
5adf7987e6ace7000a023556    idmgmt                        service-identity-manager
ClusterAdministrator, Operator, Editor, Viewer, Administrator
5adf7987e6ace7000a023557    idprovider                    service-identity-provider
ClusterAdministrator, Operator, Editor, Viewer, Administrator
5adf7987e6ace7000a023558    idauth                        service-auth-service
ClusterAdministrator, Operator, Editor, Viewer, Administrator
5adf7987e6ace7000a023559    identity                      service-identity
ClusterAdministrator, Operator, Editor, Viewer, Administrator
5adf79e1fc55aa00c8e05bf1    helm-api-service              helmapi-repos
ClusterAdministrator, Administrator, Operator, Editor, Viewer
5adf79e7fc55aa00c8e05bf2    elasticsearch-service         elasticsearch
ClusterAdministrator, Administrator, Operator
5adf79e8fc55aa00c8e05bf3    service-monitoring-service    service-monitoring
ClusterAdministrator, Administrator, Operator
```

3. Crie o ID de serviço para um serviço, execute o seguinte comando, em que `NAME` é `<meteringserviceId>` e `[-d, --description DESCRIPTION]` é `<service id for metering>`:

```
 cloudctl iam service-id-create <meteringserviceId> -d <service id for metering>

 Saída
 ------
 Criando o ID de serviço meteringserviceId ligado à conta atual como administrador...
OK
O ID de serviço meteringserviceId foi criado com sucesso

 Nome          meteringserviceId
Descrição     service id for metering
CRN           crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-58451b31-607b-42b4-
99c8-1ceeea96bb48
Ligado a      crn:v1:icp:private:k8::n/kube-system:::
```

É possível gerenciar IDs de serviço, criar e gerenciar chaves API e criar e gerenciar políticas para acesso a serviços específicos que são necessários por um aplicativo. Veja Comandos da chave API do serviço da CLI do IBM Cloud Private para mais comandos e exemplos de serviço, chave API e política.

Criando um ID de serviço usando o console de gerenciamento do IBM Cloud Private

É possível criar um ID de serviço que forneça aos usuários permissões de função específicas para um serviço identificado em seu cluster.

Conclua as etapas a seguir para criar um ID de serviço:

1. Efetue login no console da web do IBM Cloud Private de seu cluster com um ID que possua acesso de administrador de cluster.
2. No menu de navegação, selecione **Gerenciar** > **Identidade e Acesso**.
3. Selecione **IDs de Serviço** .
4. Selecione **Criar um ID de Serviço** .

5. Insira um nome e uma descrição para seu ID de serviço. O nome deve ser uma única sequência contendo apenas letras , números, sublinhados (_) e hífens (-).
6. O tipo de ligação é para um namespace.
7. Selecione um namespace existente na lista. A seleção do namespace define o escopo do ID de serviço.
8. Selecione **Criar** para criar o ID do serviço.
9. Ligue uma política de acesso ao ID de serviço. Deve-se ter uma política de acesso associada para identificar quais funções são afetadas pelo ID de serviço. Conclua as etapas a seguir:
    1. Navegue para **Gerenciar** > **Identidade e Acesso** > **IDs de Serviço**, se você ainda não estiver nessa tela.
    2. Selecione o nome do ID de serviço que você deseja atualizar.
    3. Selecione a guia **Access Policies** . Uma lista de políticas de acesso que já estão associadas a esse ID de serviço é exibida.
    4. Selecione **Criar política de acesso** para criar a política de acesso.
    5. Selecione a função para a qual você está fornecendo as permissões.
    6. Selecione o tipo de serviço a ser gerenciado por esta política. As 3 etapas a seguir são opcionais e restringem o escopo de onde o ID de serviço possui permissões.
    7. Especifique uma instância do tipo de serviço selecionado para controlar o limite de acesso a essa instância.
    8. Insira o tipo de recurso e o identificador de recurso da instância especificada para restringir ainda mais o escopo dessa instância.
    9. Selecione **Incluir** para associar a política de acesso ao ID de serviço.
10. Crie uma chave API para o ID de serviço. Usando a chave API, a chamada é identificada como proveniente deste ID de serviço. Conclua as etapas a seguir:
    1. À medida que você visualiza os detalhes do ID de serviço, selecione a guia **Chaves de API**.
    2. Selecione **Criar chave API** para obter uma chave designada ao ID de serviço.
    3. Insira o *Nome e a descrição* para sua chave de API. Isso ajuda a identificá-la ao fazer download dela.
    4. Selecione **Criar** para fazer download da chave API. A chave é transferida por download como um arquivo `.json` para seu local padrão.
       Lembre-se: não é possível visualizar a chave API depois de sair dessa tela.
11. Clique em **Incluir equipes** e selecione a equipe que você deseja ligar ao ID de serviço. Você deve ligar uma equipe ao ID de serviço para identificar as funções que são afetadas pelo ID de serviço.
12. É possível remover um ID de serviço selecionando o ícone *Abrir e fechar a lista de opções* (...) para o ID de serviço e selecionar **Remover**.

Nota: se você deseja criar um ID de serviço usando a interface da linha de comandos, consulte Criando um ID de serviço usando a CLI do IBM Cloud Private.

Incluindo o RBAC em suas equipes para IDs de serviço

É possível implementar permissões de função específicas nas suas equipes para IDs de serviço.

Antes de implementar o RBAC, conclua as etapas a seguir:

- Configure sua conexão LDAP. Para obter mais informações, consulte Configurando a conexão LDAP.
- Crie uma equipe e inclua usuários. Para obter mais detalhes para incluir usuários em sua equipe, consulte Incluir usuários em uma equipe.
- Ligue sua equipe a um ID de serviço. Para obter mais detalhes, consulte Ligando um ID de serviço a uma equipe

Funções específicas são incluídas no ID do serviço a partir da política de acesso. Para obter mais detalhes sobre como criar uma política de acesso, consulte Criar uma política de acesso para um ID de serviço.

Inclua um ID de serviço em sua equipe para que administradores e operadores possam gerenciar o ID de serviço.

1. No menu de navegação, clique em **Gerenciar** > **Identidade & Acesso** > **Equipes**.

   Nota: sua equipe deve ter acesso ao mesmo namespace que é ligado pelo seu ID de serviço.

2. Clique na guia **IDs de serviço**.
3. Clique em **Incluir IDs de serviço** e selecione o ID de serviço na lista.

   Nota: a lista é limpa se não houver IDs de serviço ligados a um namespace que esteja designado à equipe

Ativar Istio com o IBM Cloud Private

Istio é uma plataforma aberta que pode ser usada para conectar, proteger, controlar e observar microsserviços. Com o Istio, é possível criar uma rede de serviços implementados que incluem o balanceamento de carga, a autenticação de serviço para serviço, o monitoramento e muito mais, sem mudar o código de serviço.

Limitação: o Istio não suporta Federal Information Processing Standards (FIPS). Para obter mais informações, consulte Criptografia FIPS 140-2 usando Istio. O Istio fica desativado por padrão no instalador do IBM Cloud Private Cloud Foundry.

Para incluir o suporte do Istio aos serviços, deve-se implementar um proxy sidecar especial em todo seu ambiente, que intercepta toda a comunicação de rede entre microsserviços, configurada e gerenciada, usando a funcionalidade do plano de controle fornecida no Istio.

O IBM Cloud Private versão 3.2.0 suporta dois métodos para ativar o Istio. É possível optar por ativar o Istio durante a instalação do cluster ou instalar o gráfico do Istio a partir do Catalog após a instalação do cluster. O Istio suporta totalmente as plataformas Linux®, Linux® on Power® (ppc64le) e Linux® on IBM® Z and LinuxONE.

Ativando o Istio durante a instalação do cluster

Nota: deve-se ter um mínimo de 8 núcleos em seu nó de gerenciamento.

Para ativar o Istio, mude o valor do parâmetro `istio` para `enabled` na lista de serviços de gerenciamento no arquivo `config.yaml`. Você pode configurar o parâmetro conforme ele é exibido no exemplo a seguir:

```
management_services:
  istio: enabled
  vulnerability-advisor: disabled
  storage-glusterfs: disabled
  storage-minio: disabled
```

É possível instalar o IBM Cloud Private. O Istio é instalado durante a instalação do cluster do IBM Cloud Private.

Ativando Kiali, Grafana e Prometheus durante a instalação do cluster

Para ativar Kiali e Grafana, primeiro ative o Istio. Em seguida, inclua a seguinte parte de código no arquivo `config.yaml`:

```
istio:
  kiali:
    enabled: true
  grafana:
    enabled: true
  prometheus:
    enabled: true
```

Instalando o Istio para um cluster existente

Nota: o cluster do IBM Cloud Private 3.2.0 suporta o gráfico `ibm-istio` versões 1.0.x e 1.1.x. Os gráficos `ibm-istio` 1.1.x agora estão disponíveis no repositório `ibm-charts`: `https://raw.githubusercontent.com/IBM/charts/master/repo/stable/`.

É possível implementar o Istio se você já tiver um cluster do IBM Cloud Private 3.2.0 instalado. Para instalar por meio da console de gerenciamento do IBM Cloud Private, clique em **Catálogo** e procure pelo gráfico `ibm-istio`.

1. Se você estiver ativando o Grafana com o modo de segurança, crie o segredo primeiro, seguindo o procedimento:
    1. Codifique o nome do usuário executando o comando a seguir, é possível mudar o nome do usuário:

       ```
       echo -n 'admin' | base64 YWRtaW4=
       ```

    2. Codifique a passphrase executando o comando a seguir, que também pode ser mudada:

       ```
       echo -n 'admin' | base64 YWRtaW4=
       ```

    3. Configure o namespace no qual o Istio está instalado executando o comando a seguir:

       ```
       NAMESPACE=istio-system
       ```

    4. Crie um segredo para o Grafana executando o comando a seguir:

       ```
       cat <<EOF | kubectl apply -f -
       apiVersion: v1
       kind: Secret
       metadata:
       name: grafana
       namespace: $NAMESPACE
       labels:
       app: grafana
       type: Opaque
       data:
       username: YWRtaW4=
       passphrase: YWRtaW4=
       EOF
       ```

2. Se você estiver ativando `kiali`, também será necessário criar o segredo que contém o nome do usuário e a passphrase para o painel Kiali. Execute os comandos a seguir:

   ```
   echo -n 'admin' | base64 YWRtaW4=

   echo -n 'admin' | base64 YWRtaW4=

   NAMESPACE=istio-system
   cat <<EOF | kubectl apply -f -
   apiVersion: v1
   kind: Secret
   metadata:
     name: kiali
     namespace: $NAMESPACE
     labels:
       app: kiali
   type: Opaque
   data:
     username: YWRtaW4=
     passphrase: YWRtaW4=
   EOF
   ```

3. Efetue login no IBM Cloud Private console de gerenciamento. Para instalar por meio do console de gerenciamento do IBM Cloud Private, clique em **Menu** > **Catálogo**.

4. É possível procurar por `Istio` na barra de Procura. Também é possível localizar o Istio por meio do Filtro ou de Categorias (categoria Operação). Após a procura ser concluída, o gráfico `ibm-istio` é exibido.

Nota: o gráfico `ibm-istio` está localizado em vários repositórios Helm. Se quiser instalar o gráfico `ibm-istio` 1.0.x, é possível escolhê-lo entre vários repositórios Helm. Entretanto, se você quiser implementar o gráfico `ibm-istio` 1.1.x, ele só estará disponível no repositório `ibm-charts`.

1. Clique no `ibm-istio` gráfico. Um arquivo leia-me exibe informações sobre a instalação, a desinstalação, a configuração e outros detalhes do gráfico para o Istio.

2. Clique em **Configurar** para navegar para a página de configuração.

3. Nomeie sua liberação do Helm e selecione o namespace `istio-system` no menu. O nome deve consistir em caracteres alfanuméricos minúsculos ou caracteres de traço (-) e deve iniciar e terminar com um caractere alfanumérico.

4. Certifique-se de ler e concordar com o contrato de licença.

5. **Opcional:** customize os campos *Todos os parâmetros* para sua preferência.

6. Clique em **Instalar** para implementar o gráfico do Istio e criar uma liberação do Istio.

Ativando o rastreio para um cluster existente

Para ativar o rastreio em seu cluster existente, execute os seguintes comandos:

1. Instale a CLI do Helm. Para obter mais informações, consulte Instalando a CLI do Helm (helm).

2. Obtenha os valores existentes do arquivo `values.yaml`.

```
helm get values istio --tls > istio-old-values.yaml
```

3. Faça upgrade do gráfico `istio`.

```
helm upgrade istio <path-to-the-istio-chart> --namespace istio-system --force -f istio-old-
values.yaml --set tracing.enabled=true --tls
```

Verificando a instalação

Após a instalação ser concluída, verifique se todos os componentes que você ativou para o `control plane` do Istio estão criados e em execução:

1. Assegure-se de que os serviços sejam implementados, executando o comando a seguir para obter uma lista de serviços:

```
kubectl -n istio-system get svc
```

Nota: os serviços do Kubernetes a seguir são obrigatórios: `istio-pilot`, `istio-ingressgateway`, `istio-egressgateway`, `istio-policy`, `istio-telemetry`, `istio-citadel`, `istio-statsd-prom-bridge`, `istio-galley` e, opcionalmente, `istio-sidecar-injector`, `prometheus`, `grafana`, `jaeger-*`, `kiali*`, `servicegraph`, `tracing`, `zipkin`.

A saída pode ser semelhante ao conteúdo a seguir:

```
NAME                       TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)
AGE
grafana                    ClusterIP      10.0.0.135   <none>        3000/TCP
37m
istio-citadel              ClusterIP      10.0.0.167   <none>        8060/TCP,9093/TCP
37m
istio-egressgateway        ClusterIP      10.0.0.79    <none>        80/TCP,443/TCP
37m
istio-galley               ClusterIP      10.0.0.70    <none>        443/TCP,9093/TCP
37m
istio-ingressgateway       LoadBalancer   10.0.0.233   <pending>
80:31380/TCP,443:31390/TCP,31400:31400/TCP,15011:30692/TCP,8060:32603/TCP,15030:31295/TCP,15031
:31856/TCP   37m
istio-pilot                ClusterIP      10.0.0.148   <none>
15010/TCP,15011/TCP,8080/TCP,9093/TCP
37m
istio-policy               ClusterIP      10.0.0.89    <none>
9091/TCP,15004/TCP,9093/TCP
37m
istio-sidecar-injector     ClusterIP      10.0.0.199   <none>        443/TCP
37m
istio-statsd-prom-bridge   ClusterIP      10.0.0.198   <none>        9102/TCP,9125/UDP
37m
istio-telemetry            ClusterIP      10.0.0.140   <none>
9091/TCP,15004/TCP,9093/TCP,42422/TCP
37m
jaeger-agent               ClusterIP      None         <none>        5775/UDP,6831/UDP,6832/UDP
37m
jaeger-collector           ClusterIP      10.0.0.102   <none>        14267/TCP,14268/TCP
37m
jaeger-query               ClusterIP      10.0.0.118   <none>        16686/TCP
37m
kiali                      ClusterIP      10.0.0.177   <none>        20001/TCP
37m
kiali-jaeger               NodePort       10.0.0.65    <none>        20002:32439/TCP
37m
prometheus                 ClusterIP      10.0.0.200   <none>        9090/TCP
37m
servicegraph               ClusterIP      10.0.0.197   <none>        8088/TCP
```

```
37m
tracing                    ClusterIP      10.0.0.99    <none>        16686/TCP
37m
zipkin                     ClusterIP      10.0.0.134   <none>        9411/TCP
```

2. Assegure-se de que os pods do Kubernetes correspondentes estejam implementados e que todos os contêineres estejam ativos. Execute o comando a seguir:

   ```
   kubectl -n istio-system get pods
   ```

   Nota: os pods a seguir são obrigatórios: `istio-pilot-*`, `istio-ingressgateway-*`, `istio-egressgateway-*`, `istio-policy-*`, `istio-telemetry-*`, `istio-citadel-*`, `istio-statsd-prom-bridge-*`, `istio-galley-*` e, opcionalmente, `istio-sidecar-injector-*`, `prometheus-*`, `grafana-*`, `istio-tracing-*`, `kiali*`, `servicegraph-*`.

   A saída pode ser semelhante ao conteúdo a seguir:

```
NAME                                     READY     STATUS    RESTARTS   AGE
grafana-75f4f8dcf7-2p92z                 1/1       Running   0          37m
istio-citadel-5d5d5bcd5-tmv2w            1/1       Running   0          37m
istio-egressgateway-6669b4888d-t8fqs     1/1       Running   0          37m
istio-galley-d6d995d66-d6tb8             1/1       Running   0          37m
istio-ingressgateway-57bf47dc7c-ntz8h    1/1       Running   0          37m
istio-pilot-745899bb46-kf4z4             2/2       Running   0          37m
istio-policy-57567ff748-96vvb            2/2       Running   0          37m
istio-sidecar-injector-76fc499f9c-r57bw  1/1       Running   0          37m
istio-statsd-prom-bridge-676dcc4f8b-d7fhc 1/1      Running   0          37m
istio-telemetry-6fc9f55c4f-4229b         2/2       Running   0          37m
istio-tracing-66f4676d88-wjgzr           1/1       Running   0          37m
kiali-7bdd48bd7d-b6vwd                   1/1       Running   0          37m
prometheus-b8446f488-fpprf               1/1       Running   0          37m
servicegraph-fcdc4c44d-tdm2z             1/1       Running   0          37m
```

Implementando os Aplicativos

Após o `control plane` do IStio ser implementado com sucesso, é possível iniciar a implementação de seus aplicativos.

Criando *imagePullSecrets* para o registro privado do

Docker do IBM Cloud Private

Se você implementar o Istio durante a instalação do cluster, deverá criar `imagePullSecrets` para o registro privado do Docker do IBM Cloud Private: `<cluster_hostname>:<registry_port>` (o valor padrão é `mycluster.icp:8500`), no namespace em que seus aplicativos estão implementados, seus aplicativos podem obter as imagens de sidecar do registro privado do Docker.

1. Crie um `secret` que seja denominado `infra-registry-key` em seu registro do Docker do IBM Cloud Private que contenha seu token de autorização. Execute o comando a seguir:

   ```
   kubectl -n <application_namespace> create secret docker-registry infra-registry-key --docker-
   server=<cluster_hostname>:<registry_port> --docker-username=<your-name> --docker-password=
   <your-password> --docker-email=<your-email>
   ```

2. Corrija seu segredo para o `ServiceAccount` que está associado aos seus aplicativos. Execute o comando a seguir:

   ```
   kubectl get serviceaccount <your-service-account-name> -o yaml | grep -w infra-registry-key ||
   kubectl patch serviceaccount <your-service-account-name> -p '{"imagePullSecrets": [{"name":
   "infra-registry-key"}]}'
   ```

Injeção automática de sidecar

Se você ativou a injeção de sidecar automática, o `istio-sidecar-injector` injetará automaticamente contêineres do Envoy em seus pods de aplicativo que são executados em namespaces que são rotulados com `istio-injection=enabled`.

Para injetar contêineres do Envoy automaticamente, conclua as etapas a seguir:

1. Rotule seu namespace como `istio-injection=enabled`. Execute o comando a seguir:

   ```
   kubectl label namespace <namespace> istio-injection=enabled
   ```

2. Execute o comando a seguir para criar seu namespace em seu arquivo `.yaml`:

```
kubectl create -n <namespace> -f <your-app-spec>.yaml
```

Injeção manual de sidecar

Se você não ativou a injeção automática de sidecar, será possível injetar o contêiner do Envoy manualmente.

Para ativar manualmente a injeção sidecar, deve-se usar o `istioctl`. Implemente seus aplicativos com sua injeção de sidecar manualmente. Execute o comando a seguir:

```
kubectl create -f <(istioctl kube-inject -f <your-app-spec>.yaml)
```

Coletando e visualizando

Coletar extensões de rastreio usando Jaeger

Por padrão, o Istio ativa `Jaeger` com um tipo de serviço de `ClusterIP`. Durante a instalação, é possível mudar o tipo de serviço padrão para `NodePort` para que seja possível acessar o `Jaeger` por meio de um ambiente externo.

Para visualizar outros tipos de serviço do `NodePort` que possuem acesso ao `Jaeger`, execute os comandos a seguir:

```
  kubectl expose service jaeger-query --type=NodePort --name=<jaeger-query-svc> --namespace istio-
system

  export JAEGER_URL=$(kubectl get po -l app=jaeger -n istio-system -o 'jsonpath=
{.items[0].status.hostIP}'):$(kubectl get svc <jaeger-query-svc> -n istio-system -o 'jsonpath=
{.spec.ports[0].nodePort}')

  echo http://${JAEGER_URL}/
```

É possível acessar `http://${JAEGER_URL}/` a partir de seu navegador para visualizar extensões de rastreio.

Coletando Métricas Usando Prometheus

Semelhante à seção *Coletando extensões de rastreio usando o Jaeger*, se você instalar o Istio com o `prometheus` ativado, haverá um serviço `prometheus` com um tipo de `ClusterIP` por padrão. É possível mudar o tipo de serviço padrão para `NodePort`.

Para visualizar outros tipos de serviço de `NodePort` que tenham acesso ao `prometheus` a partir de um ambiente externo, execute os comandos a seguir:

```
  kubectl expose service prometheus --type=NodePort --name=<prometheus-svc> --namespace istio-system

  export PROMETHEUS_URL=$(kubectl get po -l app=prometheus -n istio-system -o 'jsonpath=
{.items[0].status.hostIP}'):$(kubectl get svc <prometheus-svc> -n istio-system -o 'jsonpath=
{.spec.ports[0].nodePort}')

  echo http://${PROMETHEUS_URL}
```

É possível acessar o `http://${PROMETHEUS_URL}/` de seu navegador para verificar se as métricas estão sendo coletadas no Prometheus.

Visualizando métricas com Grafana

Semelhante aos serviços `Jaeger` e `Prometheus`, se você instalar o Istio com o `grafana` ativado, haverá um serviço `grafana` com um tipo de `ClusterIP` por padrão. É possível mudar o tipo de serviço padrão para `NodePort`.

Para visualizar outros tipos de serviço de `NodePort` que tenham acesso ao `grafana` a partir de um ambiente externo, execute os comandos a seguir:

```
  kubectl expose service grafana --type=NodePort --name=<grafana-svc> --namespace istio-system

  export GRAFANA_URL=$(kubectl get po -l app=grafana -n istio-system -o 'jsonpath=
{.items[0].status.hostIP}'):$(kubectl get svc <grafana-svc> -n istio-system -o 'jsonpath=
{.spec.ports[0].nodePort}')

  echo http://${GRAFANA_URL}/
```

É possível acessar o `http://${GRAFANA_URL}/` de seu navegador para visualizar a página da web Grafana.

Observar os microsserviços com o Kiali

Assim como os serviços `Jaeger`, `Prometheus` e `Grafana`, se você instalar o Istio com o `kiali` ativado, haverá um serviço `kiali` com um tipo de `ClusterIP` por padrão. É possível mudar o tipo de serviço padrão para `NodePort`.

Para visualizar outros tipos de serviço de `NodePort` que tenham acesso ao `kiali` a partir de um ambiente externo, execute os comandos a seguir:

```
  kubectl expose service kiali --type=NodePort --name=<kiali-svc> --namespace istio-system

  export KIALI_URL=$(kubectl get po -l app=kiali -n istio-system -o 'jsonpath=
{.items[0].status.hostIP}'):$(kubectl get svc <kiali-svc> -n istio-system -o 'jsonpath=
{.spec.ports[0].nodePort}')

  echo http://${KIALI_URL}/
```

É possível acessar o `http://${KIALI_URL}/` a partir de seu navegador para visualizar o painel do Kiali.

Para obter mais informações sobre o Istio, consulte os Docs do Istio.

Restrições

Istio no Linux® on Power® (ppc64le)

Com o Istio, é possível criar seus próprios filtros, cujos tipos são filtros HTTP. Ao executar o Istio no Linux® on Power® (ppc64le), os filtros HTTP Lua não são suportados. Os filtros usam o compilador LuaJIT, que não possui suporte little-endian de 64 bits para o Linux® on Power® (ppc64le). No momento não há suporte de filtro HTTP Lua para o Linux® on Power® (ppc64le).

Para obter mais informações sobre como criar seus próprios filtros usando Lua ou outras extensões, consulte a documentação do Envoy para sua liberação específica.

Se você estiver implementando aplicativos com injeção de Istio para namespaces não padrão, deve-se criar uma `ClusterRoleBinding` extra para conceder permissões `privilegiadas` para contas de serviço nesse namespace.

Por exemplo, para implementar um aplicativo para o namespace não padrão, `istio-lab`, edite o seu YAML. Seu YAML pode ser semelhante ao conteúdo a seguir:

```
export APPLICATION_NAMESPACE=istio-lab
cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: istio-privileged-users
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
  - apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts:${APPLICATION_NAMESPACE}
EOF
```

Em seguida, você está pronto para implementar seus aplicativos com injeção manual ou injeção automática.

Serviço Catalog

Com o Service Catalog, é possível usar serviços dos brokers e configurar seus aplicativos para usar os serviços. O Service Catalog fornece uma interface do Kubernetes para uma ou mais APIs do Open Service Broker que são compatíveis com os brokers de serviço.

IBM® Cloud Private 3.2.0 inclui a versão beta v0.1.40 do Catalog de serviço. Para obter mais informações, consulte Documentos do Catalog de serviço.

Para obter informações sobre as mudanças que foram incluídas com a v0.1.40, consulte v0.1.40.

- Gerenciando os recursos de Catalog Serviço
- Gerenciando um recurso de broker de serviço a partir da interface da linha de comandos (CLI)
- Gerenciando um recurso de broker de serviço a partir da console de gerenciamento

Gerenciando recursos do Catalog Serviço

É possível gerenciar o Service Catalog a partir da interface da linha de comandos (CLI) e da console de gerenciamento. O componente Service Catalog inclui cinco recursos do Kubernetes que você pode usar.

Detalhes do Catalog recurso de serviço

Recurso	Detalhes do recurso
• ClusterServiceBroker • ServiceBroker	Gerencia um conjunto de um ou mais serviços
• ClusterServiceClass • ServiceClass	Descreve o serviço e os planos associados que o broker oferece para cada serviço
• ClusterServicePlan • ServicePlan	Descreve os planos que são oferecidos para o serviço
• ServiceInstance	Cria uma instância de um serviço gerenciado que está disponível para uso por um ou mais

aplicativos em cluster | |

- ServiceBinding

| ServiceBinding refere-se a e usa um ServiceInstance. Ele também cria um segredo do Kubernetes que contém os detalhes da conexão e as credenciais para o serviço que é representado pelo ServiceInstance |

Gerenciando os recursos do Service Catalog com a interface da linha de comandos

É possível usar os oito recursos da interface da linha de comandos do `kubectl`. A tabela a seguir descreve o escopo e o uso dos recursos:

Tabela 1. Recursos do Kubernetes para o catálogo de serviços

Recurso Kubernetes	Escopo	Uso de Exemplo
ClusterServiceBrokers	cluster	`kubectl get clusterservicebrokers`
ClusterServiceClasses	cluster	`kubectl get clusterserviceclasses`
ClusterServicePlans	cluster	`kubectl get clusterserviceplans`
ServiceBrokers	namespace	`kubectl get servicebrokers -n <namespace>`
ServiceClasses	namespace	`kubectl get serviceclasses -n <namespace>`
ServicePlans	namespace	`kubectl get serviceplans -n <namespace>`
ServiceInstances	namespace	`kubectl get serviceinstances -n <namespace>`
ServiceBindings	namespace	`kubectl get servicebindings -n <namespace>`

É possível estender os recursos de seu cluster do IBM Cloud Private integrando-se com os serviços que são implementados em seu cluster do IBM Cloud Private ou cluster externo. Consulte Criando um recurso de broker de serviço a partir da interface da linha de comandos para obter mais detalhes.

Gerenciando os recursos do Catalog Serviço a partir do IBM Cloud Private console de gerenciamento

É possível carregar e implementar brokers como gráficos Helm a partir do IBM Cloud Private Catalog na console de gerenciamento. Os gráficos incluem o modelo de criação de registro e de segredo autorizado. Os Gráficos Helm podem incluir os modelos para implementação de um broker, se ele precisar ser implementado dentro de um cluster.

Os serviços do broker que estão disponíveis para implementação são listados na visualização do Catalog juntamente com os gráficos Helm. Os serviços podem ser identificados com o rótulo `serviço`.

Depois que um broker de serviço é registrado com seu sistema operacional, é possível selecionar os serviços e os planos internos de cada serviço no Catalog.

O IBM Cloud Private suporta a página de detalhes do serviço. A página de detalhes do serviço contém uma visão geral do serviço e lista os planos disponíveis, o processo de implementação, a criação de uma instância do serviço e a opção de ligação de serviço.

Nota: sob o controle de acesso baseado na função *(RBAC)*, somente o administrador de cluster tem permissão para implementar e registrar gráficos do broker e obter todos os ClusterServiceClasses e ClusterServicePlans. Os recursos ClusterServiceClasses e ClusterServicePlans são designados à equipe.

Consulte Criando um recurso de broker de serviço a partir da console de gerenciamento para obter mais detalhes.

Gerenciando um recurso de broker de serviço a partir da interface da linha de comandos (CLI)

É possível implementar um broker de serviço para visualizar os serviços e planos disponíveis, criar uma instância dos serviços e planos disponíveis e criar ligações para se conectar à instância de serviço.

É possível implementar um broker de serviço de amostra da comunidade para testar a função ou instalar seu próprio broker de serviço que segue a especificação da API Open Service Broker para seus próprios requisitos.

É possível usar o broker de serviço de amostra `ups_broker` do Kubernetes como uma maneira fácil de examinar a função do broker de serviço. É possível implementá-lo em seu cluster como um gráfico do Helm concluindo as etapas a seguir:

1. Opcionalmente, é possível criar um novo repositório do Helm denominado `charts`, no qual é possível fazer download do broker de serviço. Consulte Incluindo um repositório do Helm para o procedimento.
2. Se você criou um novo repositório, clone o repositório para fazer download do broker de serviço para o novo repositório do Helm.
3. Se você optou por não criar um novo repositório, é possível fazer download do gráfico do broker de serviço para um repositório do Helm existente. Se você não deseja usar um repositório do Helm, é possível fazer download do gráfico e usar o comando `helm install` para instalar o gráfico no cluster. Consulte Instalando a CLI Helm (helm) para obter mais informações sobre o comando `helm install`.
4. Execute o comando Helm a seguir na CLI do Helm para implementar o broker de serviço:

   ```
   helm install charts/ups-broker --name ups-broker --namespace ups-broker
   ```

5. Registrando um broker de serviço para o IBM Cloud Private Catalog
6. Visualizando ServiceClasses e ServicePlans
7. Criando um ServiceInstance
8. Ligando o ServiceInstance a um aplicativo
9. Desvinculando um ServiceInstance
10. Excluindo um ServiceInstance
11. Excluindo um ServiceBroker

Registrando um broker de serviço para o Catalog de serviço do IBM Cloud Private

É possível registrar um broker para um serviço em seu cluster ou namespace. Para poder visualizar os serviços e planos dos quais criar uma instância, deve-se registrar o broker no Catalog de serviço com o recurso **ServiceBroker**.

Registrando um broker de serviço em nível de cluster

Registre um broker de serviço em nível de cluster para o Catalog de serviço. Conclua as etapas a seguir:

1. Crie um recurso ServiceBroker criando um arquivo `broker.yaml`, que contém os detalhes do broker. Deve-se especificar valores para o `name` do broker e a `url` para o broker. Seu arquivo `broker.yaml` pode ser semelhante ao conteúdo a seguir:

   ```
   apiVersion: servicecatalog.k8s.io/v1beta1
   kind: ClusterServiceBroker
   metadata:
     name: ups-broker
   ```

```
spec:
  url: http://ups-broker-ups-broker.ups-broker.svc.cluster.local
```

1. Execute o comando a seguir para criar seu recurso ServiceBroker em nível de cluster:

   ```
   kubectl create -f broker.yaml
   ```

Um broker em nível de cluster é criado.

Registrando um broker de serviço em nível de namespace

Conclua as etapas a seguir para registrar um broker de serviço em nível de namespace para o Catalog de serviço:

1. Crie um recurso ServiceBroker criando um arquivo `ns-broker.yaml`, que contém os detalhes do broker. Deve-se especificar valores para o `nome`, o `namespace` e a `url` do broker. O arquivo `ns-broker.yaml` pode ser semelhante ao conteúdo a seguir:

   ```
   apiVersion: servicecatalog.k8s.io/v1beta1
   kind: ServiceBroker
   metadata:
     name: ups-broker
     namespace: ns-broker
   spec:
     url: http://ups-broker-ups-broker.default.svc.cluster.local
   ```

2. Execute o comando a seguir para criar seu recurso ServiceBroker em nível de namespace:

   ```
   kubectl create -f ns-broker.yaml
   ```

Um broker em nível de namespace é criado.

Visualizando ServiceClasses e ServicePlans

Cada serviço tem um recurso ServiceClass correspondente e pode ter um ou mais planos que estão associados a ele.

- Para exibir as classes de serviço registradas, execute o comando a seguir:
  - Exiba as classes de serviço registradas para os brokers em nível de cluster:

    ```
    kubectl get clusterserviceclasses
    ```

  - Exiba as classes de serviço registradas para os brokers em nível de namespace:

    ```
    kubectl get serviceclasses -n ns-broker
    ```

- Para exibir os planos de serviços registrados, execute o comando a seguir:
  - Exiba os planos de serviços registrados para os brokers em nível de cluster:

    ```
    kubectl get clusterserviceplans
    ```

  - Exiba os planos de serviços registrados para os brokers em nível de namespace:

    ```
    kubectl get serviceplans -n ns-broker
    ```

Criando um ServiceInstance

Crie um recurso ServiceInstance criando um arquivo `instance.yaml` para um broker de serviço. Deve-se criar uma instância de uma classe de serviço antes que seus aplicativos possam usá-la. Conclua as etapas a seguir:

Criando um ServiceInstance para seu broker em nível de cluster

1. Crie um recurso ServiceInstance para seu broker em nível de cluster no arquivo `instance.yaml`. Seu arquivo `instance.yaml` pode ser semelhante ao conteúdo a seguir:

   ```
      apiVersion: servicecatalog.k8s.io/v1beta1
   kind: ServiceInstance
   metadata:
    name: ups-instance
    namespace: test-ns
   spec:
   ```

```
clusterServiceClassExternalName: user-provided-service
clusterServicePlanExternalName: default
parameters:
  credentials:
    name: root
    password: letmein
```

2. Crie a instância executando o comando a seguir:

```
kubectl create -f instance.yaml
```

Uma instância de serviço é criada para seu broker de serviço de cluster.

Criando um ServiceInstance para seu broker em nível de namespace

1. Crie um recurso ServiceInstance para seu broker em nível de namespace no arquivo `instance.yaml`. Seu arquivo `instance.yaml` pode ser semelhante ao conteúdo a seguir:

```
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceInstance
metadata:
  name: ups-ns-instance
  namespace: ns-broker
spec:
  serviceClassExternalName: user-provided-service
  servicePlanExternalName: default
  parameters:
    credentials:
      name: root
      password: letmein
```

2. Crie a instância executando o comando a seguir:

```
kubectl create -f ns-instance.yaml
```

Uma instância de serviço é criada para seu broker de serviço de namespace.

Visualizando o ServiceInstance

É possível visualizar seus recursos ServiceInstance a partir da interface da linha de comandos (CLI).

Conclua as etapas a seguir para visualizar seus recursos ServiceInstance:

1. Obtenha o nome do namespace no qual o recurso ServiceInstance está localizado:

```
kubectl get serviceinstance -n test-ns
```

2. Execute o comando a seguir para visualizar os detalhes do seu recurso:

```
kubectl get serviceinstance ups-instance -n test-ns -o yaml
```

Também é possível visualizar seus recursos a partir da console de gerenciamento. Para obter mais informações, consulte Gerenciando recursos do Catalog de serviço a partir da IBM Cloud Private console de gerenciamento

Ligando o ServiceInstance a um aplicativo

Para ligar o ServiceInstance a um aplicativo, conclua as etapas a seguir:

1. Crie o arquivo `binding.yaml` que liga a instância de serviço a um aplicativo. Deve-se fornecer a ligação `name` e um `secretName`. O arquivo `binding.yaml` pode ser semelhante ao conteúdo a seguir:

```
 apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceBinding
metadata:
name: <binding_name>
namespace: test-ns
spec:
secretName: <secret_name>
instanceRef:
 name: ups-instance
```

Nota: o valor para o parâmetro `namespace` do arquivo `binding.yaml` deve ser o mesmo que o valor para o namespace da instância de serviço. O YAML de exemplo de ligação de serviços é para recursos em nível de cluster, mas arquivos YAML em nível de namespace seguem o mesmo formato.

Para obter mais detalhes sobre os rótulos do seletor de pod na documentação do Kubernetes, consulte Rótulos e seletores.

2. Crie um arquivo `binding.yaml` executando o seguinte comando:

   ```
   kubectl create -f binding.yaml
   ```

   O controlador do catálogo de serviços cria um segredo do Kubernetes que contém os detalhes de conexão e as credenciais para a instância de serviço.

3. Para visualizar os detalhes de seu arquivo `binding.yaml`, execute o comando a seguir:

   ```
   kubectl get servicebindings service_binding -o yaml
   ```

4. Visualize os detalhes secretos para se conectar ao aplicativo. Execute o comando a seguir:

   ```
   kubectl get secrets <secret_name> -o yaml
   ```

   - `<secret_name>` é o nome do segredo que é usado na ligação. Seu `<secret_name>` pode ser semelhante ao seguinte conteúdo:

     ```
     apiVersion: v1
     data:
     . . Secret data to connect to the database instance . .
     kind: Secret
     metadata:
     . . metadata parameters and values . .
     type: Opaque
     ```

Desvinculando um ServiceInstance

Deve-se excluir o ServiceBinding que você criou antes de desvincular uma instância de um aplicativo. Depois de excluir uma ligação, o segredo que a ligação usa também é excluído.

Para desvincular uma instância, execute o comando a seguir:

```
kubectl delete -n test-ns servicebindings ups-binding
```

- `ups-binding` é o nome da ligação a ser removida.

Excluindo um ServiceInstance

Exclua a instância de serviço executando o comando a seguir:

```
kubectl delete -n test-ns serviceinstances ups-instance
```

- `ups-instance` é o nome da instância a ser excluída.

Excluindo um ServiceBroker

Para excluir um ClusterServiceBroker, deve-se cancelar o registro do broker. Para cancelar o registro do broker, execute o comando a seguir:

```
kubectl delete clusterservicebrokers ups-broker
```

- `ups-broker` é o nome do broker de serviço no nível do cluster.

Para excluir um NamespaceServiceBroker, deve-se cancelar o registro do broker. Para cancelar o registro do broker, execute o comando a seguir:

```
kubectl delete servicebrokers ups-broker -n ns-broker
```

- `ups-broker` é o nome do broker de serviço em nível de namespace a ser excluído.

Excluindo uma implementação do Helm

Deve-se excluir a implementação do Helm ao limpar o servidor de broker de serviço. Para excluir a implementação do Helm, execute o comando a seguir:

```
helm delete --purge ups-broker
```

- `ups-broker` é o nome da implementação do Helm a ser excluída.

  Importante: deve-se concluir o processo de exclusão na ordem a seguir: desvincule sua instância de serviço, exclua sua instância de serviço, exclua seu broker de serviço e exclua a implementação do Helm.

Gerenciando um recurso de broker de serviço a partir da console de gerenciamento

Registre um recurso de broker de serviço para criar uma instância de seus serviços e planos.

Criando um recurso de broker de serviço de cluster

Para criar um broker de serviço a partir do console de gerenciamento, conclua as seguintes etapas:

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Gerenciar** > **Brokers de serviço**.
3. Na guia *Broker de Serviço de Cluster*, clique no botão **Incluir ClusterServiceBroker** para incluir um recurso de broker de serviço de cluster.
4. Na caixa de diálogo Incluir ClusterServiceBroker, forneça os seguintes detalhes:
   - Nome: o nome do recurso de broker de serviço de cluster.
   - Url: O terminal do broker de serviço.
   - Pacote configurável de CA: o nome do pacote configurável de CA confiável para o servidor TLS.
   - insecureSkipTLSVerify: Marque a caixa de seleção insecureSkipTLSVerify para ignorar a verificação de TLS.
   - Namespace secreto: O local do namespace para o segredo.
   - Segredo: O nome do segredo associado.

   Nota: deve-se fornecer um valor para os campos *Nome* e *Url*.
5. Ao concluir a customização do recurso de broker de serviço de cluster, você pode clicar na régua de controle **Modo JSON** para visualizar o modo JSON do recurso. O recurso de broker de serviço pode ser semelhante à seguinte saída:

   ```
   {
     "kind": "ClusterServiceBroker",
     "apiVersion": "servicecatalog.k8s.io/v1beta1",
     "metadata": {
       "name": "ups-broker"
     }, "spec": {
       "url": "http://ups-broker-ups-broker.ups-broker.svc.cluster.local"
     }
   }
   ```
6. Clique em **Criar**.

Um broker de serviço de cluster é criado a partir da console de gerenciamento.

Criando um recurso de broker de serviço de namespace

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Gerenciar** > **Brokers de serviço**.
3. Na guia *Broker de Serviço de Namespace*, clique em **Incluir NamespaceServiceBroker** para incluir um recurso de broker de serviço de namespace.
4. Na caixa de diálogo Incluir NamespaceServiceBroker, forneça os seguintes detalhes:
   - Nome: o nome do recurso de broker de serviço de namespace.
   - Namespace: o namespace para o recurso de broker de serviço.
   - Url: O terminal do broker de serviço.
   - Pacote configurável de CA: o nome do pacote configurável de CA confiável para o servidor TLS.

- insecureSkipTLSVerify: Marque a caixa de seleção insecureSkipTLSVerify para ignorar a verificação de TLS.
- Namespace secreto: O local do namespace para o segredo.
- Segredo: O nome do segredo associado.

Nota: deve-se fornecer um valor para os campos *Nome, Namespace e Url*.

5. Ao concluir a customização do recurso de broker de serviço de namespace, é possível clicar na régua de controle **Modo JSON** para visualizar o modo JSON do seu recurso. O recurso de broker de serviço pode ser semelhante à seguinte saída:

```
{
  "kind": "ServiceBroker",
  "apiVersion": "servicecatalog.k8s.io/v1beta1",
  "metadata": {
    "name": "ups-ns-broker"
    "namespace": <namespace-name>
  },
  "spec": {
    "url": "http://ns-broker-ns-broker.ns-broker.svc.cluster.local"
  }
}
```

6. Clique em **Criar**.

Um recurso de broker de serviço de namespace é criado.

Criando um ServiceInstance

1. Efetue login em seu cluster do IBM Cloud Private.
2. Clique em **Catálogo** .
3. Clique no gráfico do Helm `example-service`.
4. Na seção *Planos*, selecione um plano de serviço para configurar.
5. Clique em **Configurar**.
6. Insira um valor para o campo *Nome da Instância*.
7. Selecione um namespace customizado no campo *Namespace*.
8. (Opcional) Inclua uma ligação de serviços para a instância do servidor marcando a caixa de seleção *Incluir Ligação de Serviços*.
    1. Insira um valor para o campo *Nome de Ligação* e para o campo *Nome do Segredo de Ligação*.
    2. Selecione um namespace customizado no campo *Namespace de Ligação*.**Nota**: o campo *Namespace de Ligação* fica desativado, pois o namespace de ligação deve corresponder ao namespace da instância de serviço.
9. Clique em **Instalar** .

Visualizando um Serviço Brokered

Depois de criar uma instância para um serviço e plano do broker específicos, essa instância de serviço em broker poderá ser visualizada.

1. No menu de navegação, clique em **Cargas de trabalho** > **Serviços em broker**.
2. Selecione uma instância de serviço do broker específica para visualizar o objeto de ligação que é criado para a instância de serviço.
3. (Opcional) Inclua uma ligação de serviços no serviço do broker.
    1. Clique em **Incluir Ligação de Serviço** para abrir Criar Ligação de Serviço.
    2. Deve-se inserir um valor para os seguintes campos: *Nome de Ligação, Namespace de Ligação, Nome da Instância de Serviço e Nome do Segredo de Ligação*.
    3. Ao concluir a customização do recurso de broker de serviço de namespace, é possível clicar na régua de controle **Modo JSON** para visualizar o modo JSON do seu recurso.
    4. Clique em **Criar**.

Nota: antes de excluir uma instância de serviço do broker, deve-se remover os recursos de ligação.

URL do painel

Se uma instância de serviço do broker definir uma URL do painel, será possível clicar em **Ativar**, para ativar a URL.

Para obter mais informações sobre como usar brokers de serviço, consulte Gerenciando recursos do Catalog de serviço.

Desvinculando um ServiceInstance

Deve-se excluir a ServiceBinding que você criou antes de excluir uma instância de seu aplicativo. Conclua as etapas a seguir para desvincular uma instância de serviço:

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Cargas de trabalho** > **Serviços em broker**.
3. Clique na instância de serviço que você deseja desvincular de seu aplicativo.
4. Na seção *ServiceBindings*, clique no ícone **Abrir e fechar lista de opções** para excluir a ligação de serviço.
5. Clique em **Excluir**.
6. Na caixa de diálogo *ServiceBinding*, clique em **Excluir ServiceBinding**.

Sua instância de serviço é desvinculada de seu aplicativo.

Excluindo um ServiceInstance

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Cargas de trabalho** > **Serviços em broker**.
3. Localize uma instância de serviço específica que você deseja excluir. Clique no ícone **Abrir e fechar lista de opções**.
4. Clique em **Excluir**.
5. Na caixa de diálogo *ServiceInstance*, clique em **Excluir ServiceInstance**.

Sua instância de serviço é excluída.

Excluindo um ServiceBroker

Visualize as seções a seguir para excluir um broker de serviço de nível de cluster ou um broker de serviço de nível de namespace.

Excluindo um broker de serviço em nível de cluster

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Gerenciar** > **Brokers de serviço**.
3. Na guia *Brokers do serviço de cluster*, localize um broker de serviço que você deseja excluir.
4. Clique no ícone **Abrir e fechar lista de opções**.
5. Clique em **Excluir**.
6. Na caixa de diálogo *ClusterServiceBroker*, clique em **Excluir ClusterServiceBroker**.

Excluindo um broker de serviço de nível de namespace

1. Efetue login em seu cluster do IBM Cloud Private.
2. No menu de navegação, clique em **Gerenciar** > **Brokers de serviço**.
3. Clique na guia *Brokers de serviço de namespace*.
4. Na guia *Brokers de serviço de namespace*, localize um broker de serviço que você deseja excluir.
5. Clique no ícone **Abrir e fechar lista de opções**.
6. Clique em **Excluir**.
7. Na caixa de diálogo *NamespaceServiceBroker*, clique em **Excluir NamespaceServiceBroker**.

Seu broker de serviço é excluído.

Excluindo uma implementação do Helm

Deve-se excluir a implementação do Helm ao limpar o servidor de broker de serviço. Exclua a implementação do Helm a partir da console de gerenciamento com as seguintes etapas:

1. No menu de navegação, clique em **Cargas de trabalho** > **Liberações do Helm**.
2. Procure pela liberação do Helm que você deseja excluir.
3. Clique no ícone **Abrir e fechar lista de opções**.
4. Clique em **Excluir**.
5. No modal de resposta, clique em **Excluir**.

Sua liberação do Helm é excluída.

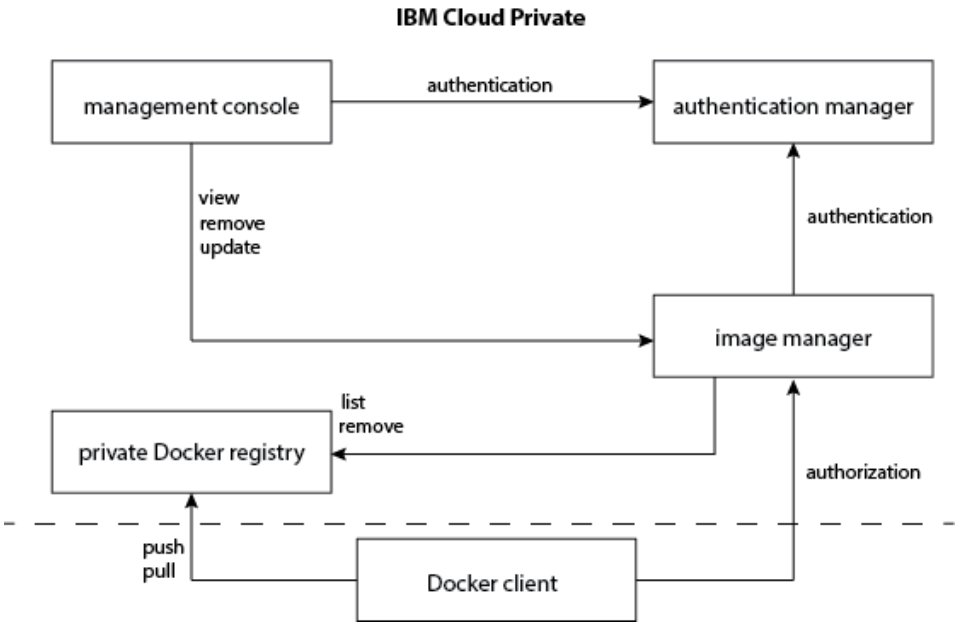
Importante: deve-se concluir o processo de exclusão na ordem a seguir: desvincule sua instância de serviço, exclua sua instância de serviço, exclua seu broker de serviço e exclua a implementação do Helm.

Gerenciando imagens

O gerenciador de imagem é um local centralizado para gerenciar imagens dentro de seu cluster. O gerenciador de imagem fornece as mesmas funções que o Docker Hub.

O gerenciador de imagens

O gerenciador de imagens é uma camada executada sobre a API do registro do Docker V2. O gerenciador de imagem fornece funções de gerenciamento e autorização para repositórios de imagem que o registro do Docker armazena.



O gerenciador de imagem integra-se ao registro do Docker para fornecer um serviço de registro local que funcione da mesma maneira que o serviço de registro baseado em nuvem no Docker Hub. O registro local, diferentemente do serviço externo Docker Hub, fornece restrições sobre quais usuários podem visualizar ou efetuar pull de imagens. Este registro local mantém restrições de push que o Docker Hub cumpre.

É possível incluir imagens do Docker no registro de imagem do cluster do IBM Cloud Private usando as operações da linha de comandos do Docker. Para obter informações adicionais sobre o Docker, consulte *Introdução, Parte 1: Orientação e configuração* na página de documentos do Docker.

É possível usar o cliente Docker para efetuar push ou pull de imagens em seu cluster. Em seguida, o gerenciador de imagem usa o serviço de autenticação do cluster para acessar as credenciais de um usuário que está com login efetuado e fornece acesso às imagens.

As imagens que são incluídas no registro de imagem pertencem aos namespaces. Todos os usuários dentro de um namespace são proprietários das imagens. Um proprietário pode remover ou atualizar as imagens da console de gerenciamento do cluster. Superadministradores têm acesso total a todas as imagens no cluster. Os proprietários também podem atualizar o escopo de uma imagem. A configuração do escopo pode restringir uma imagem a um namespace específico ou permitir que a imagem fique acessível para todos os namespaces.

A console de gerenciamento permite visualizar todas as imagens que estão disponíveis para elas. Clique no nome da imagem para visualizar informações adicionais. Depois de instalar o Vulnerability Advisor, a página de imagem exibe o status de sua varredura para todas as imagens no registro de imagem.

Para remover ou atualizar imagens que estão dentro do registro de imagem privado, deve-se usar a API de gerenciamento de imagem. Para obter mais informações, consulte API de gerenciamento de imagem.

Configurando a autenticação para a CLI do Docker

Para acessar o registro de imagem privado de fora de seu cluster do IBM® Cloud Private, configure a autenticação de seu computador para o cluster.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Antes de iniciar

1. Deve-se instalar o Docker em seu computador. Para obter mais informações, consulte Instalar o Docker.
2. Se você tiver o proxy do Docker ativado em seu nó, conclua as etapas a seguir.
    1. Inclua `<cluster_CA_domain>:8500` na lista `NO_PROXY`. Em que `<cluster_CA_domain>` é o domínio de autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

        ```
        sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf
        ```

        A atualização é semelhante ao código a seguir.

        ```
        [Service]
         Environment="HTTP_PROXY=http://1.2.3.4:3128" "HTTPS_PROXY=http://1.2.3.4:3128"
        "NO_PROXY=localhost,127.0.0.1,<cluster_CA_domain>:8500"
        ```

    2. Reinicie o serviço do Docker.

        ```
        sudo systemctl daemon-reload
        sudo systemctl restart docker
        ```

Para o sistema operacional Linux

1. No sistema que hospeda a imagem do Docker, inclua a linha a seguir no arquivo `/etc/hosts`.

    ```
    <Cluster Master Host> <cluster_CA_domain>
    ```

    Em que `<Cluster Master Host>` está definido em Terminal principal.

2. Em seu computador, crie um diretório para armazenar o certificado de registro do Docker.

    ```
    mkdir /etc/docker/certs.d/<cluster_CA_domain>:8500/
    ```

3. Na máquina do cliente (sistema operacional Linux®), proteja uma cópia do certificado de registro do nó principal do cluster do IBM Cloud Private. O `<user>` no comando a seguir é o usuário que possui permissões sudo.

    ```
    scp <user>@<cluster_CA_domain>:/etc/docker/certs.d/<cluster_CA_domain>\:8500/ca.crt
    /etc/docker/certs.d/<cluster_CA_domain>\:8500/ca.crt
    ```

4. No computador cliente, reinicie o serviço do Docker executando o comando a seguir:

    ```
    service docker restart
    ```

5. Efetue login em seu registro de imagem privado executando o comando a seguir:

    ```
    docker login <cluster_CA_domain>:8500
    ```

Para o sistema operacional macOS

1. No computador cliente, inclua a linha a seguir no arquivo `/etc/hosts`:

    ```
    <Cluster Master Host> <cluster_CA_domain>
    ```

    Em que `<Cluster Master Host>` está definido em Terminal principal.

2. Na máquina cliente (macOS), proteja uma cópia do certificado de registro no nó principal de seu cluster do IBM Cloud Private.

```
mkdir -p ~/.docker/certs.d/<cluster_CA_domain>\:8500 scp
root@<cluster_CA_domain>:/etc/docker/certs.d/<cluster_CA_domain>\:8500/ca.crt
~/.docker/certs.d/<cluster_CA_domain>\:8500/ca.crt
```

3. No computador cliente, inclua o certificado no keychain.

```
sudo security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain
~/.docker/certs.d/<cluster_CA_domain>\:8500/ca.crt
```

4. No computador cliente, reinicie o serviço do Docker.
5. Efetue login em seu registro de imagem privado executando o comando a seguir:

```
docker login <cluster_CA_domain>:8500
```

Para o sistema operacional Windows

1. No sistema Windows™ que hospeda a imagem do Docker, inclua a linha a seguir no arquivo `%SystemRoot%\System32\drivers\etc\hosts`.

   ```
   <Cluster Master Host> <cluster_CA_domain>
   ```

   Em que `<Cluster Master Host>` está definido em Terminal principal.

2. Inclua o certificado de registro do Docker concluindo as seguintes etapas:
   1. Selecione **Iniciar** > **Ferramentas Administrativas** > **Gerenciar certificados de computador**.
   2. Clique com o botão direito em **Autoridades de certificação raiz confiável** e selecione **Todas as tarefas** > **Importar**.
   3. Navegue para localizar e selecionar o arquivo `.crt`.
   4. Conclua o assistente para configurar o certificado. Os padrões geralmente são aceitáveis.
   5. Reinicie o Docker para Windows para aplicar as mudanças.
3. No nó principal de seu cluster do IBM Cloud Private, proteja uma cópia do certificado de registro para o computador.

   ```
   scp /etc/docker/certs.d/<cluster_CA_domain>\:8500/ca.crt \
   root@<client_node>:/etc/docker/certs.d/<cluster_CA_domain>\:8500/
   ```

4. Efetue login em seu registro de imagem privado executando o comando a seguir:

   ```
   docker login <cluster_CA_domain>:8500
   ```

Boot2Docker para Windows

1. Em seu computador, execute o comando a seguir para criar um diretório `boot2docker` para executar contêineres do Docker:

   ```
   mkdir /var/lib/boot2docker/certs
   ```

2. Converta suas certificações para o formato `.pem`. Copie suas certificações para seu `boot2docker`. Execute os comandos a seguir: para converter sua certificação:

   ```
   openssl x509 -inform der -in certificate.cer -out certificate.pem
   ```

   ```
   cp /c/Users/my.username/certs / * .pem /var/lib/boot2docker/certs/
   ```

3. Crie um arquivo vazio para seus certificados. A criação de um arquivo vazio permite que seus certificados sejam copiados para o diretório correto. Execute o comando a seguir:

   ```
   touch /var/lib/boot2docker/bootlocal.sh & & chmod + x /var/lib/boot2docker/bootlocal.sh
   ```

4. Com seu editor de texto, execute o comando a seguir para acessar seu arquivo:

   ```
   vi /var/lib/boot2docker/bootlocal.sh
   ```

5. Salve seu arquivo executando o comando a seguir:

   ```
   mkdir -p /etc/docker/certs.d & & cp certs/certificate.pem /etc/docker/certs.d
   ```

6. Reinicie o computador. Execute o comando a seguir:

   ```
   docker-padrão de reinicialização da máquina
   ```

Enviando por push e efetuando pull de imagens

É possível enviar por push ou efetuar pull de imagens em seu sistema de arquivos local para o registro de imagem privado.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Pré-requisitos:

- Instale o Docker no computador. Consulte Instalar Docker.
- Configure a CLI do Docker. Consulte Configurando a autenticação para a CLI do Docker
- Se seu cluster usar uma arquitetura de alta disponibilidade, assegure-se de montar um diretório compartilhado sob `/var/lib/registry`. Consulte Clusters do IBM® Cloud Private de alta disponibilidade.

Etapas:

1. Efetue login em seu registro de imagem privado.

   ```
   docker login <cluster_CA_domain>:8500
   ```

   `<cluster_CA_domain>` é o domínio de autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação. Se você não especificar um nome de domínio de CA, o valor padrão será `mycluster.icp`.

2. Envie por push ou efetue pull da imagem necessária. Deve-se enviar por push ou fazer pull da imagem para/de um namespace existente. É possível enviar por push ou fazer pull da imagem somente se o recurso de namespace estiver designado a uma equipe para a qual você tem a função correta. Os administradores e operadores podem enviar por push ou fazer pull da imagem. Editores e visualizadores podem extrair imagens. A menos que você especifique um `imagePullSecret`, é possível acessar a imagem somente por meio do namespace que a hospeda. Para obter mais informações sobre namespaces, consulte Namespaces.

   Se você estiver usando um `serviceAccount` padrão e a imagem no mesmo namespace, a conta do serviço padrão será configurada automaticamente com um novo segredo de pull de imagem quando a imagem for carregada. Neste cenário, a conta do serviço está apta a fazer pull da imagem.

   Qualquer outra conta do serviço no mesmo namespace pode fazer pull da imagem somente sob as condições a seguir:

   - A conta de serviço é corrigida com um segredo de pull de imagem válido.
   - A especificação de POD inclui o nome de um segredo de pull de imagem válido.
   - O escopo da imagem é mudado para `global` após a imagem ser enviada por push.

     - Para enviar por push uma imagem de seu sistema de arquivos local para o registro de imagem privado, execute os comandos a seguir. O nome da imagem que você envia por push deve ter menos de 253 caracteres de comprimento.

       1. Identifique a imagem. É possível obter o `imagename` e o `tagname` para sua imagem executando o comando `docker images`.

          ```
          sudo docker tag imagename: tagname < cluster_CA_domain>:
          8500/namespacename/imagename:tagname
          ```

          `namespacename` é o nome do namespace ao qual você designa a imagem.

       2. Envie a imagem por push para o registro de imagem privado.

          ```
          sudo docker push < cluster_CA_domain>: 8500/namespacename/imagename:tagname
          ```

          Por padrão, a imagem é incluída no `scope namespace`. Se você desejar que a imagem esteja disponível para todos os namespaces, mude seu `scope` para `global`. Consulte Mudando o escopo da imagem. A imagem é armazenada em um repositório que contém apenas imagens que são designadas a esse nome.

     - Para efetuar pull de uma imagem em seu registro de imagem privado para seu sistema de arquivos local, execute o comando pull do Docker.

```
sudo docker pull < cluster_CA_domain>: 8500/namespacename/imagename:tagname
```

3. (Opcional) Implemente um aplicativo ou tarefa usando a imagem recém-incluída. Para obter mais informações sobre como implementar aplicativos e tarefas, consulte Criando implementações.

Mudando o escopo da imagem

É possível mudar o escopo de uma imagem para restringir ou expandir sua disponibilidade dentro do cluster.

Por padrão, as imagens no registro de imagem privado são designadas ao escopo do `namespace`. É possível disponibilizar as imagens para todos os namespaces configurando um escopo `global` ou limitar as imagens a um namespace específico.

- Mudando o escopo da imagem por meio da console de gerenciamento
- Mudando o escopo da imagem por meio da linha de comandos

Tipo de usuário ou nível de acesso necessário: administrador de cluster, administrador da equipe, editor ou operador

Deve-se ter acesso ao namespace que atualmente contém a imagem. É possível designar imagens para um escopo global ou de namespace. Uma imagem com o escopo namespace é acessível apenas de um único namespace, mas uma imagem com o escopo global pode ser acessada de todos os namespaces.

Mudando o escopo da imagem por meio da console de gerenciamento

1. No menu de navegação, clique em **Imagens do contêiner**.
2. Para a imagem que você deseja atualizar, clique no botão **Abrir e fechar lista de opções** e selecione **Mudar escopo**.
3. Selecione o escopo no menu suspenso na caixa de diálogo Imagem.
4. Clique em **Mudar escopo da imagem**.

Mudando o escopo da imagem por meio da linha de comandos

1. Configure a CLI kubectl. Consulte Acessando o cluster do IBM Cloud Private usando a CLI do kubectl para obter informações adicionais.

2. Edite a imagem.

   - Para mudar o escopo de `namespace` para `global`, execute o comando a seguir:

     ```
     kubectl get image <image name> -n=namespace -o yaml \
     | sed 's/scope: namespace/scope: global/g' | kubectl replace -f -
     ```

     O valor `<image name>` é o nome da imagem que você deseja mover.

   - Para mudar o escopo de `global` para `namespace`, execute o comando a seguir:

     ```
     kubectl get image <image name> -n=namespace -o yaml \
     | sed 's/scope: global/scope: namespace/g' | kubectl replace -f -
     ```

     Por exemplo, para mudar o escopo de uma imagem que é denominada `dev/keepalived` de `global` para `namespace`. Execute o comando a seguir:

     ```
     kubectl get image keepalived -n=dev -o yaml \
     | sed 's/scope: global/scope: namespace/g' | kubectl replace -f -
     ```

O escopo de sua imagem foi mudado.

Criando imagePullSecrets para um namespace específico

Um **imagePullSecrets** é um token de autorização, também conhecido como um segredo, que armazena credenciais do Docker que são usadas para acessar um registro.

Dois formatos estão disponíveis para você criar um aplicativo do console de gerenciamento. É possível criar aplicativos inserindo os valores de parâmetro na janela Criar implementações ou colando um arquivo YAML na janela "Criar recurso".

Se você deseja usar imagens de um namespace diferente em seu registro de imagem privado, deve-se fornecer o valor **imagePullSecrets** para esse namespace no arquivo YAML.

Para criar o **imagePullSecrets**:

1. Instale a interface da linha de comandos `kubectl` e configure a conexão com o cluster do IBM® Cloud Private. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Alterne para o namespace no qual você deseja criar a implementação.

   ```
   kubectl config set-context <cluster_name>-context --user=<user_name> --namespace=
   <namespace_name>
   ```

   Em que `<cluster_name>` é o nome do cluster, conforme definido em ConfigMap de configuração de cluster.

3. Crie o segredo. Execute o comando a seguir:

   ```
   kubectl create secret docker-registry myregistrykey --docker-server=<cluster_CA_domain>:8500 --
   docker-username=<user_name> --docker-password=<user_password> --docker-email=<user_email>
   ```

   Em que `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

4. Visualize o segredo. Execute o comando a seguir:

   ```
   kubectl get secret
   ```

   A saída se assemelha ao texto a seguir:

   ```
   NAME                  TYPE                                  DATA      AGE
   myregistrykey         kubernetes.io/dockercfg               1         5d
   default-token-5gjfc   kubernetes.io/service-account-token   3         5d
   ```

   Nesse exemplo, o segredo `myregistrykey` está disponível para uso no namespace `default`.

Usando o imagePullSecret em uma implementação

Inclua o parâmetro **imagePullSecrets** no arquivo YAML da implementação. Coloque o parâmetro **imagePullSecrets** na seção de especificação da seção de modelos, conforme mostrado na amostra a seguir:

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx-demo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      hostNetwork: false
      containers:
      - name: nginx
        image: mycluster.icp:8500/developer/nginx
        ports: []
        resources:
          limits: {}
      imagePullSecrets:
      - name: myregistrykey
```

Removendo uma imagem do console

É possível remover uma imagem da console de gerenciamento do IBM® Cloud Private.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. No menu de navegação, clique em **Imagens do contêiner**.
2. Para a imagem que você deseja remover, selecione **Ação** > **Remover**.
3. Clique em **Remover imagem**. A imagem é removida da console de gerenciamento.

4. Remova os arquivos de imagem do armazenamento de registro privado.

Removendo arquivos de imagem do armazenamento de registro privado

A exclusão de uma imagem do console de gerenciamento do IBM Cloud Private é uma exclusão recuperável. Para uma exclusão recuperável, a imagem é removida do cluster do IBM Cloud Private, no entanto os arquivos para a imagem permanecem no armazenamento de registro privado. Para excluir os arquivos do armazenamento de registro de imagem, deve-se usar a ferramenta garbage collection (GC) do registro.

Para remover imagens do armazenamento de registro privado, conclua as etapas a seguir:

1. Desative o gerenciador de imagem. Se as imagens são enviadas por push para o registro enquanto o processo GC está em execução, há um risco de que as camadas da nova imagem sejam excluídas. Isso pode resultar em uma imagem corrompida. Para desativar o gerenciador de imagem, execute o comando a seguir:

   ```
   kubectl patch svc image-manager -n kube-system  -p '{"spec": {"selector": {"app": "image-manager-dummy"}}}'
   ```

   A saída se assemelha ao código a seguir:

   ```
   Serviço "image-manager" corrigido
   ```

2. Visualize os arquivos que devem ser limpos pelo processo de coleta de lixo.

   ```
   kubectl exec -it image-manager-0 -c icp-registry  -n kube-system -- registry garbage-collect --dry-run /etc/docker/registry/config.yml
   ```

   NOTA: a opção `--dry-run` imprime o plano de limpeza sem remover quaisquer dados.

3. Execute a ferramenta de coleta de lixo.

   ```
   kubectl exec -it image-manager-0 -c icp-registry  -n kube-system -- registry garbage-collect /etc/docker/registry/config.yml
   ```

4. Remova as pastas do repositório de imagem órfã.

   ```
   kubectl exec -it image-manager-0 -c icp-registry  -n kube-system --  /bin/sh -c "find /var/lib/registry/docker/registry/v2/repositories/ -maxdepth 2  -mindepth 2 | tee /tmp/image_all"

   kubectl exec -i image-manager-0 -c icp-registry  -n kube-system -- /bin/sh -c "registry garbage-collect --dry-run  /etc/docker/registry/config.yml 2>&1 |grep 'marking manifest'| cut -d ":" -f 0 |xargs -n1 echo '/var/lib/registry/docker/registry/v2/repositories/' |sed 's/ //g'|tee /tmp/image_valid"

   kubectl exec -i image-manager-0 -c icp-registry  -n kube-system -- /bin/sh -c  "grep -F -v  -f /tmp/image_valid  /tmp/image_all |tr -d '\r'| xargs -n1 rm -rf  2>&1"
   ```

5. Ative o gerenciador de imagem.

   ```
   kubectl patch svc image-manager -n kube-system  -p '{"spec": {"selector": {"app": "image-manager"}}}'
   ```

   A saída se assemelha ao código a seguir:

   ```
   Serviço "image-manager" corrigido
   ```

Anotando cargas de trabalho para o serviço de medição do IBM Cloud Private

Os gráficos Helm que você cria para implementar no IBM Cloud Private podem incluir anotações que identificam o identificador, o nome e a versão para a oferta do produto. Esses rótulos são especificados nos metadados do gráfico Helm e descobertos pelo daemon de medição. As anotações são usadas para associar métricas de tempo de execução medidas com a instância da oferta que está em execução.

Nota: liberações anteriores do serviço de medição IBM Cloud Private também podiam usar rótulos de imagem do Docker em vez de anotações, mas essa capacidade não é mais permitida para segurança aprimorada. Você deve usar as anotações do gráfico Helm.

É possível especificar as seguintes anotações:

- **productID**
  - Use esta anotação para especificar um identificador de produto que identifique exclusivamente a oferta.
- **productName**
  - Use esta anotação para especificar um nome de produto que identifique o nome legível para a oferta.
- **productVersion**
  - Use esta anotação para especificar um identificador de versão do produto que especifique a versão, a liberação, a modificação e o nível de correção (v.r.m.f) da oferta.

A seção de anotação do arquivo YAML que é usado pelo gráfico Helm é mostrada no exemplo a seguir:

```
kind: Deployment
spec:
  template:
     metadata:
       annotations:
          productName: My Product
          productID: ABCD1234
          productVersion: v1.10.0
```

É possível ter múltiplos contêineres que estão contidos em um gráfico Helm único. Para especificar valores separados para cada contêiner na mesma sequência, forneça um caractere de barra vertical `|` seguido por um par de chave-valor `codeontainerName:productString` para cada contêiner. O primeiro caractere será o caractere de barra vertical `|`. Por exemplo, em um gráfico contendo três contêineres, o `productName` YML especifica um nome de produto diferente para cada contêiner:
`productName: '|containerName1:Product Name 1|containerName2:Second Product Name|containerName3:Final Product`

As anotações são usadas para associar métricas que são reunidas para propósitos de medição com a oferta implementada. O licenciamento para essas ofertas depende dos termos e condições de autorização quando a oferta é comprada e não pelo serviço de medição. O serviço de medição simplesmente mede as métricas para a oferta em execução e relata esse uso para o administrador.

Impondo segurança da imagem do contêiner

Ao usar o recurso do IBM Container Image Security Enforcement, é possível verificar a integridade de suas imagens de contêiner antes de implementá-las em um cluster do IBM Cloud Private.

O IBM Container Image Security Enforcement controla de onde as imagens são implementadas e cumpre as políticas do Vulnerability Advisor (VA). Se uma imagem não atender aos seus requisitos de política definidos, o pod não será implementado.

Definição de política

Para cada imagem em um repositório, um escopo de política de imagem de `cluster` ou `namespace` é aplicado. Ao implementar um aplicativo, o IBM Container Image Security Enforcement verifica se o namespace do Kubernetes que você está implementando possui quaisquer regulamentos de política que devem ser aplicados. Se uma política `namespace` não existir, então a política `cluster` será aplicada. Se as políticas de `namespace` e de `cluster` se sobrepuserem, o escopo do cluster será ignorado. Se nenhuma das políticas de escopo `cluster` ou `namespace` existir, sua implementação falhará ao ser ativada. É possível que você veja uma mensagem de erro semelhante à seguinte:

```
... release ... failed: Internal error occurred: admission webhook
"trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the request:
Deny "docker.io/rook/rook:v0.7.1", no matching repositories in ClusterImagePolicy and no
ImagePolicies in the "default" namespace
```

Nota: qualquer pod que é implementado em namespaces que são reservados para os serviços do IBM Cloud Private ignora a verificação de segurança da imagem do contêiner. Os namespaces a seguir são reservados para os serviços do IBM Cloud Private:

- `kube-system`
- `cert-manager`
- `istio-system`

Para resolver o problema, crie uma política.

A definição de política é configurada no arquivo `<installation_cluster>/cluster/config.yaml` ou usando o console da web.

```
apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
kind: <ClusterImagePolicy_or_ImagePolicy>
metadata:
  name: <crd_name>
spec:
   repositories:
    - name: <repository_name>
      policy:
        va:
          enabled: <true_or_false>
```

- `<repository_name>` - especifica os repositórios para o quais permitir imagens. Esta é a lista de repositórios que possuem conteúdo confiável. Um caractere curinga (`*`) é permitido no nome do repositório. Este caractere curinga (`*`) denota que as imagens de todos os repositórios são permitidas ou confiáveis. Para configurar todos os repositórios para confiável, configure o nome do repositório para (`*`) e omita as subseções de política. Os repositórios por padrão requerem uma verificação de política, com exceção do repositório padrão `mycluster.icp:8500`. Um valor de nome de repositório vazio ou em branco bloqueia a implementação de todas as imagens.
- Quando o `va` é configurado como `enabled: true` para um registro de contêiner, qualquer tentativa de implementar pods de imagens nesse registro é bloqueada. Se você deseja implementar imagens a partir desses registros, deve-se remover a especificação de política `va`. O registro de contêiner integrado do IBM Cloud Private padrão é o único registro que suporta o cumprimento da política do Vulnerability Advisor.

Ativando e desativando o IBM Container Image Security Enforcement

O IBM Container Image Security Enforcement está disponível como um gráfico do Kubernetes. O IBM Container Image Security Enforcement é ativado por padrão durante a instalação do IBM Cloud Private.

Para desativar o IBM Container Image Security Enforcement, inclua, durante a instalação, o `image_security_enforcement` na lista de serviços que estão desativados (`management_services`) no cluster do IBM Cloud Private `config.yaml`.

Após a instalação de um cluster do IBM Cloud Private, o administrador de cluster pode desinstalar o gráfico Kubernetes.

Política padrão

A política de imagem de cumprimento de segurança padrão é do escopo do `cluster`. Com essa política, somente as imagens que são armazenadas no registro de contêiner integrado (a primeira na lista de desbloqueio de política) e as imagens que são usadas no Catalog do IBM Cloud Private (outras na lista de desbloqueio de política) podem ser usadas no cluster. Por exemplo:

```
image-security-enforcement:
  clusterImagePolicy:
    - name: "{{ cluster_CA_domain }}:8500/*"
    -name: "registry.bluemix.net/ibm / *"
    -Nome:...
```

Nota: o Container Image Security Enforcement do Vulnerability Advisor (VA) não se aplica à `Política padrão`.

Customizando sua política (durante a instalação)

É possível modificar a política de imagem no nível de `cluster` ou `namespace` depois de instalar o cluster do IBM Cloud Private. Em sua política, é possível especificar regras de cumprimento diferentes para diferentes imagens.

Também é possível predefinir a política de imagem do escopo do `cluster` antes de instalar o IBM Cloud Private. Esta configuração predefinida sobrescreve a política de imagem do escopo do `cluster` padrão durante a instalação.

Para predefinir a política de imagem do escopo do `cluster` antes da instalação, modifique o arquivo `config.yaml`.

Por exemplo, ao incluir o seguinte no arquivo `config.yaml`, permita que todas as imagens que estão no repositório `quay.io` sejam usadas para implementações em seu cluster.

```
image-security-enforcement:
  clusterImagePolicy:
    - name: "quay.io/*"
      policy:
```

Customizando sua política (pós-instalação)

Também é possível implementar a política como um objeto Kubernetes após a instalação de seu cluster. Para implementar a política como um objeto Kubernetes, use o comando `kubectl apply`.

1. Crie um arquivo `policy.yaml` que contenha as especificações de política. A seguir estão algumas configurações de política de amostra que podem ser usadas para seu arquivo `policy.yaml`.

   - Essa política permite imagens de contêiner do registro do contêiner do Docker Hub, do registro do contêiner do CoreOS, do registro de contêiner do Google, do registro de contêiner do Azure, do registro de contêiner do Amazon Elastic e do registro de contêiner da IBM.

     ```
     apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
     kind: ClusterImagePolicy
     metadata:
     name: ibmcloud-default-cluster-image-policy
     spec:
      repositories:
      # Docker hub Container Registry
       - name: "docker.io/*"
         policy:

       # Registro do Contêiner do CoreOS
       - name: "quay.io/*"
         policy:

       # Google Container Registry
       - name: "gcr.io/*"
         policy:

       # Azure Container Registry
       - name: "*azurecr.io/*"
         policy:

       # Amazon Elastic Container Registry
       - name: "*amazonaws.com/*"
         policy:

       # IBM Container Registry
       - name: "registry*.bluemix.net/*"
         policy:
     ```

   - Essa política permite imagens de qualquer registro do contêiner.

     ```
     apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
     kind: ClusterImagePolicy
     metadata:
     name: ibmcloud-default-cluster-image-policy
     spec:
      repositories:
       # allow all images
       - nome: "*"
         política:
     ```

   - Essa política nega todas as imagens de qualquer Registro de contêiner, incluindo a imagem do IBM Container Image Security Enforcement.

     ```
     apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
     kind: ClusterImagePolicy
     metadata:
     name: ibmcloud-default-cluster-image-policy
     spec:
      repositories:
     ```

2. Aplique a política.

   ```
   kubectl apply -f policy.yaml
   ```

Cumprimento de segurança de imagem usando o console da web do IBM Cloud Private

É possível criar uma política de cumprimento de imagem usando o console da web do IBM Cloud Private que configura diretrizes para os pods que são criados em seu cluster. Conclua as etapas a seguir para criar uma política de imagem:

1. Efetue login no console da web do IBM Cloud Private de seu cluster com um ID que possua acesso de administrador de cluster.
2. No menu de navegação, selecione **Gerenciar** > **Segurança do recurso**.
3. Selecione **Políticas de Imagem** . Uma lista de políticas de imagem disponíveis é exibida.
4. Selecione **Criar Política de Imagem** .
5. Insira um nome para a política de imagem. O nome deve ser uma única sequência contendo apenas letras , números, sublinhados (_) e hífens (-).
6. Selecione o *Escopo* da política para definir quais recursos são restritos pela política a partir das opções a seguir:
   - Cluster - a política se aplica a tudo no cluster atual.
   - Namespace - a política se aplica a tudo no namespace especificado.
7. Se você selecionou namespace como seu escopo, selecione um namespace existente na lista para identificar qual nomear.
8. Selecione **Incluir registro** para especificar uma área confiável a partir da qual é possível receber imagens. O comportamento padrão é rejeitar todas as imagens. Ao incluir um registro, ele identifica esse local como uma origem de imagens permitidas.
   1. Especifique uma *URL de registro* para permitir esse registro como uma origem confiável. A *URL de registro* tem um formato semelhante aos exemplos a seguir:
      - Permitir todas as imagens do docker hub: `docker.io/*`
      - Permitir todas as imagens do repositório ibmcom: `docker.io/ibmcom/*`
   2. Especifique se você deseja executar a política de varredura do Vulnerability Advisor, caso ele tenha sido instalado. Se ele estiver instalado e essa configuração estiver ativada, a imagem deverá ser aprovada na varredura do Vulnerability Advisor antes de ser instalada.
   3. Selecione **Incluir** para criar o novo registro.
9. Selecione **Incluir** para salvar e criar a nova política de imagem.
10. É possível remover uma política de imagem selecionando o ícone *Abrir e fechar a lista de opções* (...) para a política de imagem e, em seguida, selecionando **Remover**.

Gerenciando Cargas de Trabalho

Saiba como atualizar e monitorar seus aplicativos, configurações, serviços e políticas.

Este guia supõe que os usuários estejam familiarizados com os conceitos e a terminologia do Kubernetes. Termos e componentes chaves do Kubernetes não estão definidos. Para obter informações adicionais sobre conceitos do Kubernetes, consulte a Documentação do Kubernetes.

- Criando DaemonSets
- Gerenciando implementações
- Gerenciando liberações do Helm
- Gerenciando tarefas
- Criando StatefulSets
- Gerenciando ReplicaSets

Usando o Gerenciador de certificados do IBM Cloud Private (cert-manager)

É possível usar o cert-manager do IBM Cloud Private para criar e montar um certificado para um Deployment, StatefulSet ou DaemonSet do Kubernetes. Também é possível criar e incluir um certificado em um Ingresso do Kubernetes.

Issuer, *ClusterIssuer* e *Certificate* são tipos de recursos do Kubernetes que foram introduzidos para suportar a geração e o gerenciamento de ciclo de vida de certificados. Para obter mais informações sobre cert-manager, consulte a Documentação da

comunidade do cert-manager.

Consulte a lista a seguir para saber como o cert-manager do IBM Cloud Private funciona:

- O Issuer assina novos certificados e pares de chaves.
- O certificado representa um certificado X.509 e um par de chaves para o TLS ou autenticação.
- O certificado é armazenado como um Segredo do Kubernetes.
- O certificado é renovado automaticamente.

Primeiro, crie um Emissor e, em seguida, crie um certificado que será assinado pelo Emissor. O gerenciador de certificado IBM Cloud Private gera um certificado X.509 e um par de chaves e os armazena em um Segredo do Kubernetes.

Para obter mais informações sobre o Gerenciador de Certificados e outras ferramentas de configuração, consulte a documentação do produto a seguir:

- Criando seus próprios Emissores autoassinados e de CA
- Criando certificados cert-manager do IBM Cloud Private
- Visualizando recursos do cert-manager do IBM Cloud Private
- Atualizando certificados do (cert-manager) do IBM Cloud Private
- Incluindo certificados usando o Vault Issuer
- Incluindo certificados usando o Emissor Acme
- Incluindo certificados usando o algoritmo ECDSA para criptografia

Para obter informações sobre como atualizar, substituir e restaurar certificados criados e gerenciados pelo instalador, consulte Certificados no IBM Cloud Private

Criando seus próprios Emissores CA e autoassinados

Para criar certificados no IBM® Cloud Private que são gerenciados pelo Gerenciador de Certificados, deve-se primeiramente criar um Emissor.

Crie um Emissor autoassinado e, em seguida, use esse Emissor para criar um certificado CA. Seu certificado de CA pode ser gerenciado pelo Gerenciador de Certificados.

1. Crie um Emissor autoassinado. Use o arquivo `.yaml` a seguir para definir um Emissor autoassinado.

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Issuer
metadata:
 name: hello-myself-tls
 namespace: foobar
spec:
 selfSigned: {}
```

2. Depois de criar o Emissor autoassinado, crie um certificado de CA que referencie o Emissor autoassinado e especifique o campo `isCA`.

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Certificate
metadata:
 name: hello-ca-tls
 namespace: foobar
spec:
 # name of the tls secret to store
 # the generated certificate/key pair
 secretName: hello-deployment-tls-ca-key-pair
 isCA: true
 issuerRef:
   # issuer created in step 1
   name: hello-myself-tls
   kind: Issuer
 commonName: "foo1.bar1"
 dnsNames:
 # one or more fully-qualified domain name
 # can be defined here
 - foo1.bar1
```

3. Edite a amostra a seguir de um Emissor que referencie o segredo anterior. Edite `<name>` e `<namespace>` na seção *metadata* do arquivo `.yaml`. Certifique-se de que `secretName` da seção *spec* corresponda ao `secretName` da etapa anterior:

   ```
   apiVersion: certmanager.k8s.io/v1alpha1 kind: Issuer metadata: name: hello-deployment-tls
   namespace: foobar spec: ca: secretName: hello-deployment-tls-ca-key-pair
   ```

O segredo `hello-deployment-tls-ca-key-pair` foi criado pelo gerenciador de Certificados e é gerenciado, juntamente com outros certificados. Esse certificado de CA pode ser usado pelo Emissor `hello-deployment-tls`.

Também é possível fornecer o certificado de CA e a chave privada em vez de usar o Gerenciador de Certificados para criá-lo. Execute o comando a seguir para criar seu segredo com seus arquivos de certificado de CA e de pares de chaves: `ca.crt` e `ca.key`.

```
kubectl create secret tls hello-deployment-tls-ca-key-pair --cert=ca.crt --key=ca.key --
namespace=foobar
```

Este é o Segredo que você especificaria no Emissor que você criou anteriormente na etapa 3.

É possível criar um Certificado assinado por este Emissor mudando o `issuerRef.name` do Certificado para o nome deste Emissor. Para obter mais informações, consulte Incluindo um certificado para uma carga de trabalho do Kubernetes.

Consulte Sobre o Gerenciador de Certificados do IBM Cloud Private (cert-manager) para obter mais tópicos do Gerenciador de Certificados.

Criando certificados do Gerenciador de Certificados do IBM Cloud Private (cert-manager)

O serviço do Gerenciador de Certificado do IBM® Cloud Private é usado para emitir e gerenciar certificados para serviços que são executados no IBM Cloud Private. O gerenciador de certificados é baseado no Projeto jetstack/cert-manager.

- Incluindo um certificado em uma carga de trabalho do Kubernetes
- Incluindo um certificado em um Kubernetes Ingress
- Customizado certificados do gerenciador de certificado (cert-manager) do IBM Cloud Private

Incluindo um certificado em uma carga de trabalho do Kubernetes

1. Defina um Emissor. É possível definir *Issuers* (com escopo no namespace) ou *ClusterIssuers* (com escopo no cluster). IBM Cloud Private O serviço do gerenciador de certificados suporta os seguintes tipos de emissores. Para obter informações sobre como definir e criar Emissores, selecione os links aplicáveis na seguinte lista:

   - Emissores de CA e autoassinados
   - Emissor de Área Segura
   - Emissor Acme

2. Para definir o Certificado, edite os `metadata` no qual o `<name>` está associado ao certificado e o `<namespace>` é o local onde o certificado é criado. Além disso, edite a seção `spec` do exemplo a seguir. O exemplo a seguir define um certificado que usa o Emissor Autoassinado a partir das instruções do Emissor Autoassinado na etapa um. Observe que o `<namespace>` do certificado corresponde ao `<namespace>` do Emissor:

   ```
   apiVersion: certmanager.k8s.io/v1alpha1
   kind: Certificate
   metadata:
     name: hello-deployment-tls-1
     namespace: foobar
   spec:
     # name of the tls secret to store
     # the generated certificate/key pair
     secretName: hello-deployment-tls-1
     issuerRef:
       # Issuer Name
       name: hello-myself-tls
       # The default value is Issuer (i.e.
       # a locally namespaced Issuer)
       kind: Issuer

     commonName: "foo1.bar1"
     dnsNames:
   ```

```
  # one or more fully-qualified domain names
  # can be defined here
- foo1.bar1
ipAddresses:
  # one or more IP addresses can be defined here
- 0.0.0.0
- 127.0.0.1
```

3. Mounte o Segredo no Deployment, DaemonSet ou StatefulSet.

   O Segredo do Kubernetes que contém o certificado é montado no sistema de arquivos da mesma maneira que qualquer outro segredo. Para obter mais informações, consulte a Documentação do Kubernetes.

Incluindo um certificado para Kubernetes Ingress

O tipo de recurso Ingresso do Kubernetes é usado para expor serviços a uma rede externa. Os certificados gerados pelo cert-manager podem ser incluídos em recursos de ingresso. O IBM Cloud Private fornece um ponto de Ingresso do NGINX Kubernetes que está pronto para uso imediato.

Manipulando Diversos Nomes de Domínios

As solicitações para múltiplos hosts virtuais são manipuladas pelo mesmo Ingresso. Cada host virtual pode ser finalizado com seus próprios certificados. Nesse caso, o nome completo do domínio na solicitação TLS/HTTPS é usado para identificar o host virtual solicitado. A extensão do protocolo TLS-SNI define esse processo.

Conclua o procedimento a seguir para proteger o Ingresso do Kubernetes:

1. Defina um Emissor. IBM Cloud Private O serviço do gerenciador de certificados suporta os seguintes tipos de emissores. Clique no link para ver como definir e criar cada um:

   - Emissores de CA e autoassinados
   - Emissor de Área Segura
   - Emissor Acme

2. Para definir o Certificado, edite os `metadados`, em que `<name>` é associado ao Certificado e o `<namespace>` é onde o Certificado é criado. Além disso, edite a seção `spec` da amostra a seguir, que define um Certificado que usa o Emissor de CA das instruções do Emissor de CA em uma etapa. Observe que o `<namespace>` do Certificado corresponde ao `<namespace>` do Emissor:

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Certificate
metadata:
 name: hello-k8s-ingress-tls-1
 namespace: foobar
spec:
 # name of the tls secret to store
 # the generated certificate/key pair
 secretName: hello-k8s-ingress-tls-1
 issuerRef:
   # Issuer Name
   name: hello-ca-tls
   # The default value is Issuer (i.e.
   # a locally namespaced Issuer)
   kind: Issuer
 commonName: "foo1.bar1"
 dnsNames:
  # one or more fully-qualified domain names
  # can be defined here
 - foo1.bar1
```

   O cert-manager cria o certificado com base na definição de recurso do certificado e o armazena como um Segredo do Kubernetes.

3. Inclua o Segredo no Ingresso do Kubernetes. O exemplo a seguir define um Ingresso do Kubernetes ativado para TLS que está integrado com o cert-manager. Aqui, `hello-k8s-ingress-tls-1` corresponde ao `secretName` que você definiu anteriormente e `host` corresponde ao nome DNS que você definiu anteriormente no certificado.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
```

```
    name: hello-k8s-ingress-tls
    annotations:
      kubernetes.io/ingress.class: "nginx"
      ingress.kubernetes.io/rewrite-target: "/"
   spec:
    tls:
    # k8s ingress defines different tls certificates
    # for each nginx server blocks.
     # k8s ingress default cert is used if
    # no host-specific secret specified
    - hosts:
    # this is the fully-qualified domain name
    # of the first server block
      - foo1.bar1
      # certificate hello-k8s-ingress-tls-1
      # is only used by foo1.bar1
      secretName: hello-k8s-ingress-tls-1
    rules:
      # each server block redirects request
      # to its own backend service
    - host: foo1.bar1
      http:
        paths:
        - backend:
            serviceName: hello-world-svc
            servicePort: 80
          path: /fb
```

Nota: certificados que são criados pelo cert-manager são renovados automaticamente antes da expiração. Consulte Atualizando certificados do gerenciador de certificado (cert-manager) do IBM Cloud Private para obter mais informações.

Consulte o Usando o Gerenciador de certificador do IBM Cloud Private (cert-manager) para obter mais tópicos do gerenciador de certificado.

Customizando certificados do cert-manager do IBM Cloud Private

O serviço do gerenciador de certificado IBM® Cloud Private oferece recursos que podem ser usados para customizar seus certificados.

Customizando a expiração do certificado

A expiração padrão para todos os certificados é de 90 dias. O tempo padrão antes da expiração, quando o serviço do gerenciador de certificado IBM® Cloud Private renova os certificados, é de 30 dias.

O serviço Gerenciador de certificados do IBM Cloud Private oferece prazos de expiração customizados. O recurso é oferecido por meio dos novos campos `duration` e `renewBefore` em uma definição de Certificado.

`duration` significa por quanto tempo o certificado é válido. `renewBefore` é o tempo antes de o certificado expirar quando o serviço do gerenciador de certificado IBM Cloud Private tenta renovar o certificado.

A definição de Certificado a seguir no exemplo abaixo tem um período de validade de 30 dias (720 horas). O serviço do Gerenciador de certificados do IBM Cloud Private tenta renovar este certificado 10 dias (240 horas) antes de ele expirar.

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Certificate
metadata:
  name: hello-world-cert-1
  namespace: default
spec:
  secretName: hello-world-cert-1
  issuerRef:
    name: hello-ca-tls
    kind: Issuer
  commonName: "foo2.bar2"
  dnsNames:
  - foo2.bar2
  duration: 720h
  renewBefore: 240h
```

Notas:

- O campo `duration` não deve ser menor que `1h`.
- O campo `duration` deve ser maior que o campo `renewBefore`.
- Os parâmetros que são usados para os campos `duration` e `renewBefore` devem estar no formato ParseDuration do Golang. As unidades de tempo aceitas incluem ["ns", "us" (ou "µs"), "ms", "s", "m", "h"].
- Se você especificar um `duration` menor que tempo padrão `renewBefore` de 30 dias (720 horas), também deverá especificar o campo `renewBefore` que é menor que seu `duration`.

Visualizando recursos do cert-manager do IBM Cloud Private

O serviço do gerenciador de certificado IBM® Cloud Private cria três `CustomResourceDefinitions` (CRD) do Kubernetes:

1. Certificate
2. Issuer
3. ClusterIssuer

É possível usar o comando `kubectl` para visualizar esses recursos do Kubernetes.

- Visualize os certificados em seu cluster
- Visualize Issuers e ClusterIssuers em seu cluster

Visualize os certificados em seu cluster

Para visualizar todos os certificados que são criados e gerenciados pelo serviço do gerenciador de certificados do IBM® Cloud Private, execute o seguinte comando:

```
kubectl get certificate --all-namespaces
```

A saída de comando fornece as seguintes informações:

Cabeçalho	Descrição
Namespace	O namespace no qual este certificado está localizado.
Nome	O nome do certificado.
Pronto	Se o certificado foi emitido com sucesso e está pronto para uso.
Segredo	O nome do segredo que contém o certificado emitido e a chave privada para o certificado.
Idade	Quando o recurso do certificado foi criado pela primeira vez no Kubernetes.
Expiração	Quando o certificado expira.

Para obter mais informações sobre certificados, consulte Sinalizações kubectl.

Por exemplo, um `kubectl describe` fornece mais informações sobre o certificado:

```
kubectl describe certificate <certificate name> -n <certificate namespace>
```

Visualize Issuers e ClusterIssuers no cluster

Para visualizar os Emissores ou ClusterIssuers disponíveis em seu cluster, execute o seguinte comando:

```
# To view all Issuers
kubectl get issuer --all-namespaces

# To view ClusterIssuers
kubectl get clusterissuer
```

Atualizando certificados do cert-manager do IBM Cloud Private

O serviço do gerenciador de certificado do IBM Cloud Private atualiza automaticamente os certificados que vão expirar.

- Atualizando manualmente seus certificados
- Desativar a reinicialização de seu serviço quando um certificado for atualizado

Atualizando manualmente seus certificados

Certificados que são gerados pelo Gerenciador de certificados do IBM Cloud Private podem ser atualizados antes que o Gerenciador de certificados do IBM Cloud Private tente atualizá-los seguindo estas etapas:

1. Determine o nome do Segredo associado ao certificado. Essas informações podem ser localizadas usando o comando `kubectl get certificate` e anotando o namespace e o nome do Segredo próximos do seu certificado.

   ```
   kubectl get certificate --all-namespaces
   ```

2. Exclua o Segredo associado ao certificado que deseja atualizar.

   ```
   kubectl delete secret <secret name> -n <namespace>
   ```

   NOTA: o Gerenciador de certificados do IBM Cloud Private recria o certificado, recria o Segredo para esse certificado e reinicia automaticamente quaisquer Pods associados a qualquer Deployment, StatefulSet e DaemonSet que usa esse certificado.

   NOTA: ao atualizar seu certificado, você reconfigura os valores `duration` e `renewBefore` do novo certificado. A menos que seu novo certificado especifique valores para os parâmetros `duration` e `renewBefore`, os padrões a seguir serão aplicados:

   - Valor `duration` de `90` dias
   - Valor `renewBefore` de `30` dias

3. Espere a reinicialização de todos os serviços que usam esse segredo.

Desativar a reinicialização de seu serviço quando um certificado for atualizado

Ao atualizar um certificado usando o serviço do Gerenciador de certificados do IBM Cloud Private, ele reinicia automaticamente quaisquer Pods associados a qualquer Deployment, StatefulSet e DaemonSet que usa esse Certificado.

É possível desativar esse recurso caso você não deseja que o Gerenciador de certificados do IBM Cloud Private reinicie os Pods associados ao seu Deployment, StatefulSet ou DaemonSet.

Para desativar o recurso, forneça a anotação `certmanager.k8s.io/disable-auto-restart: "true"` na definição do yaml Deployment, StatefulSet ou DaemonSet do Kubernetes. Por exemplo:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  annotations:
    certmanager.k8s.io/disable-auto-restart: "true"
...
```

DISCLAIMER: quando o gerenciador de certificado do IBM Cloud Private reinicia seu serviço, ReplicaSets adicionais são criados. Pode haver ReplicaSets antigos remanescentes. Para minimizar o problema de ReplicaSets extras não usados, configure o `spec.revisionHistoryLimit` em sua Implementação para um número razoável. Se o `spec.revisionHistoryLimit` não estiver configurado, o valor padrão será 10. Para obter mais informações, veja a documentação do Kubernetes.

Usando o Vault para emitir certificados

O Issurr do gerenciador de certificados suporta o uso do servidor HashiCorp Vault para criar e emitir certificados. Para usar o Vault Issuer, deve-se ter configurado um servidor Vault que esteja acessível ao Gerenciador de Certificados.

Importante: o gerenciador de certificados tenta criar certificados com uma expiração de 90 dias ao usar o Vault Issuer, portanto, o *max_ttl* precisa ter pelo menos 90 dias (2160 horas). Não use um *max_ttl* que tenha menos de 30 dias, porque o Gerenciador de Certificados tenta renovar os certificados quando a expiração está dentro de 30 dias.

Configurando o mecanismo de certificado no Vault

Conclua a configuração a seguir em seu servidor Vault:

1. Ative o mecanismo de Segredos de PKI (Certificados) executando o comando a seguir:

   ```
   vault secrets enable pki
   ```

2. Execute o comando a seguir para ajustar o mecanismo secreto para que seus certificados satisfaçam à solicitação de duração de 90 dias do gerenciador de certificados:

   ```
   vault secrets tune -max-lease-ttl=8760h pki
   ```

3. Execute o comando a seguir para criar um certificado CA autoassinado e um par de chaves com uma expiração customizada.

   ```
   vault write pki/root/generate/internal common_name=ibm.com ttl=8760h
   ```

   O comando anterior é usado para a CA raiz, mas pode ser estendido para usar uma CA intermediária. A documentação do produto HashiCorp Vault fornece mais informações sobre este cenário.

4. Configure URLs para criar os certificados com o comando a seguir, em que `vault_server` é o endereço IP do servidor Vault. Execute o comando a seguir:

   ```
   vault write pki/config/urls issuing_certificates="http://<vault_server>:8200/v1/pki/ca"
   crl_distribution_points="http://<vault_server>:8200/v1/pki/crl"
   ```

5. Crie uma função que possa processar as solicitações de assinatura de certificado. Execute o comando a seguir:

   ```
   vault write pki/roles/my-role allowed_domains=ibm.com allow_subdomains=true allow_any_name=true
   allow_localhost=true enforce_hostnames=false max_ttl=8760h
   ```

Depois que o servidor Vault é configurado para emitir certificados, os recursos do Kubernetes podem ser criados que se referem ao servidor Vault para a criação do certificado.

Autenticação com o servidor Vault

O suporte do servidor Vault no Gerenciador de Certificados suporta dois métodos de autenticação. O Emissor do Gerenciador de Certificados deve usar um dos métodos de autenticação a seguir.

- Autenticação do token
- Autenticação de AppRole

  Nota: Se o TLS estiver ativado em seu servidor Vault, inclua o parâmetro `caBundle` na seção `vault`. Configure o valor `caBundle` para o certificado de autoridade de certificação codificado em base64 em formato PEM. Para obter informações adicionais, consulte a documentação do Listener TLS do Servidor Vault HashiCorp.

Crie Emissores usando a Autenticação do Token

O uso do Vault com a autenticação do token requer um Segredo do Kubernetes que contenha o token de autenticação do Vault. O token de autenticação deve ser codificado com `base64` quando armazenado no Segredo. Crie esse recurso no mesmo namespace onde você deseja o Emissor. Se você estiver usando um ClusterIssuer, deverá criar esse Segredo no namespace `kube-system`.

Os tokens podem ser criados usando as APIs do Vault. Lembre-se de que é necessário renovar os tokens que expiram, já que o Gerenciador de certificados não reconhece expirações de token. A documentação do HashiCorp Vault contém mais detalhes sobre a autenticação de token.

Conclua a configuração a seguir em seu servidor Vault:

1. Crie uma política que permita o uso das APIs do PKI Vault. A política deve incluir os recursos a seguir: *create*, *read*, *update*, *delete*, *list* e *sudo*.

   - Crie um arquivo de políticas e nomeie-o como `pki_policy.hcl`. Inclua o seguinte conteúdo em seu arquivo de políticas:

     ```
     path "pki*" { capabilities = ["create", "read", "update", "delete", "list", "sudo"]}
     ```

   - Execute o comando a seguir para criar a política, em que `pki_policy` é o nome da política e `pki_policy.hcl` é o nome do arquivo:

     ```
     vault policy write pki_policy pki_policy.hcl
     ```

2. Crie um token que use a política recém-criada. Execute o comando a seguir, em que `pki_policy` é o nome da política que você acabou de criar. Certifique-se de que a configuração de `ttl` seja maior que o período em que você deseja emitir renovações com relação ao token que você criou:

```
 vault write /auth/token/create policies=<"pki_policy"> no_parent=true no_default_policy=true
renewable=true ttl=767h num_uses=0
```

3. Crie um Segredo do Kubernetes que contenha o token de autenticação codificado em base64. É possível codificar o token com o comando a seguir, em que `abe02917-7494-c94c-a4f1-99890caf06d7` é seu token:

   ```
   echo abe02917-7494-c94c-a4f1-99890caf06d7 | base64
   ```

   Consulte a amostra YAML a seguir, que define um Segredo com um token Vault:

   ```
   apiVersion: v1
   kind: Secret
   type: Opaque
   metadata:
     name: my-vault-token
     namespace: default
   data:
     token: "YWJlMDI5MTctNzQ5NC1jOTRjLWE0ZjEtOTk4OTBjYWYwNmQ3Cg=="
   ```

   Importante: os tokens no Vault expirarão. O gerenciador de certificados não atualiza o token para evitar que ele expire, portanto, certifique-se de endereçar as renovações do token. Um token raiz não expira e também não deve ser usado, exceto em um ambiente de desenvolvimento ou de teste.

   Para renovar o token, execute um comando Vault semelhante ao comando a seguir, em que `abe02917-7494-c94c-a4f1-99890caf06d7` é o seu token:

   ```
   vault write /auth/token/renew token=abe02917-7494-c94c-a4f1-99890caf06d7
   ```

4. Crie o Vault Issuer que usa o Segredo do token do Vault. O Vault Issuer referencia o Segredo recém-criado, a URL para o servidor Vault e o caminho para uma função:

   ```
   apiVersion: certmanager.k8s.io/v1alpha1
   kind: Issuer
   metadata:
     name: my-vault-issuer
     namespace: default
   spec:
     vault:
       path: pki/sign/my-role
       server: http://192.168.44.241:8200
       auth:
         tokenSecretRef:
           name: my-vault-token
           key: token
   ```

5. Crie um certificado que usa o Vault Issuer. O exemplo a seguir define um certificado que usa o Issuer, que é referenciado na etapa anterior:

   ```
   apiVersion: certmanager.k8s.io/v1alpha1
   kind: Certificate
   metadata:
     name: my-cert1-com
     namespace: default
   spec:
     secretName: my-cert1-tls
     issuerRef:
       name: my-vault-issuer
       commonName: myhostname.ibm.com
     dnsNames:
     - myhostname.ibm.com
   ```

Crie emissores usando a autenticação AppRole

O Vault suporta a autenticação do AppRole, que permite que o Gerenciador de Certificados se conecte ao Vault usando um identificador secreto do AppRole em vez de um token. Conclua a configuração a seguir em seu servidor Vault para configurar a autenticação do AppRole.

Consulte a documentação do HashiCorp Vault para obter mais informações.

1. Ative a Autenticação do AppRole com o comando a seguir:

   ```
   vault auth enable approle
   ```

2. Crie uma política que permita o uso das APIs do PKI Vault. A política deve incluir os recursos a seguir: *create*, *read*, *update*, *delete*, *list* e *sudo*.

   - Crie um arquivo de políticas e nomeie-o como `pki_policy.hcl`. Inclua o seguinte conteúdo em seu arquivo de políticas:

     ```
     path "pki*" { capabilities = ["create", "read", "update", "delete", "list", "sudo"]}
     ```

   - Execute o comando a seguir para criar a política, em que `pki_policy` é o nome da política e `pki_policy.hcl` é o nome do arquivo:

     ```
     vault policy write pki_policy pki_policy.hcl
     ```

3. Execute o comando a seguir para criar uma função nomeada que use a política recém-criada:

   ```
   vault write auth/approle/role/my-role secret_id_ttl=8760h token_num_uses=0 token_ttl=20m
   token_max_ttl=30m secret_id_num_uses=0 policies=pki_policy
   ```

   Importante: a expiração de *secret_id_ttl* é semelhante à expiração do token, exceto que o secret_id não pode ser renovado.

4. Obtenha o *role_id* do AppRole com o comando a seguir:

   ```
   vault read auth/approle/role/my-role/role-id
   ```

5. Obtenha o *secret_id* para o AppRole com o comando a seguir:

   ```
   vault write -f auth/approle/role/my-role/secret-id
   ```

6. Em seguida, é necessário criar um Segredo do Kubernetes que contenha o seu secret_id do AppRole codificado em base64. O uso de Vault com autenticação de AppRole requer um Segredo do Kubernetes que contém o `secret_id` do Vault AppRole.

   O `secret_id` do AppRole deve ser codificado em base64 quando armazenado no Segredo. Crie esse recurso no mesmo namespace onde você deseja o Emissor. Se você estiver usando um ClusterIssuer, deverá criar esse Segredo no namespace `kube-system`.

   ```
   echo abe02917-7494-c94c-a4f1-99890caf06d7 | base64
   ```

   Consulte a amostra YAML a seguir, que define um Segredo do Kubernetes que contém o seu AppRole `secret_id` codificado em base64:

   ```
   apiVersion: v1
   kind: Secret
   type: Opaque
   metadata:
    name: cm-vault-approle
    namespace: default
   data:
    secretId: "ZmNlODI5OWUtNjlkNi1hZjY1LWRhYTItYWYxODI4OWZkYjVjCg=="
   ```

7. Em seguida, crie o Vault Issuer que usa o Segredo do AppRole. Edite a amostra YAML a seguir.

   ```
   apiVersion: certmanager.k8s.io/v1alpha1
   kind: Issuer
   metadata:
    name: cm-vault-issuer
    namespace: default
   spec:
    vault:
      path: pki/sign/my-role
      server: http://192.168.230.158:8200
      auth:
        appRole:
          path: approle
          roleId: "92b6d857-6917-b2cf-6a51-94a44989d2b2"
          secretRef:
            name: cm-vault-approle
            key: secretId
   ```

   Aqui, o Vault Issuer do Gerente de Certificados faz referência ao Segredo recém-criado, que contém o `secret_id` da autenticação do AppRole. O Emissor especifica o `role_id`. O servidor Vault e o caminho da URL para processamento de solicitações de assinatura de certificado também são fornecidos.

Importante: ao usar a autenticação AppRole, tenha cuidado para não confundir as duas funções. Uma função é usada para a assinatura do certificado, que é especificada no `vault path` do Emissor. A outra função é para autorizar determinados recursos dentro do AppRole.

8. Crie um certificado que usa o Vault Issuer.

   A amostra a seguir define um certificado que usa o Emissor, que é referenciado na etapa anterior:

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Certificate
metadata:
 name: vault-approle-aludel1-cert
 namespace: default
spec:
 secretName: vault-approle-aludel1-cert-secret
 issuerRef:
   name: cm-vault-issuer
 commonName: myhostname.ibm.com
 dnsNames:
 - myhostname.ibm.com
```

Se o `secret_id_ttl` estiver configurado muito pequeno e se esse valor expirar, um novo `secret_id` deverá ser criado. O Emissor e o Segredo referenciado precisam ser editados para configurar o novo `secret_id`.

Consulte o Usando o Gerenciador de certificador do IBM Cloud Private (cert-manager) para obter mais tópicos do gerenciador de certificado.

Usando o ACME para emitir certificados

Use o protocolo ACME para emitir certificados quando precisar de prova de propriedade de domínio. O emissor ACME HTTP envia uma solicitação de HTTP para os domínios especificados na solicitação de certificado. O servidor ACME espera que uma determinada página da web seja publicada em cada nome de domínio solicitado no certificado. O serviço `cert-manager` publica a página da web esperada criando um pod e um ingresso temporários. Quando a validação estiver concluída, o pod e o ingresso temporários serão limpos. Em seguida, o servidor ACME emite o certificado.

O emissor é usado principalmente com o servidor ACME que está hospedado em letsencrypt.org. Para obter informações adicionais sobre o emissor ACME HTTP e a autoridade de certificação letsencrypt.org, consulte:

- Documentação da autoridade de certificação Let's Encrypt
- Tutorial do emissor ACME HTTP do gerenciador de certificados

O seguinte exemplo de `Emissor` usa um servidor temporário que é fornecido pela Let's Encrypt. Os detalhes importantes são o nome do `servidor` e o endereço de `e-mail`.

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Issuer
metadata:
  name: letsencrypt-staging
  namespace: default
spec:
  acme:
    # The ACME server URL
    server: https://acme-staging-v02.api.letsencrypt.org/directory
    # Email address used for ACME registration
    email: certificates@us.ibm.com
    # Name of a secret used to store the ACME account private key
    privateKeySecretRef:
      name: letsencrypt-staging
    # Enable the HTTP-01 challenge provider
    http01: {}
```

O seguinte exemplo de `Certificado` usa o `Emissor` definido na etapa anterior. Os certificados `commonName` e `dnsNames` são desafiados pelo servidor ACME. O serviço gerenciador de certificados cria automaticamente regras de pod e de ingresso para resolver os desafios para os dois `hostnames` listados no exemplo a seguir.

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Certificate
metadata:
  name: cm-aws-example-com
```

```
  namespace: default
spec:
  secretName: cm-aws-example-com-secret
  issuerRef:
    name: letsencrypt-staging
    kind: Issuer
  commonName: evolving-moray-master.purple-chesterfield.com
  dnsNames:
  - www.evolving-moray.purple-chesterfield.com
  acme:
    config:
    - http01:
        ingressClass: nginx
      domains:
      - www.evolving-moray.purple-chesterfield.com
      - evolving-moray-master.purple-chesterfield.com
```

Cada um dos `hostnames` listados no certificado em `dnsNames` e `commonName` também devem estar presentes na lista de domínios ACME.

O emissor ACME tem as seguintes limitações:

- O keyAlgorithm ECDSA não é suportado.Você deve usar o keyAlgorithm RSA.
- O campo `organização` não é suportado.
- O campo `duração do certificado` não é suportado.
- O Emissor Acme não suporta endereços IP como SANs.
- O nó no qual o cert-manager é executado deve ter acesso de saída à Internet.

Incluindo certificados usando o algoritmo ECDSA para criptografia

Com o gerenciador de certificados, é possível criptografar com o algoritmo ECDSA. Um certificado pode especificar o algoritmo de assinatura de chave e o tamanho da chave. Utilize os parâmetros `keyAlgorithm` e `keySize` para especificar suas customizações para a chave privada. Se esses parâmetros não estiverem configurados, uma chave RSA de 2048 bits será criada.

Criando um certificado com o ECDSA

Para criar um certificado que usa o algoritmo ECDSA para criptografia, siga o procedimento em Criando certificados do gerenciador de certificados IBM Cloud Private (cert-manager), mas use a amostra a seguir em que `keyAlgorithm` e `keySize` são necessários:

```
apiVersion: certmanager.k8s.io/v1alpha1
 kind: Certificate
 metadata:
   name: ecdsa-ca-cert
   namespace: default
 spec:
   secretName: ecdsa-ca-secret
   keyAlgorithm: ecdsa
   keySize: 521
   isCA: true
   issuerRef:
     name: ss-issuer
     kind: Issuer
   commonName: foo1.bar1
   dnsNames:
   - foo1.bar1
```

Usando o ECDSA com Emissores

Emissor de CA

1. Consulte a amostra YAML a seguir, que usa o Gerenciador de Certificados para criar um Emissor de CA que usa ECDSA:

   ```
   apiVersion: certmanager.k8s.io/v1alpha1
   kind: Issuer
   metadata:
   ```

```
 name: ss-issuer
 namespace: default
spec:
 selfSigned: {}
```

2. Em seguida, crie um certificado de autoridade de certificação que seja emitido a partir do Emissor autoassinado e usando o algoritmo de chave ECDSA. Consulte a amostra a seguir:

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Certificate
metadata:
name: ecdsa-ca-cert
namespace: default
spec:
secretName: ecdsa-ca-secret
keyAlgorithm: ecdsa
keySize: 521
isCA: true
issuerRef:
  name: ss-issuer
  kind: Issuer
commonName: foo1.bar1
dnsNames:
- foo1.bar1
```

3. Edite a amostra a seguir para criar o Emissor de CA com o certificado CA:

```
apiVersion: certmanager.k8s.io/v1alpha1
kind: Issuer
metadata:
 name: ecdsa-ca-issuer
 namespace: default
spec:
 ca:
   secretName: ecdsa-ca-secret
```

Vault Issuer

É possível criar certificados a partir de um Vault Issuer que usa chaves ECDSA em vez de RSA configurando a função de atualização do certificado de terminal PKI Vault. Para obter mais informações sobre o Vault Issuer, consulte Incluindo um certificado usando o Vault Issuer.

Acesse seu servidor Vault e execute o comando de configuração a seguir e edite o `key_bits` e o `key_type`:

```
  vault write auth/approle/role/my-role secret_id_ttl=8760h token_num_uses=0 token_ttl=20m
token_max_ttl=30m secret_id_num_uses=0 policies=pki_policy key_type=ec key_bits=521
```

Ao criar recursos de certificado, certifique-se de usar um `keySize` que corresponda ou que seja maior que o `key_bits` que você especificar em seu servidor Vault:

```
  key_type=ec
  key_bits=256, 384 or 521
```

Consulte o Usando o Gerenciador de certificador do IBM Cloud Private (cert-manager) para obter mais tópicos do gerenciador de certificado.

Criando DaemonSets

É possível criar um `DaemonSet` para permitir que cada nó no cluster execute uma cópia de um pod.

Para obter informações adicionais sobre DaemonSets, consulte a página Conceitos do Kubernetes.

Dois formatos estão disponíveis para você criar um DaemonSets na console de gerenciamento.

É possível criar DaemonSets inserindo os valores de parâmetro na janela Criar DaemonSet ou colando um arquivo YAML na janela Criar Recurso.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando DaemonSets na janela Criar DaemonSet

1. No menu de navegação, clique em **Cargas de trabalho** > **DaemonSets**.
2. Clique em **Criar DaemonSet**.
3. Forneça os detalhes do `DaemonSet`. Forneça valores individuais na janela Criar DaemonSet. Deve-se fornecer valores para vários parâmetros:
    - Na guia Geral, forneça esses valores:
        - **Nome** - Um nome para seu `DaemonSet`
    - Na guia Configurações do contêiner, forneça esses valores:
        - **Nome** - Um nome para o contêiner.
        - **Imagem** - a imagem a ser usada para o contêiner. Se você usar uma imagem do registro de imagem privado, deverá fornecer o nome da imagem no seguinte formato. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

          ```
          <cluster_CA_domain>:8500/namespace/imagename
          ```

        - **Protocolo** e **Porta do contêiner** - o protocolo de comunicações e o número da porta para o contêiner.

    Observação: os valores para outros parâmetros são opcionais. Se você não especificar limites de recursos para a CPU, a memória e a GPU, os contêineres poderão usar recursos de cluster ilimitados.
4. Clique em **Criar**.

Criando DaemonSets na janela Criar Recurso

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML na caixa de diálogo Criar Recurso. Para obter mais informações sobre como criar um arquivo YAML DaemonSet, consulte *Gravando uma especificação de DaemonSet* na página Conceitos do Kubernetes.
3. Clique em **Criar**.

Gerenciando implementações

Saiba como atualizar e monitorar suas implementações.

Este guia supõe que os usuários estejam familiarizados com os conceitos e a terminologia do Kubernetes. Termos e componentes chaves do Kubernetes não estão definidos. Para obter mais informações sobre conceitos do Kubernetes, consulte https://kubernetes.io/docs/home/

- Criando implementações
- Modificando uma implementação
- Removendo uma implementação
- Ajuste de escala de implementações

Criando implementações

Saiba como criar e configurar implementações em seu cluster.

Este guia supõe que os usuários estejam familiarizados com os conceitos e a terminologia do Kubernetes. Termos e componentes chaves do Kubernetes não estão definidos. Para obter mais informações sobre conceitos do Kubernetes, consulte https://kubernetes.io/docs/home/

- Criando uma implementação
- Criando uma implementação com recursos de GPU conectados

Criando uma implementação

Crie uma nova implementação em seu namespace.

Novas implementações são designadas ao namespace do usuário que as cria. Quando os administradores criam implementações, eles devem designá-las ao namespace `default`.

Nota: o namespace `default` não deve ser usado no ambiente de produção.

Dois formatos estão disponíveis para você criar uma implementação do console de gerenciamento.

É possível criar implementações inserindo os valores de parâmetro na janela Criar implementação ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando uma implementação usando a janela Criar implementação

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Clique em **Criar implementação**.
3. Forneça os detalhes da implementação. Forneça valores individuais na janela Criar implementação.

   Deve-se fornecer valores para os parâmetros a seguir:

   - Na guia Geral, forneça esses valores:
     - **Nome** - Um nome para sua implementação
     - **Réplicas** - o número de pods ou réplicas. O valor padrão é 1.
   - Na guia Configurações do contêiner, forneça esses valores:
     - **Nome** - Um nome para o contêiner.
     - **Imagem** - a imagem a ser usada para o contêiner. Se você usar uma imagem do registro de imagem privado, deverá fornecer o nome da imagem no seguinte formato. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

       ```
       <cluster_CA_domain>:8500/namespace/imagename
       ```

     - **Protocolo** e **Porta do contêiner** - o protocolo de comunicações e o número da porta para o contêiner.

   Observação: os valores para outros parâmetros são opcionais. Se você não especificar limites de recursos para a CPU, a memória e a GPU, os contêineres poderão usar recursos de cluster ilimitados.
4. (Opcional) Conecte uma solicitação `PersistentVolume` ao contêiner. Para obter mais informações, consulte Anexando PersistentVolumeClaims a uma implementação.
5. Clique em **Criar**.

Criando uma implementação usando a janela Criar recurso

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo **Criar recurso**. Para obter mais informações sobre como criar uma implementação do Kubernetes usando um arquivo YAML, consulte *Criando uma implementação* na página Conceitos do Kubernetes.

   Se desejar usar imagens do registro de imagem privado que estão em um namespace diferente, você deve ter acesso a esse namespace e fornecer o valor `imagePullSecrets` para esse namespace no arquivo YAML. Para obter mais informações, consulte Criando imagePullSecrets para um namespace específico.
3. Clique em **Criar**.

Após a conclusão da implementação, uma nova implementação será exibida na página Implementações. As colunas **DESEJADO**, **ATUAL**, **PRONTO** e **DISPONÍVEL** todas exibem o mesmo valor, que é o número de pods ou a réplica que você especificou durante a implementação.

Clique no nome de implementação para visualizar informações detalhadas sobre a implementação. Revise as propriedades de implementação e assegure-se de que elas sejam exatas.

Importante: Para acessar sua implementação a partir da Internet, você deve expor sua implementação como um serviço. Para obter mais informações, consulte Criando Serviços .

Criando uma implementação com recursos de GPU conectados

Saiba como criar um contêiner com recursos de GPU conectados.

- Para obter uma visão geral do suporte de GPU Nvidia no IBM® Cloud Private, consulte Suporte de GPU Nvidia.
- Assegure-se de que os drivers GPU estejam instalados em nós do trabalhador. É possível usar o comando `kubectl describe nodes` para indicar que o `nvidia.com/gpu` está usando um recurso GPU.

O IBM Cloud Private oferece suporte de GPU integrado para as imagens no projeto nvidia-docker. Para especificar recursos de GPU, suas implementações devem especificar imagens do projeto`nvidia-docker`.

- Para nós do Linux® on Power® (ppc64le), use as imagens no repositório do Docker Hub nvidia/cuda-ppc64le ou imagens que você deriva de conteúdos.
- Para nós do Linux®, use as imagens no repositório do Docker Hub nvidia/cuda ou imagens que você deriva de conteúdos.

Dois formatos estão disponíveis para você criar uma implementação do console de gerenciamento.

É possível criar implementações inserindo os valores de parâmetro na janela Criar implementação ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Antes de iniciar, assegure que os nós estejam prontos para implementação. Para obter informações adicionais, consulte Configurando um nó do trabalhador de GPU.

Problemas e Limitações Conhecidos

O nó IBM® Z (s390x) não suporta GPU.

Se você executar IBM Cloud Private em um ambiente misto que tenha nós Linux® (x86_64), Linux® on Power® (ppc64le) e IBM® Z (s390x), o `nvidia-device-plugin DaemonSet` será executado apenas em nós do cluster Linux® (x86_64) e Linux® on Power® (ppc64le).

Criando uma implementação com recursos de GPU conectados usando a janela Criar implementação

1. No menu de navegação, clique em **Cargas de Trabalho** > **Implementações** > **Criar implementação**.
2. Na guia **Configurações do contêiner**, especifique o número da GPU solicitada para a implementação. Assegure-se de que esse valor seja um número inteiro positivo.
3. Insira todas as outras opções de parâmetros que são necessárias para sua implementação.
4. Clique em **Criar**.

Criando uma implementação usando a janela "Criar recurso"

1. Crie um arquivo `gpu-demo.yaml`. Esse arquivo `gpu-demo.yaml` de amostra cria uma implementação de contêiner com um único recurso de GPU conectado.

   Essa implementação de amostra usa a imagem `nvidia/cuda:7.5-runtime`, que é uma imagem `nvidia-docker` para sistemas Linux®. É possível obter essa imagem do repositório do Docker Hub nvidia/cuda. Para Power Systems, use uma das imagens `nvidia/cuda-ppc64le` que estão disponíveis no repositório do Docker Hub nvidia/cuda-ppc64le.

   ```
   apiVersion: apps/v1beta2
   kind: Deployment
   metadata:
    name: gpu-demo
   spec:
    replicas: 1
    selector:
      matchLabels:
        run: gpu-demo
    template:
      metadata:
        labels:
          run: gpu-demo
      spec:
   ```

```
containers:
- name: gpu-demo
  image: nvidia/cuda:7.5-runtime
  command:
  - "/bin/sh"
  - "-c"
  args:
  - nvidia-smi && tail -f /dev/null
  resources:
    limits:
      nvidia.com/gpu: 1
```

1. No painel, clique em **Criar recurso**.
2. Copie e cole o arquivo `gpu-demo.yaml` na caixa de diálogo "Criar recurso".
3. Clique em **Criar**.

Verifique se o recurso de GPU foi detectado dentro do contêiner

1. Instale a interface da linha de comandos `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Para visualizar uma lista de contêineres em execução, execute este comando:

   ```
   kubectl get pods
   ```

   Na saída retornada, é possível localizar a implementação `gpu-demo`.

3. Acesse os logs para a implementação `gpu-demo`. Por exemplo:

   ```
   kubectl logs gpu-demo-3638364752-zkqel
   ```

   A saída se assemelha ao código a seguir:

```
Tue Feb  7 08:38:11 2017
+-----------------------------------------------------+
| NVIDIA-SMI 352.63     Driver Version: 352.63        |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GT 730      Off  | 0000:01:00.0     N/A |                  N/A |
| 36%   41C    P8    N/A /  N/A |      4MiB /  1023MiB |     N/A      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|    0                  Not Supported                                         |
+-----------------------------------------------------------------------------+
```

Após a conclusão da implementação, uma nova implementação será exibida na página Implementações. As colunas **DESEJADO**, **ATUAL**, **PRONTO** e **DISPONÍVEL** todas exibem o mesmo valor, que é o número de pods ou a réplica que você especificou durante a implementação.

Clique no nome de implementação para visualizar informações detalhadas sobre a implementação. Revise as propriedades de implementação e assegure-se de que elas sejam exatas.

Para acessar sua implementação a partir da Internet, você deve expor sua implementação como um serviço. Consulte Criando serviços.

Modificando uma implementação

Edite as propriedades de uma implementação.

É possível modificar as propriedades de uma implementação editando seu arquivo JSON.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Para a implementação que você deseja modificar, selecione **Ação** > **Editar**. O arquivo JSON da implementação é exibido.

3. Atualize as propriedades.
4. Clique em **Enviar**. A implementação é atualizada. Se você modificou o modelo de pod, um lançamento será iniciado.

Para confirmar as mudanças, revise os detalhes da implementação.

Removendo uma implementação

Remova uma implementação que não é mais necessária.

Quando você remove uma implementação, também remove todos os objetos subjacentes, incluindo ReplicaSets, pods e serviços.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Para a implementação que você deseja remover, selecione **Ação** > **Remover**. A janela "Remover implementação" é exibida.
3. Clique em **Remover**.

A implementação selecionada é removida da lista de implementações e todos os seus objetos subjacentes são excluídos.

Ajuste de escala de implementações

Ajustar manualmente o número de instâncias de implementação.

É possível ajustar manualmente o número de instâncias de implementação.

Nota: para definir uma política que escale automaticamente o número de réplicas de implementação, consulte Gerenciando políticas.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Para a implementação que você deseja escalar, selecione **Ação** > **Escala**. A janela "Implementação de escala" é exibida.
3. Insira o número de pods necessários.
4. Clique em **Implementação de escala**. O número de pods implementados é aumentado ou diminuído para atender ao novo número de pods que você precisa.

Os valores **DESIRED**, **CURRENT**, **READY** e **AVAILABLE** da implementação são atualizados.

Gerenciando liberações do Helm

Visualize a lista de liberações do Helm que você pode gerenciar. É possível ler mais sobre liberações no Glossário do Helm.

- Visualizar liberações do Helm
- Fazer upgrade da versão de liberação do Helm
- Retroceder versão de liberação do Helm
- Excluir liberação do Helm

Visualizar liberações do Helm

Para visualizar as liberações do Helm, conclua as etapas a seguir:

1. No menu de navegação, clique em **Cargas de trabalho** > **Liberações de Helm**.
2. Clique em cada liberação para visualizar os detalhes de todos os recursos que estão disponíveis.

Fazer upgrade da versão de liberação do Helm

É possível fazer upgrade de sua liberação atual do Helm para quaisquer mudanças de versão principal ou secundária da correção que estejam disponíveis. Durante um upgrade, é possível fazer modificações nos parâmetros de configuração de gráficos.

Tabela 1. Caminhos de upgrade de liberação do Helm

Caminhos de Upgrade	Versão Atual	Nova Versão
Correção	1.0.0	1.0.1
Secundário	1.0.0	1.1.0
Grave	1.0.0	2.0.0

Para fazer upgrade de sua versão de liberação do Helm, conclua as etapas a seguir:

1. No menu de navegação, clique em **Cargas de trabalho** > **Liberações de Helm**.
2. Revise a coluna **VERSÃO DISPONÍVEL** para ver se quaisquer atualizações de versão estão disponíveis para as liberações do Helm.
3. Para a liberação do Helm que você deseja atualizar, selecione **Ação** > **Upgrade**.
4. Na opção **Versão**, selecione a versão de atualização principal ou secundária da correção que você deseja.

   Nota: o menu inclui versões da liberação do Helm anteriores à sua versão atual.
5. Selecione quais definições de configuração você deseja usar para o upgrade usando a régua de controle. As opções a seguir estão disponíveis:
   - **Reutilizar valores** Esta configuração retém os valores que foram usados durante o upgrade anterior, onde possível.
   - **Reconfigurar valores** Esta configuração substitui as configurações usadas anteriormente pelos valores padrão que são fornecidos no arquivo yaml.
6. Modifique as definições de configuração individuais conforme necessário. **Dica:** selecione **Todos os parâmetros** para visualizar os parâmetros customizáveis.

   Nota: assegure-se de verificar seus parâmetros de configuração, pois os parâmetros não são validados ao fazer upgrade da liberação do Helm.
7. Clique em **Fazer upgrade**.

Retroceder versão de liberação do Helm

É possível retroceder sua liberação atual do Helm para quaisquer versões antigas da liberação que foi configurada anteriormente.

Para retroceder a versão de liberação do Helm, conclua as etapas a seguir:

1. No menu de navegação, clique em **Cargas de trabalho** > **Liberações de Helm**.
2. Para a liberação do Helm que você deseja retroceder, selecione **Ação** > **Retroceder**.
3. Selecione a versão para a qual você deseja retroceder.
4. Clique em **Recuperar**.

Excluir liberação do Helm

Para excluir uma liberação do Helm, conclua as etapas a seguir:

1. No menu de navegação, clique em **Cargas de trabalho** > **Liberações de Helm**.
2. Para a liberação do Helm que você deseja retroceder, selecione **Ação** > **Excluir**.
3. Clique em **Remover**.

Gerenciando tarefas

Saiba como criar diferentes tipos de tarefas.

Este guia supõe que os usuários estejam familiarizados com os conceitos e a terminologia do Kubernetes. Termos e componentes chaves do Kubernetes não estão definidos. Para obter mais informações sobre os conceitos do Kubernetes, consulte http://kubernetes.io/docs/reference/

- Criando tarefas
- Criando CronJobs

Criando Tarefas

Crie tarefas para assegurar que os pods sejam implementados ou para executar vários pods em paralelo.

Para obter mais informações sobre tarefas em lote, consulte https://kubernetes.io/docs/user-guide/jobs.

Dois formatos estão disponíveis para você criar um aplicativo do console de gerenciamento.

É possível criar tarefas inserindo os valores de parâmetro na janela Criar tarefa ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando uma tarefa usando a janela Criar tarefa

1. No menu de navegação, clique em **Cargas de trabalho** > **Tarefas**.
2. Clique em **Criar tarefa**.
3. Forneça os detalhes da tarefa. Forneça valores individuais na janela Criar tarefa.

   Deve-se fornecer valores para vários parâmetros:

   - Na guia Geral, forneça esses valores:
     - **Nome**
   - Na guia Configurações do contêiner, forneça esses valores:
     - **Nome** - O nome do contêiner.
     - **Imagem** - a imagem a ser usada para os contêineres. Se você usar uma imagem do registro de imagem privado, deverá fornecer o nome da imagem no seguinte formato. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

       ```
       <cluster_CA_domain>:8500/namespace/imagename
       ```

     - **Protocolo** e **Porta** - o protocolo de comunicações e o número da porta para o contêiner.

   Observação: os valores para outros parâmetros são opcionais.
4. (Opcional) Anexe o armazenamento PersistentVolume no contêiner. Consulte Conectando volume a um aplicativo.
5. Clique em **Criar**.

Criando uma tarefa usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar uma tarefa em lote do Kubernetes usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#running-an-example-job.

   Nota: se você desejar usar imagens de seu registro de imagem privado que estiverem em um namespace diferente, deverá ter acesso a esse namespace e fornecer o valor **imagePullSecrets** para esse namespace no arquivo YAML. Consulte Criando imagePullSecrets para um namespace específico.
3. Clique em **Criar**.

Criando CronJobs

As Tarefas Cron são usadas para gerenciar tarefas em lote baseadas em tempo. Essas tarefas são executadas uma vez em um ponto especificado no tempo ou repetidamente em um ponto especificado no tempo.

Para obter mais informações sobre Tarefas Cron, consulte CronJob.

Dois formatos estão disponíveis para você criar um aplicativo do console de gerenciamento.

É possível criar CronJobs inserindo os valores de parâmetro na janela Criar CronJob ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando um CronJob usando a janela Criar CronJob

Crie uma CronJob seguindo as tarefas:

1. No menu de navegação, clique em **Cargas de trabalho** > **Tarefas** e, em seguida, clique na guia **Tarefas Cron**.
2. Clique em **Criar CronJob**.
3. Forneça os detalhes da tarefa. Forneça valores individuais na janela Criar CronJob.
   - Na guia Geral, forneça esses valores:
     - **Nome**
     - **Planejamento** - O planejamento de implementação no formato Cron.
   - Na guia Configurações do contêiner, forneça esses valores:
     - **Nome** - O nome do contêiner.
     - **Imagem** - a imagem a ser usada para os contêineres. Se você usar uma imagem do registro de imagem privado, deverá fornecer o nome da imagem no seguinte formato. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação. Seu comando pode ser semelhante ao seguinte:

       ```
       <cluster_CA_domain>:8500/namespace/imagename
       ```

     - **Protocolo** e **Porta** - o protocolo de comunicações e o número da porta para o contêiner.

       Observação: os valores para outros parâmetros são opcionais.
4. Conecte o armazenamento de PersistentVolume ao contêiner. Consulte Conectando volume a um aplicativo.
5. Clique em **Criar**.

Criando um CronJob usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar um CronJob do Kubernetes usando um arquivo YAML, consulte https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/#creating-a-cron-job.

Nota: se você desejar usar imagens de seu registro de imagem privado que estiverem em um namespace diferente, deverá ter acesso a esse namespace e fornecer o valor **imagePullSecrets** para esse namespace no arquivo YAML. Consulte Criando imagePullSecrets para um namespace específico.

1. Clique em **Criar**.

Criando StatefulSets

Crie pods com identidade de rede e armazenamento assegurada.

Implementações de pod normais são projetadas com um conceito simples de identidade e são tratadas como unidades stateless. Se um pod estiver com mau funcionamento ou for substituído por uma versão mais recente, o sistema removerá o pod mais antigo ou com mau funcionamento. Com a introdução de aplicativos stateful, também conhecidos como StatefulSets, é possível criar aplicativos que têm uma noção mais forte de identidade. Esses aplicativos stateful também são capazes de fornecer armazenamento on demand na nuvem.

Para obter mais informações sobre aplicativos stateful ou StatefulSets, consulte https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/.

Dois formatos estão disponíveis para você criar um StatefulSet por meio da console de gerenciamento.

É possível criar aplicativos stateful inserindo os valores de parâmetro na janela Criar StatefulSet ou colando um arquivo YAML na janela "Criar recurso".

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

Criando StatefulSets usando a janela Criar StatefulSet

1. No menu de navegação, clique em **Cargas de trabalho** > **StatefulSets**.
2. Clique em **Criar StatefulSet**.
3. Forneça os detalhes do aplicativo.

   Deve-se fornecer valores para vários parâmetros:

   - Na guia Geral, forneça esses valores:
     - **Nome** - Um nome para seu StatefulSet.
     - **Nome do serviço** - O serviço que define o acesso aos pods.
     - **Réplicas** - o número de pods ou réplicas. O valor padrão é 1.
   - Na guia Configurações do contêiner, forneça esses valores:
     - **Nome** - O nome do contêiner.
     - **Imagem** - a imagem a ser usada para os contêineres. Se você usar uma imagem do registro de imagem privado, deverá fornecer o nome da imagem no seguinte formato. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

       ```
       <cluster_CA_domain>:8500/namespace/imagename
       ```

     - **Protocolo** e **Porta do contêiner** - o protocolo de comunicações e o número da porta para o contêiner.

`**Observação:** os valores para outros parâmetros são opcionais. Se você não especificar limites de recursos para a CPU, a memória e a GPU, os contêineres poderão usar recursos de cluster ilimitados.`

1. (Opcional) Anexe o armazenamento PersistentVolume no contêiner. Consulte Conectando volume a um aplicativo.
2. Clique em **Criar**.

Criando StatefulSets usando a janela "Criar recurso"

1. No painel, clique em **Criar recurso**.
2. Copie e cole um arquivo YAML ou JSON na caixa de diálogo "Criar recurso". Para obter mais informações sobre como criar um StatefulSet usando um arquivo YAML, consulte https://kubernetes.io/docs/tutorials/stateful-application/basic-stateful-set/#creating-a-statefulset.

   Nota: se você desejar usar imagens de seu registro de imagem privado que estiverem em um namespace diferente, deverá ter acesso a esse namespace e fornecer o valor **imagePullSecrets** para esse namespace no arquivo YAML. Consulte Criando imagePullSecrets para um namespace específico.

3. Clique em **Criar**.

Gerenciando ReplicaSets

Use ReplicaSets para criar e gerenciar o número de pods em uma implementação.

ReplicaSets asseguram que o número correto de pods estejam em execução o tempo todo. Os pods que são gerenciados por ReplicaSets são reprogramados automaticamente quando uma falha ocorre. Essas falhas incluem problemas de nó ou de rede.

ReplicaSets substitui o Controlador de Replicação como o gerenciador de pod. Para obter mais informações sobre o ReplicaSets, consulte https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/.

Para visualizar uma lista de todos os ReplicaSets no cluster, no menu de navegação, clique em **Cargas de trabalho** > **ReplicaSets**. Nessa visualização, também é possível filtrar ReplicaSets por seus namespaces.

Tipo de usuário ou nível de acesso necessário: administrador de cluster ou administrador da equipe

- Criando ReplicaSets
- Escalando um ReplicaSet
- Atualizando um ReplicaSet

Criando ReplicaSets

1. No menu de navegação, clique em **Cargas de trabalho** > **ReplicaSets**.
2. No menu suspenso, selecione um namespace. Se um namespace não for selecionado, o ReplicaSet será criado no namespace `default`.
3. Clique em **Criar ReplicaSet**.
4. Forneça os detalhes para o seu ReplicaSet. Deve-se fornecer valores para vários parâmetros:
5. Na guia Geral, forneça esses valores:
   - **Nome** - Um nome para seu ReplicaSet
   - **Réplicas** - o número de pods ou réplicas. O valor padrão é 3.
6. Na guia Configurações do contêiner, forneça esses valores:
   - **Nome** - Um nome para o contêiner.
   - **Imagem** - a imagem a ser usada para o contêiner. Se você usar uma imagem do registro de imagem privado, deverá fornecer o nome da imagem no seguinte formato. `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

     ```
     <cluster_CA_domain>:8500/namespace/imagename
     ```

   - **Protocolo** e **Porta do contêiner** - o protocolo de comunicações e o número da porta para o contêiner. **Observação:** os valores para outros parâmetros são opcionais.
7. Clique em **Criar**.

Escalando um ReplicaSet

1. No menu de navegação, clique em **Cargas de trabalho** > **ReplicaSets**.
2. Para o ReplicaSet que você deseja escalar, selecione **Ação** > **Escalar**.
3. Especifique o número de pods ou instâncias que você deseja.
4. Clique em **Escalar ReplicaSet**.

Atualizando um ReplicaSet

1. No menu de navegação, clique em **Cargas de trabalho** > **ReplicaSets**.
2. Para o ReplicaSet que você deseja modificar, selecione **Ação** > **Editar**. O arquivo JSON do ReplicaSet é exibido.
3. Atualize as propriedades.
4. Clique em **Enviar**. O ReplicaSet é atualizado.

Serviços com recursos

O IBM® Cloud Private inclui vários serviços com recursos.

Aplicativos em Pacote Configuráveis e Serviços Principais

No Catalog, também é possível ativar instâncias extras de serviços principais, como monitoramento e criação de log. Também é possível importar os aplicativos empacotados que estão disponíveis com sua edição do IBM Cloud Private.

- Aplicativos em pacote configurável
- Serviços principais

Software IBM Enterprise

O IBM Cloud Private Catalog fornece acesso a uma ampla matriz de software IBM Enterprise que são licenciados e distribuídos separadamente do produto IBM Cloud Private. O software IBM Enterprise pode ser acessado diretamente a partir do Catalog ou importado para o Catalog a partir do Passport Advantage.

- IBM API Connect
- IBM App Connect Enterprise

- IBM Cloud App Management
- IBM Aspera CLI
- IBM Cloud Event Management
- IBM Cloud Transformation Advisor
- IBM DataPower Gateway
- IBM Data Server Manager
- IBM DB2
- IBM Event Streams
- IBM Integration Bus
- IBM MobileFirst Platform Foundation
- IBM MQ
- IBM Netcool Operations Insight
- IBM Netcool Operations Insight Probes
- IBM Operational Decision Manager
- IBM PowerAI
- IBM PowerVC FlexVolume Driver
- App de amostra do IBM SDK for Node.js
- IBM Spectrum LSF Community Edition
- IBM Spectrum Symphony
- IBM Voice Gateway
- IBM Watson Compare and Comply: Element Classification
- IBM Watson Explorer
- IBM WebSphere Application Server for IBM® Cloud Private VM Quickstarter
- IBM Workload Automation
- Microclimate

Software livre

Para obter detalhes sobre o software livre, consulte Gráficos do IBM Cloud Private.

- Hazelcast IMDG

Aplicativos em pacote configurável

Alguns pacotes configuráveis do IBM® Cloud Private contêm aplicativos que você pode acessar apenas instalando o gráfico Helm.

Antes de poder acessar esses aplicativos, será necessário torná-los disponíveis no Catalog do IBM Cloud Private. Consulte Instalando o software IBM no IBM Cloud Private.

Para obter mais informações sobre os pacotes configuráveis do IBM Cloud Private, consulte Pacotes configuráveis do IBM Cloud Private.

- Cloud Automation Manager
- Cost and Asset Management
- WebSphere Application Server Liberty
- WebSphere Application Server Network Deployment

Serviços principais

É possível instalar instâncias de alguns dos serviços principais do IBM® Cloud Private após a instalação.

Se você desejar instalar o Kibana ou instâncias adicionais dos serviços de criação de log e monitoramento, será possível implementar instâncias desses gráficos em seu cluster.

- Serviço de criação de log do IBM Cloud Private
- Kibana Reprovado
- Serviço de monitoramento do IBM Cloud Private

Serviço de criação de log do IBM Cloud Private

Por padrão, o IBM Cloud Private usa uma pilha ELK para coletar logs de todos os contêineres no cluster. Também é possível implementar mais pilhas ELK para coletar logs de contêiner a partir de nós do cluster ou de namespaces específicos.

Se você não tiver configurado o fornecimento de armazenamento dinâmico, antes de implementar uma nova instância do serviço de criação de log, deverá criar um PersistentVolume para o serviço. Consulte Conectando volume a um aplicativo.

Para implementar uma nova instância do serviço de criação de log, implemente o gráfico `ibm-icplogging` por meio do Catalog. Consulte Implementando gráficos de Helm no Catalog.

Para obter mais informações sobre a pilha do ELK que o IBM Cloud Private usa, consulte criação de log do IBM Cloud Private.

Kibana

As versões do IBM® Cloud Private anteriores à 3.1.2 permitiam a instalação da criação de log sem o Kibana, que poderia então ser instalada com um gráfico do Helm separado. No IBM Cloud Private versão 3.1.2 e mais recente, você deve usar a instância do Kibana que está instalada com o IBM Cloud Private, se não houver uma versão já em execução a partir do gráfico do Helm.

O gráfico do Helm para `ibm-icplogging-kibana` foi removido do IBM Cloud Private em 08 de março de 2019. A partir dessa data, você deve ativar a instância do Kibana que é fornecida durante a instalação do IBM Cloud Private, se não houver uma versão já em execução a partir do gráfico do Helm.

Atualizando o Kibana para usar a instância integrada do IBM Cloud Private

Se a criação de log do Kibana estiver configurada para usar a versão do gráfico do Helm do Kibana, ela deverá ser reconfigurada para usar a versão instalada concluindo as seguintes etapas:

1. Extraia os parâmetros de gráfico do Kibana existentes para um arquivo chamado `logging-values.yaml` execute o seguinte comando:

   ```
   helm get values logging --tls > logging-values.yaml
   ```

2. Mude o valor de parâmetro `kibana.install` do arquivo `logging-values.yaml` para `true`:

   ```
   kibana:
    install: true
   ```

3. Inclua o repositório interno do Helm do IBM Cloud Private chamado `mgmt-charts` que é gerenciado pela implementação de `mgmt-repo` inserindo os seguintes comandos:

   ```
   export HELM_HOME=~/.helm
   helm init -c --skip-refresh
   helm repo add mgmt-charts https://<cluster_CA_domain>:<cluster Master API Port>/mgmt-
   repo/charts --ca-file $HELM_HOME/ca.pem  --cert-file $HELM_HOME/cert.pem --key-file
   $HELM_HOME/key.pem
     For example, if <cluster_CA_domain> is mycluster.icp and <cluster Master API Port> is 8443.
   helm repo add mgmt-charts https://mycluster.icp:8443/mgmt-repo/charts --ca-file
   $HELM_HOME/ca.pem --cert-file $HELM_HOME/cert.pem --key-file $HELM_HOME/key.pem
   helm repo update
   ```

4. Atualize a instância de criação de log para as novas configurações, executando o seguinte comando:

   ```
   helm upgrade logging mgmt-charts/ibm-icplogging --force -f logging-values.yaml --version
   <version> --timeout 600 --tls
   ```

5. Espere aproximadamente 5 a 10 minutos e a criação de log com o serviço Kibana é iniciada. É possível verificar o status de upgrade do Helm executando o seguinte comando:

   ```
   helm history --tls logging
   ```

Nota: Não é possível mudar a configuração durante o upgrade com a UI do console. Se precisar usar a UI do console, deverá remover completamente e reinstalar a criação de log com a nova configuração.

Implementando o Kibana ao instalar o IBM Cloud Private

1. Abra o arquivo `config.yaml`.

2. Procure o parâmetro `kibana:`.

3. Certifique-se de que o valor esteja configurado como *install: true*.

Serviço de monitoramento do IBM Cloud Private

Por padrão, o IBM® Cloud Private usa uma pilha do Prometheus/Grafana para monitoramento do sistema. Consulte Monitoramento de cluster do IBM Cloud Private para obter mais informações sobre o monitoramento da pilha que é usada no IBM Cloud Private.

Identificando os IBM Certified Containers

Os IBM Certified Containers são identificados no Catalog por um de dois badges com a entrada. Uma entrada com um badge *IBM Certified Container* atende aos critérios para esse badge. Uma entrada que exibe um badge *IBM Certified Container Certificado* indica que ela atende aos requisitos do badge *IBM Certified Container Certificado*, que são mais rigorosos do que o que é necessário para o badge *IBM Certified Container*.

Badge IBM Certified Container:

Um IBM Certified Container atende aos critérios padrão para empacotamento e implementação de software conteinerizado com integrações de plataforma. O badge *IBM Certified Container* () atende aos seguintes critérios:

Protegido:

- Gerencia vulnerabilidades de imagem do contêiner
- Segue uma política de privilégio mínima, os privilégios do documento são necessários
- Proveniente de uma origem bem conhecida e confiável (incluído no ImagePolicy)

Integrado:

- Implementado facilmente por meio da integração do catálogo
- Relata o uso por meio da integração de medição
- Verifica a compatibilidade por meio da integração de teste
- Fornece suporte por plataforma de nuvem, serviços de plataforma e software (para produtos com licenças comerciais com suporte e assinatura)

Ciclo de vida gerenciado:

- Segue o gerenciamento de versão padrão do mercado
- Mantém a moeda com versões da plataforma

Badge IBM Certified Container Certificado:

Um IBM Certified Container pode ser *Certificado* para atender aos requisitos adicionais para soluções de software conteinerizadas de classificação corporativa. O badge *IBM Certified ContainerCertificado* () atende aos seguintes

critérios adicionais:

Critérios seguros aprimorados:

- Suporta considerações de acesso seguro (ingresso)
- Permite o controle de dados sensíveis

Ainda mais integrado:

- Agiliza um cliente para uma topologia de produção por meio de integrações prontas para uso com serviços de gerenciamento adicionais
- Fornece operações de ciclo de vida simples para upgrade e retrocesso por meio de integração com a experiência da plataforma, que são consistentes em todos os IBM Certified Containers.

Critérios de disponibilidade de carga de trabalho aprimorados

- Contém uma topologia de nível de produção que foi gravada pelos especialistas do produto
- Suporta o Kubernetes ou recuperação e failover automáticos de aplicativo customizado

- Fornece resiliência e ajuste de escala

Identificando o IBM Cloud Paks

Um IBM Cloud Pak é um pacote de uma ou mais ofertas IBM Certified Container de classificação corporativa, seguras e gerenciadas pelo ciclo de vida empacotadas e integradas no ambiente do IBM Cloud Private.

Em alguns casos, uma única oferta do IBM Certified Container não pode fornecer toda a funcionalidade que é necessária para resolver requisitos complexos. Talvez você precise de várias ofertas que atendam a um requisito. Algumas das ofertas que são normalmente instaladas juntas são fornecidas como um pacote chamado IBM Cloud Pak.

Um IBM Cloud Pak é categorizado como um *IBM Cloud Pak* ou como um *IBM Cloud Pak* Certificado para a liberação 3.2.0 no Catalog. A designação é baseada no status de certificação da oferta do IBM Certified Container no IBM Cloud Pak com a classificação mais baixa. Se todos os IBM Certified Containers no IBM Cloud Pak forem IBM Certified Containers Certificados, ele é um IBM Certified Container Certificado. Se algum IBM Certified Container empacotado no IBM Cloud Pak não for um IBM Certified Container Certificado, ele será um IBM Cloud Pak. O IBM Cloud Pak está contido na seção *Solution Pak* do Catalog para a liberação 3.2.0. É possível filtrar para exibir apenas *IBM Cloud Paks* ou *IBM Cloud Pak Certificado* selecionando-os no filtro na barra de título.

Exemplos de IBM Cloud Paks incluem as seguintes ofertas:

- IBM Cloud Private para Dados
- Cloud Integration Platform
- Digital Business Automation para MultiCloud

Cloud Automation Manager

O IBM Cloud Automation Manager é uma solução de gerenciamento de nuvem no IBM® Cloud Private para implementar a infraestrutura de nuvem em múltiplas nuvens com uma experiência do usuário otimizada.

Pré-requisitos

Para obter uma lista completa de pré-requisitos, consulte Pré-requisitos para instalação do Cloud Automation Manager.

Sobre o Cloud Automation Manager

O Cloud Automation Manager usa Terraform de software livre para gerenciar e entregar a infraestrutura em nuvem como código. A infraestrutura em nuvem que é entregue como código é reutilizável, é capaz de ser colocada sob controle de versão, pode ser compartilhada entre equipes distribuídas e pode ser usada para replicar os ambientes com facilidade.

A biblioteca de conteúdo do Cloud Automation Manager é preenchida previamente com modelos de amostra para ajudá-lo a iniciar rapidamente. Use os modelos de amostra no estado em que se encontram ou customize-os conforme necessário. Também é possível usar o Cloud Automation Manager para implementar um ambiente de tempo de execução do Chef para configuração e implementação de aplicativo mais avançadas.

Para obter mais informações sobre a instalação do Cloud Automation Manager no IBM Cloud Private, consulte Instalando.

O Cloud Automation Manager é um aplicativo em pacote configurável. Para obter informações sobre aplicativos em pacote configurável, consulte Aplicativos em pacote configurável.

Hazelcast IMDG

O Hazelcast IMDG é um cache na memória distribuído de software livre. É possível usar o cache como um cache de sessão altamente disponível para cargas de trabalho do aplicativo, por exemplo, no IBM® WebSphere Application Server Liberty.

Para obter mais informações sobre o Hazelcast IMDG, consulte a documentação do Hazelcast.

Para começar a instalar e configurar o Hazelcast IMDG no IBM® Cloud Private, consulte o LEIA-ME do gráfico Helm.

Conexão da API

O IBM® API Connect é uma oferta de gerenciamento de API integrada, com recursos e conjunto de ferramentas para todas as fases do ciclo de vida da API, incluindo criação, segurança, gerenciamento, socialização e análise.

Use o serviço do IBM API Connect para criar rapidamente APIs e microsserviços com base nos tempos de execução Node.js e Java. Também é possível gerenciar suas APIs existentes configurando níveis variados de segurança, visibilidade e limites de taxa ao compartilhar APIs com desenvolvedores de aplicativos. Com o serviço do API Connect, é possível transformar e expandir seus negócios com insights por meio de análise detalhada com procuras filtradas estruturadas.

Os componentes de instalação subjacentes são formados por gráficos Helm. Ao configurar o API Connect, assegure-se de que a versão do Helm no servidor esteja no mesmo nível que a versão do Helm que está no cliente.

É possível integrar o IBM API Connect em sua instância Privada do IBM Cloud concluindo o procedimento de instalação do Kubernetes em Instalando o API Connect no ambiente do IBM Cloud Private.

IBM App Connect Enterprise

O IBM App Connect Enterprise combina as tecnologias estabelecidas e confiáveis pelo segmento de mercado do IBM Integration Bus com novas tecnologias nativas para nuvem do IBM App Connect Professional. Ele permite que os negócios aproveitem as arquiteturas orientadas a API, conectem aplicativos baseados em nuvem e façam uso de tecnologias de inteligência artificial (AI) para ampliar o valor e o investimento de seus dados e sistemas existentes.

O App Connect Enterprise fornece um conjunto abrangente de recursos de integração em uma plataforma ágil, segura e de alto desempenho e permite a conectividade universal entre sistemas, aplicativos e dados corporativos. As opções de conectividade e de conjunto de ferramentas no App Connect Enterprise V11 estendem a ampla variedade de formatos de dados e aplicativos que são suportados, incluindo formatos baseados em padrões, como XML, DFDL e JSON, formatos de segmento de mercado e padrões, como HL7, SWIFT e ISO8583. Essa conectividade permite que seus aplicativos interajam e troquem dados com outros aplicativos em uma infraestrutura flexível, dinâmica e extensível. O App Connect Enterprise pode rotear, transformar e enriquecer mensagens de um local para qualquer outro local. Por exemplo, as mensagens podem ser roteadas do remetente ao destinatário com base no conteúdo da mensagem, transformadas de um formato para outro e modificadas ou combinadas, antes de serem entregues.

O App Connect Enterprise V11 também inclui os recursos a seguir:

- Opções de conectividade extensivas entre serviços de nuvem, plataformas de nuvem, SaaS e aplicativos no local existentes
- Ferramentas simples para todos os estilos de usuário, que trabalham em conjunto para expor, orquestrar e curar dados
- Suporte para múltiplos estilos de integração, desde SOA/ESB, API e baseados em microsserviços, para integração orientada a apontar e clicar em evento
- Opções flexíveis para implementação em qualquer sistema em nuvem ou no local para que seja possível executar sua solução de integração próximo aos aplicativos que ele conecta

É possível escolher o IBM App Connect Enterprise para ambientes de produção ou experimentar o IBM App Connect Enterprise for Developers sem encargos para propósitos de desenvolvimento e teste. A edição do desenvolvedor possui todos os recursos ativados, mas está limitada a uma mensagem (transação) por segundo no nível do fluxo de mensagens. Para obter mais informações sobre as edições e requisitos de licença do IBM App Connect Enterprise, consulte IBM App Connect Enterprise - Requisitos de licença.

Para obter informações sobre a instalação do IBM App Connect Enterprise para o IBM Cloud Private, consulte:

- Implementando o IBM App Connect Enterprise no IBM Cloud Private
- Implementando o IBM App Connect Enterprise for Developers para o IBM Cloud Private

IBM Aspera CLI

Com os aplicativos IBM Aspera e ofertas SaaS, é possível explorar os sistemas de arquivos com segurança e mover dados em velocidade máxima, independentemente do tamanho do arquivo, da distância de transferência ou das condições de rede.

A interface da linha de comandos do IBM Aspera (a CLI do Aspera) é usada para transferir arquivos para ou de um Aspera Transfer Server externo. Esse servidor externo pode ser uma implementação existente do Aspera no local ou em nuvem ou uma que o Aspera gerencia como parte de uma assinatura do IBM Aspera on Cloud. Em qualquer um dos casos, a CLI do Aspera permite a transferência de dados rápida, confiável e segura de seu ambiente local para armazenamento em nuvem e data centers privados.

Para obter informações detalhadas, consulte a Documentação da CLI do Aspera.

Para obter informações sobre a instalação da CLI do Aspera no IBM Cloud Private, consulte o gráfico Helm para a CLI do Aspera .

IBM® Cloud App Management

Monitore ambientes de aplicativos em nuvem e no local com o IBM® Cloud App Management. Crie uma ponte entre sua infraestrutura existente e a nuvem.

Crie uma instância de serviço para o IBM Cloud App Management. O IBM Cloud App Management é licenciado por meio do Passport Advantage, portanto, o uso com sua instância de serviço precisa ser coberto com licenças autorizadas.

Alguns recursos-chave do IBM Cloud App Management:

- É implementado em minutos no IBM Cloud Privado com ajuste de escala dinâmico para atender às suas necessidades
- Preserva o investimento do cliente em ofertas de gerenciamento de serviços IBM existentes
- Monitora recursos de infraestrutura
- Atualizações frequentemente para entregar novos recursos

Para obter mais informações sobre como implementar e usar o produto IBM Cloud App Management, consulte o Knowledge Center do IBM Cloud App Management.

IBM Cloud Event Management

Use o IBM® Cloud Event Management para consolidar informações de seus sistemas de monitoramento e resolver problemas. O Cloud Event Management pode receber eventos de várias origens de monitoramento, seja no local ou na nuvem. Os eventos indicam que algo aconteceu em um aplicativo, serviço ou outro objeto monitorado. O Cloud Event Management correlaciona automaticamente os eventos relacionados em um único incidente priorizado.

É possível configurar políticas para ajudar na resolução de problemas, por exemplo, enriquecendo as informações do evento ou escalando incidentes automaticamente. Para iniciar respostas rápidas a incidentes, o Cloud Event Management pode notificar a equipe correta. Ele também pode incluir runbooks para ajudar as equipes a resolver problemas.

Para obter mais informações sobre o Cloud Event Management, consulte o Knowledge Center do Cloud Event Management.

Para obter informações sobre como instalar o Cloud Event Management no IBM® Cloud Private, consulte Implementando no IBM Cloud Private.

IBM Cloud Transformation Advisor

IBM® O Cloud Transformation Advisor é uma ferramenta do desenvolvedor que está disponível sem custo para ajudá-lo a avaliar rapidamente os apps Java™ EE no local para implementação na nuvem.

O aplicativo Transformation Advisor está disponível no IBM® Cloud Private. Ele pode avaliar rapidamente seus aplicativos no local para implementação rápida no WebSphere Application Server e Liberty em ambientes de nuvem pública ou privada.

Para obter mais informações sobre como instalar o Transformation Advisor no IBM Cloud Private, consulte Implementando o Transformation Advisor no IBM Cloud Private.

Para obter mais informações sobre o uso do Transformation Advisor, consulte Usando o Transformation Advisor no IBM Cloud Private.

IBM DataPower Gateway

O IBM® DataPower® Gateway é uma plataforma de gateway único que ajuda a fornecer segurança, controle, integração e acesso otimizado a cargas de trabalho entre vários canais de negócios. Esses canais incluem dispositivo móvel, web, interface de programação de aplicativos (API), arquitetura orientada a serviços (SOA), B2B e nuvem.

O DataPower Gateway fornece um ponto de cumprimento de política convergido para proteger suas cargas de trabalho com políticas de segurança consistentes entre canais, reduzindo seu custo operacional e melhorando a segurança. Usando o DataPower Gateway, é possível expandir rapidamente o escopo de ativos de TI de valor para novos canais, fornecendo acesso para clientes, funcionários e outras partes interessadas a recursos críticos. Para obter mais informações, consulte IBM DataPower Gateway no IBM Knowledge Center.

Para obter mais informações sobre a instalação do DataPower no IBM® Cloud Private, consulte Implementando o DataPower no IBM Cloud Private.

IBM® Data Server Manager

O IBM® Data Server Manager (DSM) ajuda a administrar, monitorar, gerenciar e otimizar o desempenho de bancos de dados Db2 for Linux®, UNIX® e Windows™ e ambientes BigInsights Big SQL. Ele fornece funções semelhantes para bancos de dados DB2 for z/OS. A solução também oferece recursos de gerenciamento corporativo para DB2 on Cloud, DB2 Warehouse on Cloud e DB2 Warehouse Private.

O IBM DSM permite que os administradores de banco de dados (DBAs) e outras equipes de TI gerenciem proativamente o desempenho e evitem problemas antes que eles causem impacto nos negócios. A solução é pronta para nuvem e pode ser implementada de maneira rápida e fácil.

É possível usar o DSM Edition gratuito para gerenciar e monitorar seu DB2. Para obter mais informações, consulte Data Server Manager.

Para obter mais informações sobre a instalação do DSM no IBM Cloud Private, consulte Implementar o DSM no IBM Cloud Private.

IBM DB2

O IBM® DB2® é uma solução de banco de dados que é otimizada para fornecer desempenho líder de mercado em várias cargas de trabalho enquanto reduz os custos de administração, armazenamento, desenvolvimento e servidor.

IBM DB2 é um banco de dados com várias cargas de trabalho que foi projetado para ajudá-lo a desenvolver, testar e construir rapidamente aplicativos para o seu negócio. Projetado para cargas de trabalho operacionais e analíticas, a solução fornece computação na memória e outros recursos para ajudar a assegurar alto desempenho e escalabilidade. A otimização e a compactação de armazenamento podem tornar seus aplicativos mais eficientes com relação ao custo e a ingestão contínua de dados assegura que eles sejam executados na velocidade dos negócios.

Várias edições diferentes do DB2. É possível usar o DB2 Developer-C Edition gratuito para projetar, construir e criar protótipos de aplicativos. Para obter informações adicionais, consulte DB2 for Linux® UNIX® e Windows™.

Para obter mais informações sobre como instalar o Db2 Developer-C Edition no IBM® Cloud Private, consulte Implementar o DB2 no IBM Cloud Private.

Fluxos de Eventos IBM

O IBM Event Streams é uma tecnologia de publicação/assinatura tolerante a falhas e de alto rendimento para construir aplicativos acionados por evento.

O IBM Event Streams é baseado no projeto Apache Kafka de software livre, que é amplamente usado por muitos negócios em todo o mundo. O IBM Event Streams ajuda a iniciar com o Apache Kafka em minutos.

Para obter mais informações, consulte a documentação do IBM Event Streams.

Para obter mais informações sobre como instalar o IBM Event Streams no IBM® Cloud Private, consulte as instruções de instalação do IBM Event Streams.

IBM Integration Bus

O IBM Integration Bus é uma solução de software líder de mercado para integração de aplicativos. Ele fornece um conjunto abrangente de recursos de integração em uma plataforma de alto desempenho ágil e segura e permite a conectividade universal entre sistemas corporativos, aplicativos e dados.

É possível usar o IBM Integration Bus para conectar aplicativos juntos, independentemente dos formatos de mensagens ou protocolos que eles suportam. Essa conectividade significa que seus diferentes aplicativos podem interagir e trocar dados com outros aplicativos em uma infraestrutura flexível, dinâmica e extensível.

O IBM Integration Bus roteia, transforma e enriquece mensagens de um local para qualquer outro local. Ele fornece suporte para muitas operações, incluindo roteamento, transformação, filtragem, enriquecimento, monitoramento, distribuição, coleta, correlação e detecção. Por exemplo, as mensagens podem ser roteadas do emissor para o destinatário com base no conteúdo da mensagem, elas podem ser transformadas de um formato para outro e elas podem ser modificadas ou combinadas, antes serem entregues.

O IBM Integration Bus suporta uma ampla variedade de protocolos, incluindo o WebSphere® MQ, o JMS 1.1 e 2.0, o HTTP e HTTPS, Serviços da Web (SOAP e REST), Arquivo, Enterprise Information Systems (incluindo SAP e Siebel) e TCP/IP. Ele também suporta uma ampla variedade de formatos de dados, incluindo formatos binários (C e COBOL), XML e padrões de mercado (incluindo SWIFT, EDI e HIPAA). Também é possível definir seus próprios formatos de dados.

É possível escolher o IBM Integration Bus for Developers (Developer Edition), que pode ser usado sem encargos para propósitos de desenvolvimento e teste ou o IBM Integration Bus Advanced Edition, que é adequado para um ambiente de produção. O Developer Edition tem todos os recursos ativados, mas é limitado a uma mensagem (transação) por segundo no nível do fluxo de mensagens. O Advanced Edition tem todos os recursos ativados separados do nó SalesforceRequest.

Para obter informações adicionais sobre as edições do IBM Integration Bus e os requisitos de licença, consulte IBM Integration Bus - Requisitos de licença.

Para obter informações sobre a instalação do IBM Integration Bus para IBM® Cloud Private, consulte os tópicos a seguir:

- Implementando o IBM Integration Bus no IBM Cloud Private
- Implementando o IBM Integration Bus for Developers no IBM Cloud Private

IBM MobileFirst Platform Foundation

O IBM® MobileFirst Platform Foundation é uma plataforma integrada que ajuda a estender seus negócios para dispositivos móveis. O IBM MobileFirst Platform Foundation inclui um ambiente de desenvolvimento abrangente, um middleware de tempo de execução otimizado para dispositivo móvel, um armazenamento de aplicativos corporativos privado e um console de gerenciamento e de análise integrado, todos suportados por vários mecanismos de segurança.

Com o IBM MobileFirst Platform Foundation, sua organização pode desenvolver, conectar, executar e gerenciar eficientemente aplicativos móveis (apps) completos que podem acessar os recursos integrais de seus dispositivos móveis de destino. O IBM MobileFirst Platform Foundation pode ajudar a reduzir o tempo para o mercado, o custo e a complexidade do desenvolvimento, além de permitir uma experiência do usuário do cliente e do funcionário otimizada em múltiplos ambientes. Para obter mais informações, consulte IBM MobileFirst Platform Foundation no IBM Knowledge Center. Para obter as informações mais atualizadas sobre o MobileFirst Platform Foundation v8.0, consulte MobileFirst Platform Foundation Developer Center.

O IBM MobileFirst Platform Foundation também possui uma edição de comunidade do IBM Mobile Foundation for Developers 8.0. É possível usar o IBM Mobile Foundation for Developers 8.0 gratuito para desenvolver, testar, avaliar e demonstrar os aplicativos MobileFirst em um ambiente de não produção com o banco de dados Derby integrado. A edição de comunidade também possui o IBM Mobile Foundation Analytics, que fornece uma visualização completa em sua paisagem e infraestrutura do servidor móvel.

Para obter informações sobre como instalar o IBMMobileFirst Platform Foundation no IBM Cloud Private, consulte Configurando o servidor MobileFirst no IBM Cloud Private. Para obter informações sobre como instalar o IBM Mobile Foundation for Developers 8.0 no IBM Cloud Private, consulte Implementando o IBM Mobile Foundation for Developers 8.0 no IBM Cloud Private.

IBM MQ

O IBM® MQ é um middleware de sistema de mensagens que simplifica e acelera a integração de diversos aplicativos e dados de negócios em várias plataformas.

O IBM MQ pode transportar qualquer tipo de dados como mensagens, permitindo que as empresas construam arquiteturas reutilizáveis flexíveis, tais como ambientes de arquitetura orientada a serviços (SOA). Ele funciona com uma ampla faixa de plataformas de computação, aplicativos, serviços da web e protocolos de comunicação para entrega de mensagem segura. O IBM MQ fornece uma camada de comunicações para visibilidade e controle do fluxo de mensagens e dos dados dentro e fora de sua organização. Para obter mais informações, consulte IBM MQ (anteriormente IBM WebSphere MQ).

O MQ fornece uma opção de desenvolvimento grátis e uma opção de implementação paga para teste e produção.

Para configurar o IBM MQ para IBM® Cloud Private, consulte Instalando o IBM MQ no IBM Cloud Private.

IBM Netcool Operations Insight

O IBM® Netcool® Operations Insight fornece a capacidade de monitorar o funcionamento e o desempenho de infraestrutura de TI e rede em ambientes locais, em nuvem e híbridos. Ela também incorpora fortes recursos de gerenciamento de eventos e usa análise de dados de alarme e alerta em tempo real, em conjunto com uma análise de dados históricos mais ampla.

Operations Management

Operations Management é a oferta principal do Netcool Operations Insight fornecida no IBM Cloud Private. O Operations Management alavanca o alarme em tempo real e a analítica de alerta, combinados com analíticas de dados históricos mais amplas. O Netcool Operations Insight é desenvolvido com os recursos de gerenciamento de falhas do IBM Tivoli Netcool/OMNIbus e do Log Analysis, que é a tecnologia de dados big data líder da IBM dentro do IBM Operations Analytics. Essa combinação fornece uma poderosa procura de eventos e análise histórica em uma única solução. O Operations Management integra gerenciamento de infraestrutura e de operações em uma única estrutura entre aplicativos de negócios, servidores virtualizados, dispositivos de rede e protocolos, protocolos de Internet, dispositivos de segurança e dispositivos de armazenamento. Ele inclui os recursos a seguir:

- **Analítica de eventos**: executa análise estatística dos dados do evento histórico do Tivoli Netcool/OMNIbus. É possível usar os resultados de análise sazonal para criar regras de rede, dispositivo ou supressão para reduzir o número de eventos. Também é possível reduzir o número de eventos que são apresentados aos operadores, usando os resultados da análise de eventos relacionada para implementar as regras de correlação do Netcool/Impact para agrupar eventos sob um único pai.
- **Procura de eventos**: aplica os recursos de procura e análise do Operations Analytics - Log Analysis de eventos que são monitorados e gerenciados pelo Tivoli Netcool/OMNIbus.

Para obter mais informações sobre o Operations Management, consulte Sobre o Operations Management.

Para obter mais informações sobre como instalar e configurar o Netcool Operations Insight, consulte Instalando o IBM Cloud Private.

Gerenciamento de Serviço

Opcionalmente, é possível incluir os recursos de Gerenciamento de Serviço no Netcool Operations Insight, instalando e configurando a entrada do catálogo do Netcool Agile Service Manager (ASM). O ASM fornece às equipes de operações a visibilidade e o controle atuais completos sobre a infraestrutura dinâmica e os serviços. Com o ASM, é possível consultar um recurso de rede específico e visualizar uma topologia configurável dele dentro de seu ecossistema de relacionamentos e estados. É possível visualizar essas informações em tempo real ou dentro de um espaço de tempo definido.

Para obter mais informações sobre como instalar e configurar o ASM, consulte Netcool Agile Agile Service Manager Knowledge Center.

Netcool Operations Insight Análises

O IBM® Netcool® Operations Insight integra o gerenciamento de infraestrutura e operações em uma solução única e coerente em aplicativos de negócios, servidores virtualizados, dispositivos de rede e protocolos, protocolos da Internet e dispositivos de segurança e armazenamento.

Com o Netcool Operations Insight, é possível receber e correlacionar dados operacionais, como eventos de diferentes origens, assim seus operadores podem detectar a causa de falhas. Quanto mais rápido eles sabem sobre a causa, mais rápido eles podem restaurar seus sistemas e voltar aos negócios. Para obter uma descrição de um cenário de integração entre o IBM® Cloud Private e o IBM Netcool Operations Insight, consulte Aprenda como o IBM Netcool Operations Insight fornece eficiência operacional e automação para o IBM Cloud Private.

Um dos mecanismos padrão para o IBM Netcool Operations Insight para recuperar dados é por meio do conceito de análises. As análises recuperam eventos de uma origem dedicada, ajustam os dados para que se alinhem ao formato esperado para o OMNIbus, configuram campos customizados e, em seguida, encaminham o evento para o OMNIbus. É possível usar uma configuração de amostra para receber eventos do Logstash e Prometheus por meio da análise do barramento de mensagem do IBM Netcool Operations Insight. Essas duas origens coletam todas as informações chave sobre o funcionamento de seus aplicativos.

As análises para o Netcool Operations Insight fornecem uma opção de desenvolvimento grátis para uso com o IBM® Cloud Private. É possível puxar esta imagem, Imagem do Docker do IBM Tivoli Netcool/OMNIbus Probe for Message Bus x86_64, do Docker Hub e localizar seus gráficos Helm no catálogo padrão do IBM® Cloud Private.

Para obter mais informações sobre como configurar as análises do Netcool Operations Insight para o Logstash e o Prometheus, consulte Coletar e detectar problemas usando o Netcool Operations Insight. Para obter detalhes sobre como configurar IBM Runbook Automations para corrigir problemas, consulte Resolver problemas no IBM Cloud Private usando o IBM Runbook Automation.

IBM Operational Decision Manager

O IBM® Operational Decision Manager (ODM) é uma plataforma para capturar, automatizar e governar decisões de negócios repetidas. É possível identificar situações para formular insights e agir com regras de negócios.

O ODM fornece tecnologia cognitiva que permite que uma empresa responda a dados em tempo real com decisões automatizadas e inteligentes. Os usuários de negócios e TI, da mesma forma, podem gerenciar a lógica da decisão de negócios que é usada pelos sistemas operacionais dentro de uma organização.

O ODM for Developers é um serviço com recursos, que simplifica e acelera o processo de instalação para desenvolvedores que desejam usar a opção de desenvolvimento sem encargos. O ODM também está disponível como uma opção de implementação baseada em taxas para produção no IBM® Cloud Private.

O ODM for Developers fornece uma oferta limitada que está disponível sem encargo e é ideal para ambientes de teste.

Para obter mais informações sobre as edições e os requisitos de licença do ODM, consulte Licenças.

Para configurar o ODM para IBM® Cloud Private, consulte Instalando.

IBM PowerAI

O IBM PowerAI fornece pacotes de software para várias estruturas, bibliotecas e ferramentas de apoio do Deep Learning.

Localize mais informações sobre o IBM PowerAI a partir do Portal do desenvolvedor do IBM PowerAI.

Para obter mais informações sobre como incluir o Gráfico Helm do IBM PowerAI no IBM Cloud Private, consulte LEIA-ME do Gráfico Helm do IBM PowerAI e o guia de ativação do IBM Cloud Private.

Para obter as etapas para instalar o PowerAI Vision com o IBM Cloud Private, consulte Instalando o PowerAI Vision com o IBM Cloud Private.

Driver do IBM PowerVC FlexVolume

O PowerVC pode ser usado como o provedor em nuvem que está hospedando as máquinas virtuais para os nós principal e do trabalhador do IBM® Cloud Private. Com este gráfico Helm do IBM PowerVC FlexVolume Driver (`ibm-powervc-k8s-volume-driver`), ele também pode ser usado para provisionar volumes de armazenamento e para montar armazenamento para contêineres.

O IBM PowerVC Virtualization Center é uma oferta de gerenciamento avançado de virtualização e em nuvem, que é construído no OpenStack, que fornece gerenciamento de virtualização simplificado e implementações na nuvem para máquinas virtuais IBM AIX®, IBM i e Linux® que são executadas no IBM Power Systems.

Para obter informações sobre a instalação e o uso do IBM PowerVC FlexVolume Driver, consulte Usando o armazenamento do PowerVC com o IBM Cloud Private.

Aplicativo de amostra do Node.js

Um aplicativo de amostra é fornecido para mostrar como é possível criar, implementar e monitorar um aplicativo Node.js no IBM® Cloud Private.

O aplicativo de amostra foi criado usando o IBM Cloud Developer Tools. O monitoramento é fornecido pelos seguintes módulos npm, que são incluídos no aplicativo:

- `appmetrics`, que monitora os dados de monitoramento e `appmetrics-dash`, que fornece uma visualização desses dados.

- `appmetrics-prometheus`, que fornece dados de monitoramento para visualização com o Prometheus. É possível implementar o Prometheus em seu cluster e, em seguida, usá-lo para monitorar aplicativos Node.js em execução em potencialmente muitos pods dentro do Kubernetes.

É possível instalar o aplicativo de amostra do Node.js por meio do Catalog: procure `nodejs-sample`. Após a implementação, inicie o aplicativo para visualizar as páginas da web que fornecem informações adicionais sobre o aplicativo.

O aplicativo de amostra é executado no tempo de execução do IBM SDK for Node.js. Para obter mais informações sobre o SDK, consulte IBM SDK for Node.js.

IBM Spectrum LSF Community Edition

O IBM Spectrum LSF Community Edition é uma edição de comunidade gratuita da plataforma de gerenciamento de carga de trabalho do IBM Spectrum LSF.

O IBM Spectrum LSF é um sistema de gerenciamento de carga de trabalho poderoso para ambientes de computação distribuída. O IBM Spectrum LSF fornece um conjunto abrangente de recursos inteligentes de planejamento, orientados por políticas, que permitem usar todos os seus recursos de infraestrutura de computação e assegurar o desempenho ideal do aplicativo.

O IBM Spectrum LSF possui recursos robustos de gerenciamento de aplicativos de carga de trabalho que são acessíveis usando o design mais recente nas interfaces baseadas na web, tornando-o poderoso e simples de usar. Para aplicativos que requerem MPI, a biblioteca de MPI da Plataforma acelera e escala aplicativos HPC para tempo de resolução mais curto.

A assistência da comunidade está disponível por meio dos fóruns do LSF no IBM DeveloperWorks.

Nota: para instalar e executar o aplicativo IBM Spectrum LSF Community Edition, deve-se ter um host com um núcleo de soquete duplo ou núcleos únicos. Se os núcleos únicos forem usados, um máximo de 10 será suportado.

IBM Spectrum Symphony

O IBM® Spectrum Symphony é um gerenciador de carga de trabalho de classe corporativa para aplicativos com processamento intensivo de cálculo e dados em uma grade escalável e compartilhada. Ele fornece um ambiente de computação de análise de dados eficiente que acelera dezenas de aplicativos paralelos distribuídos para obter resultados mais rápidos e uma melhor utilização de recursos.

É possível implementar o IBM Spectrum Symphony como um Gráfico Helm no IBM Cloud Private para configurar e executar rapidamente o IBM Spectrum Symphony como um aplicativo de contêiner do Docker em um cluster do Kubernetes. Em seguida, é possível gerenciar o aplicativo IBM Spectrum Symphony a partir do console de gerenciamento de cluster ou da linha de comandos no IBM Cloud Private. Consulte Introdução ao IBM Spectrum Symphony para obter informações adicionais sobre o IBM Spectrum Symphony.

Dica: O IBM Spectrum Symphony está disponível com o IBM Cloud Private como um Community Edition gratuito, gerenciado pelo cliente. O Community Edition fornece a funcionalidade integral do IBM Spectrum Symphony, mas apenas para um cluster de até 64 núcleos. Para escalar o cluster além de 64 núcleos e receber o Suporte IBM ligado ao software licenciado, você deve fazer upgrade de sua autorização para uma versão licenciada do IBM Spectrum Symphony.

Para obter informações sobre como implementar o IBM Spectrum Symphony no IBM Cloud Private, consulte Implementando o IBM Spectrum Symphony no IBM Cloud Private.

IBM Voice Gateway

O IBM® Voice Gateway fornece uma maneira de integrar um conjunto de serviços orquestrados do Watson a uma rede telefônica pública ou privada usando o Protocolo de Inicialização de Sessão (SIP).

Com Voice Gateway, é possível configurar um agente de autoatendimento cognitivo com a inteligência artificial do Watson™ em seu backbone, de modo que o agente possa se comunicar diretamente com clientes e manipular interações complexas que são difíceis para sistemas de resposta de voz interativa (IVR) tradicionais. Também é possível configurar o Voice Gateway como um assistente de agente, que pode transcrever uma ligação telefônica entre um responsável pela chamada do cliente e o agente responsável pelo atendimento ao cliente para que a conversa possa ser processada com análise para feedback do agente em tempo real. Para obter mais informações, consulte a documentação do IBM Voice Gateway.

Ao implementar o Voice Gateway no IBM Cloud Private, é possível configurar rapidamente um agente de autoatendimento ou assistente do agente na nuvem privada. Quando o Voice Gateway é implementado, ele se conecta a outros serviços externos e componentes de rede, como os serviços do Watson e um provedor SIP ou infraestrutura do centro de contato. O Voice Gateway é fornecido com uma licença para desenvolvimento e uso de produção limitado, a qual pode ser atualizada para uma licença para um ambiente de produção completo.

Para obter informações sobre como configurar e implementar o Voice Gateway no IBM Cloud Private, consulte Implementando o Voice Gateway no IBM Cloud Private.

IBM Watson Compare and Comply: Element Classification

O IBM Watson™ Compare and Comply: Element Classification é uma solução para analisar programaticamente a densa linguagem jurídica em documentos jurídicos, tais como contratos.

Com o Compare and Comply: Element Classification é possível analisar documentos do governo rapidamente para converter, identificar e classificar elementos importantes. Usando o Processamento de Linguagem Natural de última geração, o serviço extrai elementos do documento, incluindo:

- `party` (a quem o elemento se refere)
- `nature` (tipo de elemento)
- `category` (classe específica)

O serviço do Compare and Comply: Element Classification fornece os recursos a seguir:

- Entendimento da língua natural dos contratos
- A capacidade de converter PDF programático em JSON anotado
- Identificação de pessoas jurídicas e categorias que se alinham ao conhecimento do assunto

O Compare and Comply: Element Classification fornece um conjunto rico de funcionalidades de APIs do Watson integradas e automatizadas para inserir um PDF programático para identificar o seguinte:

- Seções
- Listas (numeradas e marcadas)
- Notas de Rodapé
- Tabelas (Tables)

O serviço converte esses itens em um formato HTML estruturado. Além disso, o serviço classifica esse formato estruturado, anota-o e o envia como JSON com elementos, tipos e categorias rotulados.

Para obter mais informações sobre o Compare and Comply: Element Classification, consulte a documentação do produto.

IBM Watson Explorer

O IBM Watson Explorer é uma solução analítica de conteúdo cognitivo que ajuda a fazer drill through de todos os seus dados, incluindo informações não estruturadas, para localizar as percepções profundas que você está procurando. O Watson Explorer ajuda você a incrementar todos os seus dados de três maneiras de alto nível.

- Explorar: o Watson Explorer fornece múltiplas maneiras de organizar visualmente e navegar em dados não estruturados, oferecendo aos usuários uma exploração mais rápida e completa de suas informações.
- Analisar: o Watson Explorer Content Miner permite que os usuários limitem rapidamente o foco de seus dados não estruturados, eliminando resultados estranhos e entregando informações específicas e relevantes por meio do fácil monitoramento de anomalias estatísticas.
- Aviso: o Watson Explorer usa o poder de aprendizado de máquina para iluminar padrões, descobertas e insight em todos os seus dados. O Watson Explorer age como seu próprio assistente cognitivo que direciona você para as informações mais relevantes.

Para obter mais informações sobre o IBM Watson Explorer, consulte a documentação do produto

WebSphere Application Server para IBM® Cloud Private VM Quickstarter

O IBM WebSphere Application Server for IBM® Cloud Private VM Quickstarter fornece a experiência do WebSphere no local familiar em um serviço gerenciado no IBM® Cloud Private. O serviço acelera a implementação do aplicativo permitindo que você provisione

ambientes pré-configurados do WebSphere Application Server em máquinas virtuais no IBM Cloud Private. Como resultado, você melhora a produtividade e aperfeiçoa o fornecimento, o teste e a implementação.

O WebSphere Application Server VM Quickstarter é uma parte fundamental do processo de modernização do aplicativo de refatoração de aplicativos monolíticos para uma arquitetura baseada em microsserviço. Devido à sua fidelidade com o ambiente no local, é possível migrar e reutilizar os aplicativos existentes do WebSphere no IBM Cloud Private. À medida que você refatora gradualmente esses aplicativos monolíticos nos microsserviços Liberty baseados em contêiner, seus aplicativos anteriores e novos microsserviços podem coexistir em uma única plataforma.

O WebSphere Application Server for IBM Cloud Private VM Quickstarter é um componente sem encargos do IBM Cloud Private. Para obter informações sobre como instalar e configurar o WebSphere Application Server VM Quickstarter, consulte a documentação do produto. Observe que, para usar o serviço, deve-se ter licenças para outros produtos, incluindo o WebSphere Application Server, o Cloud Automation Manager e o VMware ESXi e vSphere.

Depois que o WebSphere Application Server VM Quickstarter é configurado no IBM Cloud Private, é possível iniciar a criação de instâncias de serviço do WebSphere Application Server. Para obter mais informações, consulte Configurando o WebSphere Application Server no IBM Cloud Private com o WAS VM Quickstarter.

IBM Workload Automation

O IBM® Workload Automation é uma solução completa e moderna para gerenciamento de carga de trabalho em lote e em tempo real. Ele permite que as organizações tenham total visibilidade e controle sobre cargas de trabalho assistidas ou não assistidas. De um único ponto de controle, ele suporta várias plataformas e fornece integração avançada com aplicativos corporativos, incluindo aplicativos ERP, Business Analytics, File Transfer, Big Data e Cloud.

Agora é possível usar o IBM Cloud Private para implementar facilmente um ou mais dos pacotes fornecidos prontos para utilização. Os pacotes a seguir são fornecidos:

- IBM Workload Automation Server, correspondente aos componentes Master Domain Manager e Backup Master Domain Manager
- IBM Workload Automation Console, correspondente ao Dynamic Workload Console
- IBM Workload Automation Agent, correspondente ao componente Dynamic Agent

Para obter informações adicionais sobre o IBM Workload Automation, consulte Documentação do IBM Workload Automation V9.5.

Para instalar e configurar o IBM Workload Automation para IBM® Cloud Private, consulte Implementando o IBM Workload Automation no IBM Cloud Private.

Aplicativo de amostra do Swift

Um aplicativo de amostra é fornecido para mostrar como é possível criar, implementar e monitorar um aplicativo Swift no IBM® Cloud Private.

O aplicativo de amostra foi criado usando o Kitura, uma estrutura da web de alto desempenho e simples de usar para construir aplicativos Swift. Consulte kitura.io para obter informações sobre o Kitura, incluindo mais amostras, tutoriais e postagens de blog.

Ele inclui um terminal de verificação de funcionamento acessível em `/health` e a capacidade de monitorar as métricas do aplicativo no terminal /metrics.

É possível instalar o aplicativo de amostra do Swift por meio do Catalog: procure por `swift-sample`. Após a implementação, inicie o aplicativo para visualizar as páginas da web que fornecem informações adicionais sobre o aplicativo.

Microclimate

Microclimate é um ambiente de desenvolvimento Dockerized de ponta a ponta. Gere microsserviços em Java, Nó e Swift, edite-os e veja suas mudanças imediatamente. Quando você estiver satisfeito com o código, efetue check-in dele e o pipeline construirá e implementará automaticamente no Kubernetes.

O Microclimate não é apenas outro IDE amigável!. O Microclimate inclui ferramentas de monitoramento de aplicativo e um driver de carregamento HTTP para iniciar. Cada uma dessas ferramentas é pré-instalada e pré-configurada para seu serviço para que seja

possível alternar as guias e continuar. Você não precisa mais localizar outras ferramentas ou tentar configurá-las quando tem problemas.

O Microclimate permite que você crie seus microsserviços e a construa de forma automatizada no Docker imediatamente. Não há mais tempo gasto tentando recriar problemas que acontecem em uma máquina ou diferenças quando você está alternando para contêineres em produção. O ambiente é construído usando o Docker também. É possível executá-lo localmente ou hospedá-lo em sua plataforma de nuvem e usar na web. Sem mais problemas com configurações locais ou instaladores para múltiplas ferramentas.

Para obter mais informações sobre o Microclimate, consulte o website Microclimate no qual é possível fazer download do Microclimate, ler informações de apoio, visualizar vídeos instrutivos, obter acesso ao canal Slack e muito mais.

Para obter mais informações sobre como instalar o Microclimate do Catalog, consulte Instalando o Microclimate no IBM Cloud Private.

MongoDB

O MongoDB é um banco de dados orientado por documentos de plataforma cruzada. Classificado como um banco de dados NoSQL, o MongoDB evita a tradicional estrutura de banco de dados relacional baseado em tabela em favor de usar documentos semelhantes a JSON com esquemas dinâmicos. Essa estrutura torna a integração de dados em determinados tipos de aplicativos mais fácil e mais rápida.

É possível usar o MongoDB Community Edition para projetar, construir e criar protótipos de aplicativos. Para obter mais informações, consulte MongoDB.

Para obter mais informações sobre como instalar o MongoDB Community Edition no IBM® Cloud Private, consulte Implementar o MongoDB no IBM Cloud Private.

PostgreSQL

PostgreSQL é um banco de dados objeto-relacional de software livre.

Para obter mais informações sobre como instalar o PostgreSQL no IBM® Cloud Private, consulte Implementar o PostgreSQL no IBM Cloud Private

Skydive

O Skydive é um analisador de topologia de rede e protocolos de software livre em tempo real. Ele fornece uma maneira abrangente de entender o que está acontecendo na infraestrutura de rede. Os agentes do Skydive coletam informações de topologia e fluxos e os encaminha para um analisador central para análise adicional.

É possível usar o Skydive para depurar, solucionar problemas e explorar a rede do IBM® Cloud Private. Para obter mais informações, consulte Docs do Skydive.

Para obter mais informações sobre como instalar o Skydive IBM Cloud Private, consulte Implementar o Skydive no IBM Cloud Private.

WebSphere Application Server Liberty

Ao combinar o IBM WebSphere Application Server Liberty com a paleta completa de tecnologias que estão disponíveis no IBM® Cloud Private Catalog, é possível reduzir o esforço necessário para integrar o middleware, como bancos de dados, soluções de armazenamento em cache e soluções de sistema de mensagens.

Para executar aplicativos Liberty no IBM® Cloud Private, você deve primeiro instalar o aplicativo em um contêiner do Liberty Docker e, em seguida, configurar seu gráfico do Helm.

Quando você estiver pronto para configurar um ambiente de integração contínua para seu aplicativo Liberty, explore configurando o Microclimate. O Microclimate fornece um pipeline do DevOps de ponta a ponta que automatiza as etapas para construir, implementar e gerenciar seus aplicativos Liberty no IBM® Cloud Private. Para obter mais informações sobre como implementar aplicativos Liberty no IBM Cloud Private com o Microclimate, consulte Executando aplicativos Liberty no IBM Cloud Private.

O WebSphere Application Server Liberty é um aplicativo em pacotes configuráveis. Para obter informações sobre aplicativos em pacote configurável, consulte Aplicativos em pacote configurável.

WebSphere Application Server Network Deployment

O WebSphere Application Server Network Deployment fornece um ambiente de tempo de execução do servidor seguro e flexível para implementações de aplicativos essenciais em grande escala. Ele está disponível no local ou para a nuvem pública, privada ou híbrida. Se você estiver buscando reduzir custos, mobilizar novo valor de seu investimento no WebSphere ou acelerar o tempo para lançamento no mercado, esse produto é o mais adequado para cada necessidade de negócios.

O WebSphere Application Server Network Deployment é um aplicativo em pacotes configuráveis. Para obter informações sobre aplicativos em pacote configurável, consulte Aplicativos em pacote configurável.

Recursos e benefícios do WebSphere Application Server

- Otimize sua infraestrutura de aplicativo para reduzir custos com recursos híbridos que fornecem a flexibilidade para implementar e gerenciar aplicativos em qualquer nuvem e qualquer serviço de contêiner.
- Conecte aplicativos Java™ existentes na nuvem e mobilize novo valor com o gerenciamento de ciclo de vida da API e serviços de nuvem, como o IBM Watson ou o IBM Cloud Product Insights.
- Crie e implemente aplicativos e microsserviços da nuvem nativa e baseados na web rapidamente com um tempo de execução de produção leve e combinável que apresenta um único console administrativo para aplicativos e APIs Java e Node.js.
- Ative na nuvem e obtenha suporte para as estruturas Java Platform, Standard Edition 8 e Java Platform, Enterprise Edition 7.

Implementando aplicativos do WebSphere no Kubernetes

A implementação de seus aplicativos do WebSphere no Kubernetes requer que você refatore seus aplicativos monolíticos gradualmente para uma arquitetura baseada em microsserviço. À medida que você refatora seu código, é possível usar o padrão de transformação, coexistência e eliminação do Strangler Application para elevar e deslocar seu aplicativo para máquinas virtuais (VMs) no IBM® Cloud Private.

O pacote configurável Application Modernization contém vários produtos que facilitam a transição de uma implementação no local tradicional para um ambiente de nuvem particular baseado em Kubernetes. Use o WebSphere Connect com o API Connect para expor partes de seus aplicativos e dados como APIs. À medida que você substitui partes de seus aplicativos monolíticos por microsserviços, use o Cloud Automation Manager com o WebSphere Application Server VM Quickstarter para provisionar seu aplicativo mais antigo em uma MV que compartilha infraestrutura de rede com seu novo aplicativo no Kubernetes. Ao usar esse modelo de implementação, os dois aplicativos interoperam de maneira limpa e segura.

Os dois aplicativos coexistem até que seu aplicativo baseado em microsserviço fique totalmente pronto para ultrapassar o antigo. Nesse ponto, enquanto seu novo aplicativo é implementado no Kubernetes, o código de gerenciamento de sua implementação tradicional do WebSphere pode continuar executando sem alterações em uma VM do Cloud Automation Manager.

Use a visão geral do processo a seguir como um guia para modernizar seus aplicativos do WebSphere para execução no Kubernetes:

1. Use o WebSphere Application Server for IBM Cloud Private VM Quickstarter para implementar seus clusters do WebSphere atuais em MVs no IBM Cloud Private. Ao implementar com o WebSphere Application Server VM Quickstarter, é possível migrar facilmente seu cluster usando as ferramentas de migração de nuvem do WebSphere e continuar usando seu código e scripts de gerenciamento existentes.
2. Use o WebSphere Connect para criar APIs que possam acessar uma parte do código do aplicativo em sua forma atual. Como o pacote configurável do IBM Cloud Private Enterprise tem o WebSphere e o API Connect, é possível criar novas APIs seguras por meio de aplicativos existentes. Para obter mais informações, consulte os tutoriais do WebSphere Connect developerWorks.
3. Faça a reengenharia do código existente como microsserviços removendo partes de código, uma parte por vez. Use as APIs do WebSphere Connect para chamar entre o código antigo e o novo. Depois de criar a camada de API sobre o código existente, use a malha e o pipeline Microclimate para implementar essas APIs por meio de microsserviços nativos de nuvem menores que são implementados no Kubernetes no IBM Cloud Private. Com essa abordagem, você começa com a execução

de aplicativos monolíticos no WebSphere sobre VMs e move gradualmente cada parte do código para microsserviços em seu novo aplicativo no Kubernetes.

4. Quando seu novo aplicativo baseado em microsserviço puder preencher com êxito os requisitos de produção, sua contraparte existente poderá ser aposentada.

Ambientes Suportados

O IBM Cloud Private é uma solução de nuvem turnkey e uma solução de nuvem turnkey no local.

O IBM Cloud Private entrega o Kubernetes de envio de dados puro com os componentes de gerenciamento típicos que são necessários para executar cargas de trabalho corporativas reais. Essas cargas de trabalho incluem gerenciamento de funcionamento, gerenciamento de log, trilhas de auditoria e medição para rastreamento de uso de cargas de trabalho na plataforma. Para obter mais projetos, consulte Arquitetura de referência para o IBM Cloud Private.

IBM Cloud Private e Terraform

Os módulos a seguir estão disponíveis, nos quais é possível implementar o IBM Cloud Private Private usando o Terraform:

- Módulo Terraform: Implementar o IBM Cloud Private em qualquer fornecedor de infraestrutura suportado
- IBM Cloud: Implementar o cluster do IBM Cloud Private no IBM Cloud
- VMware: Implementar o IBM Cloud Private no VMware
- AWS: Implementar o IBM Cloud Private no AWS
- OpenStack: Implementar o IBM Cloud Private no OpenStack
- Azure: Implementar o IBM Cloud Private no Azure

IBM Cloud Private no VMware

É possível instalar o IBM Cloud Private no VMware com imagens Ubuntu ou RHEL. Para obter detalhes, consulte os projetos a seguir:

- Instalando o IBM Cloud Private com o Ubuntu
- Instalando o IBM Cloud Private com o Red Hat Enterprise

O serviço IBM Cloud Private Hosted implementa automaticamente o IBM Cloud Private Hosted em suas instâncias do VMware vCenter Server. Esse serviço traz o poder de microsserviços e contêineres para seu ambiente VMware no IBM Cloud. Com esse serviço, é possível estender o mesmo modelo e ferramentas operacionais VMware e do IBM Cloud Private do local para o IBM Cloud.

Para obter mais informações, consulte Serviço IBM Cloud Private Hosted.

IBM Cloud Private no VirtualBox

Para instalar o IBM Cloud Private em um ambiente do VirtualBox, consulte Instalando o IBM Cloud Private no VirtualBox.

IBM Cloud Private com o Red Hat OpenShift

É possível implementar os contêineres de software certificados do IBM que estão em execução no IBM Cloud Private no Red Hat OpenShift.

IBM Cloud Private on AWS

É possível implementar o IBM no Amazon Cloud Services (AWS) usando o CloudFormation ou o Terraform.

IBM Cloud Private no Azure

É possível ativar o Microsoft Azure como um provedor em nuvem para a implementação do IBM Cloud Private e aproveitar todos os recursos do IBM Cloud Private na nuvem pública do Azure.

- IBM Cloud Private with OpenShift
- IBM Cloud Private no AWS
- IBM Cloud Private no Azure

IBM Cloud Private with OpenShift

O IBM® e o Red Hat fizeram parcerias para fornecer uma solução conjunta que usa o IBM Cloud Private e o OpenShift. É possível implementar contêineres de software certificados pelo IBM em execução no IBM Cloud Private com o Red Hat OpenShift.

Ao instalar o IBM Cloud Private with OpenShift, o IBM Cloud Private fornece experiência, gerenciamento e operações para aplicativos do IBM Cloud Private e usa o registro do Kubernetes e do Docker OpenShift que já é instalado pelo Red Hat.

Semelhante ao IBM Cloud Private, o OpenShift é uma plataforma de contêiner que é construída sobre o Kubernetes. É possível instalar o IBM Cloud Private with OpenShift usando o instalador do IBM Cloud Private para OpenShift.

Recursos de integração

- Suporta a plataforma Linux® x86_64 no modo de instalação apenas off-line
- Console de gerenciamento de cluster do IBM Cloud Private integrado e Catálogo
- Serviços de Plataforma principal integrada, como monitoramento, medição e criação de log
- O IBM Cloud Private usa o registro de imagem do OpenShift

Essa integração é padronizada para usar o Open Service Broker no OpenShift. Os brokers que são registrados no OpenShift ainda são reconhecidos e podem contribuir para o IBM Cloud PrivateCatalog. O IBM Cloud Private também é configurado para usar o Servidor da API Kube do OpenShift.

Notas:

- O IBM Cloud Private Vulnerability Advisor (VA) e a criação de log de auditoria não estão disponíveis no OpenShift
- Nem todas as opções de comando da CLI, por exemplo, todos os comandos `cloudctl cm`, são suportadas

Segurança

A administração de autenticação e autorização ocorre somente do IBM Cloud Private para o OpenShift. Se um usuário for criado no OpenShift, o usuário não estará disponível no IBM Cloud Private. A autorização é manipulada pelos serviços do IBM Cloud Private IAM que se integram com o RBAC OpenShift.

Suporte

Se você precisar de suporte, entre em contato com o IBM ou com o Red Hat, dependendo do local em que o problema ocorreu. O IBM e o Red Hat definiram um modelo de suporte colaborativo e trabalharão juntos para problemas complexos que envolvem ambas as equipes de suporte.

O administrador de cluster do IBM Cloud Private é criado no OpenShift durante a instalação. Todos os outros usuários e grupos de usuários do IBM Cloud Private LDAP são criados dinamicamente no OpenShift quando os usuários chamam qualquer API Kube pela primeira vez. As funções para todos os usuários e grupos de usuários do IBM Cloud Private são mapeadas para funções equivalentes do OpenShift. Os tokens que são gerados pelo IBM Cloud Private são aceitos pelo servidor da API Kube OpenShift, pela interface com o usuário do OpenShift e pela CLI do OpenShift.

Monitoramento no OpenShift

O OpenShift fornece um componente de monitoramento opcional baseado em Prometheus, mas não fornece os mesmos recursos que o serviço de monitoramento do IBM Cloud Private. Ao instalar o IBM Cloud Private no OpenShift, o serviço de monitoramento do IBM Cloud Private é instalado por padrão. É possível desativar o serviço de monitoramento. Para obter informações adicionais, consulte a seção *Gerenciando painéis do Grafana* na página de monitoramento do IBM Cloud Private.

Se o IBM Multicloud Manager estiver instalado, o monitoramento do IBM Cloud Private deverá ser ativado para federar métricas de seus outros clusters.

Efetuando LogonOpenShift

O OpenShift fornece um serviço de criação de log opcional baseado no Elasticsearch que coleta logs de componentes de sistema e de aplicativo automaticamente. É possível optar por instalar o serviço de criação de log do IBM Cloud Private. Para obter mais informações, consulte Criação de log do IBM Cloud Private.

Saiba mais sobre o IBM Cloud Private with OpenShift.

- Preparando-se para instalar o IBM® Cloud Private no OpenShift
- Instalando o IBM Cloud Private with OpenShift
- Desinstalando o IBM Cloud Private with OpenShift
- Configurando a autenticação para o IBM Cloud Private with OpenShift
- Recursos no IBM Cloud Private em execução com o OpenShift que requerem customização
- Problemas conhecidos e limitações para o IBM Cloud Private with OpenShift

Preparando para Instalar o IBM Cloud Private with OpenShift

Antes de instalar o IBM Cloud Private with OpenShift, revise os seguintes requisitos de instalação.

Requisitos de Instalação

- A versão do IBM Cloud Private suportada para esta integração é 3.2.0.
- Você deve ter o OpenShift versão 3.11 instalado e funcionando em seu cluster que inclui serviços de registro e armazenamento.
- É necessário ter um StorageClass pré-configurado no OpenShift que pode ser usado para criar armazenamento para o IBM Cloud Private.

- Requisitos de hardware:
  - Linux® Plataforma de 64 bits
  - Nós principais: 8 Núcleos | 32 GB de RAM
    - Rotule o nó principal como cálculo, executando o comando a seguir:

      ```
      sudo kubectl label nodes {cluster_name-master.purple-chesterfield.com} node-role.kubernetes.io/compute=true
      ```
  - Nós do trabalhador: consulte Requisitos do OpenShift

- Rede:
  - O número da porta 8445 precisa ser aberto em cada nó no ambiente de S.O. para o exportador de nó no serviço de monitoramento. Esta porta é configurável e 8445 é o valor padrão.
  - Deve-se ter portas diferentes para o controlador de ingresso nginx se você implementar o ingresso nginx no nó principal do OpenShift. As portas 80 e 443 são usadas pelos serviços do OpenShift.
- Armazenamento:
  - Configure uma classe de armazenamento.
- Para o Elasticsearch, assegure-se de que a configuração vm.max_map_count seja pelo menos 262144 em todos os nós:

```
sysctl -w vm.max_map_count=262144
echo "vm.max_map_count=262144" | sudo tee -a /etc/sysctl.conf
```

Instalando o IBM Cloud Private with OpenShift

É possível instalar o IBM Cloud Private with OpenShift usando o instalador do IBM Cloud Private.

A instalação pode ser concluída em quatro etapas principais:

1. Configurar o nó de inicialização
2. Configure seu cluster
3. Execute o instalador do IBM Cloud Private
4. Tarefas pós-instalação

Configure o nó de inicialização

O instalador do IBM Cloud Private with OpenShift pode ser executado a partir de um nó de inicialização dedicado ou de um nó principal do OpenShift. Se o nó de inicialização não for um nó do OpenShift, instale o Docker apenas para seu nó de inicialização.

O nó de inicialização é o nó que é usado para a instalação de seu cluster. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre o nó de inicialização, consulte Nó de inicialização. Para o IBM Cloud Private with OpenShift, o nó de inicialização deve ser um dos nós do OpenShift.

É necessária uma versão do Docker que seja suportada por IBM Cloud Private with OpenShift instalada em seu nó de inicialização. Todas as versões do Docker que são suportadas pelo OpenShift são suportadas para o nó de inicialização. Para obter mais informações sobre as versões suportadas do Docker, consulte OpenShiftInstalação do Docker.

Para obter o procedimento para instalar o Docker, consulte Instalando manualmente o Docker.

Configure o ambiente de instalação

1. Efetue login no nó de inicialização como um usuário com permissões raiz ou como um usuário com privilégios sudo.
2. Faça download dos arquivos de instalação para o IBM Cloud Private 3.2.0. Deve-se fazer download do arquivo ou dos arquivos corretos para o tipo de nós em seu cluster. É possível obter esses arquivos no website IBM Passport Advantage®.
   - Para um cluster do Linux®, faça download do arquivo `ibm-cloud-private-rhos-3.2.0.tar.gz`.
3. Extraia a imagem e carregue-a no Docker. Extrair as imagens pode levar alguns minutos.

   ```
   tar xf ibm-cloud-private-rhos-3.2.0.tar.gz -O | sudo docker load
   ```

4. Crie um diretório de instalação para armazenar os arquivos de configuração do IBM Cloud Private nele e mude para esse diretório.

   Por exemplo, para armazenar os arquivos de configuração em `/opt/ibm-cloud-private-rhos-3.2.0`, execute os comandos a seguir:

   ```
   mkdir /opt/ibm-cloud-private-rhos-3.2.0; \
   cd /opt/ibm-cloud-private-rhos-3.2.0
   ```

5. Extraia o diretório do cluster:

   ```
   sudo docker run --rm -v $(pwd):/data:z -e LICENSE=accept --security-opt label:disable
   ibmcom/icp-inception-amd64:3.2.0-rhel-ee cp -r cluster /data
   ```

1. Criar arquivos de configuração de cluster. Os arquivos de configuração do OpenShift estão localizados no nó Principal do OpenShift.
   1. Copie o arquivo `admin.kubeconfig` do OpenShift para o diretório do cluster. O arquivo `admin.kubeconfig` do OpenShift pode ser localizado no diretório `/etc/origin/master/admin.kubeconfig`:

      ```
      sudo cp /etc/origin/master/admin.kubeconfig cluster/kubeconfig
      ```

   Se o nó de inicialização for diferente do nó principal do OpenShift, os arquivos anteriores deverão ser copiados para o nó de inicialização.

Configure seu cluster

1. Atualize o arquivo `config.yaml` extraído na etapa 5 com as seguintes configurações:

   Nota: O valor dos parâmetros `master`, `proxy` e `management` é uma matriz e pode ter vários nós. Devido a uma limitação do OpenShift, se você deseja implementar o IBM Cloud Private em qualquer nó principal do OpenShift, deve-se rotular o nó como um nó de cálculo do OpenShift com o comando a seguir:

   ```
   kubectl label --overwrite node <openshift-master-node-name> node-
   role.kubernetes.io/compute=true
   ```

   ```
   cluster_nodes:
     master:
       - <your-openshift-dedicated-node-to-deploy-icp-master-components>
     proxy:
   ```

```
    - <your-openshift-dedicated-node-to-deploy-icp-proxy-components>
  management:
    - <your-openshift-dedicated-node-to-deploy-icp-management-components>

storage_class: <storage class available in OpenShift>

openshift:
  console:
    host: <your-openshift-console-fqdn>
    port: <your-openshift-console-port>
  router:
    cluster_host: icp-console.<your-openshift-router-domain>
    proxy_host: icp-proxy.<your-openshift-router-domain>
```

2. Configure uma senha padrão no arquivo `config.yaml` que atenda à regra de regra cumprimento de senha padrão `'^([a-zA-Z0-9\-]{32,})$'`. Também é possível definir um conjunto customizado de regras de senha.

    1. Abra o arquivo `/<installation_directory>/cluster/config.yaml` e configure o `default_admin_password`. A senha deve satisfazer todas as expressões regulares que são especificadas em `password_rules`.

    2. Opcional: É possível definir uma ou mais regras como expressões regulares em uma lista de matrizes que a senha deve transmitir. Por exemplo, uma regra pode declarar que a senha deve ser maior que um número especificado de caracteres e/ou que deve conter pelo menos um caractere especial. As regras são gravadas como expressões regulares que são suportadas pela linguagem de programação Go. Para definir um conjunto de regras de senha, inclua o seguinte parâmetro e valores no arquivo `config.yaml`:

        ```
        password_rules:
        - '^.{10,}'
        - '.*[!@#\$%\^&\*].*'
        ```

        Para desativar `password_rule`, inclua `(.*)`

        ```
        password_rules:
        - '(.*)'
        ```

        Nota: O `default_admin_password` deve corresponder a todas as regras definidas. Se `password_rules` não estiver definido, o `default_admin_password` deverá atender à regra de cumprimento de passaporte padrão `'^([a-zA-Z0-9\-]{32,})$'`.

Execute o instalador do IBM Cloud Private

```
sudo docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster:z -v
/var/run:/var/run:z --security-opt label:disable ibmcom/icp-inception-amd64:3.2.0-rhel-ee install-
with-openshift
```

Nota: se você encontrar erros durante a instalação, será possível tentar novamente a instalação executando novamente o comando anterior.

Acesse o seu cluster

Acesse seu cluster usando uma porta diferente daquela que foi usada para o IBM Cloud Private independente. Em um navegador da web, navegue para a URL de seu cluster. Para obter uma lista de navegadores suportados, consulte Navegadores suportados.

Tarefas pós-instalação

Verifique qual nó executa os componentes do IBM Cloud Private

Para descobrir o nó que executa os componentes do IBM Cloud Private, execute o seguinte comando:

```
kubectl --kubeconfig /etc/origin/master/admin.kubeconfig get nodes
```

A saída mostra as funções de seu nó do OpenShift e qual nó executa os componentes do IBM Cloud Private.

Corrija as restrições de contexto de segurança

Para corrigir as restrições de contexto de segurança, execute o comando a seguir:

```
  kubectl --kubeconfig /etc/origin/master/admin.kubeconfig  patch scc icp-scc -p
'{"allowPrivilegedContainer": true}'
```

A saída deve ser semelhante ao texto a seguir:

```
  # kubectl --kubeconfig /etc/origin/master/admin.kubeconfig  patch scc icp-scc -p
'{"allowPrivilegedContainer": true}'
  securitycontextconstraints "icp-scc" patched
```

Depois de aplicar as novas restrições de contexto de segurança, a atualização a seguir será exibida:

```
  # kubectl --kubeconfig /etc/origin/master/admin.kubeconfig get scc icp-scc
  NAME      PRIV      CAPS      SELINUX     RUNASUSER   FSGROUP    SUPGROUP   PRIORITY
READONLYROOTFS   VOLUMES
  icp-scc   true      []        MustRunAs   RunAsAny    RunAsAny   RunAsAny   1          false
[configMap downwardAPI emptyDir hostPath nfs persistentVolumeClaim projected secret]
```

Desinstalando o IBM Cloud Private with OpenShift

Desinstale o IBM Cloud Private with OpenShift removendo todos os gráficos do HELM da plataforma implementados.

1. Efetue login no nó de inicialização como um usuário com permissões raiz. O nó de inicialização geralmente é o seu nó principal Para obter mais informações sobre tipos de nó, consulte Arquitetura. Durante a instalação, você especifica os endereços IP para cada tipo de nó.

2. Mude o diretório `cluster` dentro do seu diretório de instalação do IBM Cloud Private:

   ```
   cd /<installation_directory>/cluster
   ```

3. Desinstale o IBM Cloud Private executando o comando `uninstall-with-openshift`:

   ```
   docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-
  inception-$(uname -m | sed 's/x86_64/amd64/g')-amd64:latest-rhel-ee  uninstall-with-openshift
   ```

4. Reinicie o Docker em cada nó em seu cluster. Execute o comando a seguir em cada nó:

   ```
   service docker restart
   ```

5. Reinicie todos os nós em seu cluster.

Configurando autenticação para o IBM Cloud Private with OpenShift

É possível configurar o OpenShift para usar o provedor de autenticação do IBM Cloud Private OpenID Connect (OIDC).

Pré-requisito

Instale as ferramentas de CLI necessárias no nó principal do cluster OpenShift, como o comando `oc` do OpenShift, o cloudctl, o kubectl e o Helm.

- Introdução à CLI do OpenShift
- Instalando a CLI do IBM Cloud Private
- Acessando o cluster a partir da CLI do Kubernetes (kubectl) para IBM Cloud Private

Configure a CLI kubectl

A CLI do kubectl é usada para coletar informações de pré-requisito que são necessárias para a integração de autenticação do IBM Cloud Private com o OpenShift como o provedor OIDC.

1. Efetue login no nó principal do cluster OpenShift como um usuário com permissões raiz.

2. Efetue login usando a configuração do cliente do OpenShift, do IBM Cloud Private ou do kubectl. Siga as etapas na opção apropriada:

Opção 1: efetue login usando a linha de comandos do OpenShift, `oc login`, com o usuário administrativo:

1. Efetue login com o nome do usuário e a senha, executando o comando a seguir:

   ```
   oc login <openshift URL> -u <openshift admin username> -p <openshift admin password>
   ```

2. Efetue login com um token OAuth, seguindo estas etapas:

   a) Efetue login no console da web usando `admin`.

   b) Clique no nome do usuário. Copie e execute o comando a seguir:

   ```
   oc login <openshift URL> --token=<tokenID>
   ```

   Veja a lista a seguir para obter as descrições de comando:

   - `<openshift URL>`: URL do console da web do OpenShift. Por exemplo, https://api.your-openshift-server.com
   - `<openshift admin username>`: nome de usuário admin do OpenShift
   - `<openshift admin password>`: Senha do usuário administrador do OpenShift
   - `<tokenID>`: Token de login do usuário administrador do OpenShift

3. Cole o comando de login copiado em sua linha de comandos e pressione **Enter**.

Opção 2: efetue login usando a linha de comandos do IBM Cloud Private com seu nome do usuário e senha, executando o comando a seguir:

```
 cloudctl login [-a CLUSTER_URL][-u USERNAME] [-p PASSWORD][-c ACCOUNT_ID or ACCOUNT_NAME] [-n
namespace][--skip-ssl-validation]
```

Opção 3: efetue login usando a configuração do cliente kubectl, seguindo estas etapas:

1. Efetue login em seu console de gerenciamento do cluster do IBM Cloud Private como o administrador de cluster.

2. Selecione **Configurar cliente** e copie os detalhes de configuração do cluster:

   ```
   kubectl config set-cluster {cluster_name} --server=https://<Cluster Master Host>:5443 --
   insecure-skip-tls-verify=true
   kubectl config set-context {cluster_name}-context --cluster={cluster_name}
   kubectl config set-credentials {cluster_name}-user --token={token}
   kubectl config set-context {cluster_name}-context --user={cluster_name}-user --name
   space=default
   kubectl config use-context {cluster_name}-context
   ```

   Em que `<Cluster Master Host>` está definido em Terminal principal.

3. Cole as informações de configuração do cliente copiadas para seu cluster do IBM Cloud Private.

IBM Cloud Private Integração de autenticação com o OpenShift como o provedor OIDC

1. Efetue login no nó principal do cluster OpenShift como um usuário com permissões raiz.

2. Obtenha e salve as seguintes informações necessárias:

   - Salve `OAUTH2_CLIENT_REGISTRATION_SECRET`:

   ```
   export OAUTH2_CLIENT_REGISTRATION_SECRET=$(kubectl -n kube-system get secret platform-oidc-
   credentials -o yaml | grep OAUTH2_CLIENT_REGISTRATION_SECRET | awk '{ print$2}' | base64 --
   decode)
   ```

   - Salve `WLP_CLIENT_ID`:

   ```
   export WLP_CLIENT_ID=$(kubectl -n kube-system get secret platform-oidc-credentials-o yaml |
   grep WLP_CLIENT_ID | awk' { print $2}' | base64 --decode)
   ```

   - Salve `WLP_CLIENT_SECRET`:

   ```
   export WLP_CLIENT_SECRET=$(kubectl -n kube-system get secret platform-oidc-credentials -o yaml
   | grep WLP_CLIENT_SECRET | awk '{ print $2}' | base64 --decode)
   ```

   - Salve o IP de acesso:

   ```
   export MASTER_NODE_IP=<master node IP address>
   ```

- Salve os dados do JSON de registro do OIDC no arquivo `platform-oidc-registration.json`:

```
kubectl -n kube-system get cm registration-json -o "jsonpath={.data['platform-oidc-
registration\.json']}" > platform-oidc-registration.json
```

- Salve o certificado de autoridade de certificação do cluster do IBM Cloud Private em um arquivo `icp-ca.crt`:

```
kubectl -n kube-system get secret cluster-ca-cert -o yaml | grep tls.crt | awk '{ print $2}' |
base64 --decode > icp-ca.crt
```

3. Atualize o arquivo `platform-oidc-registration.json` e inclua a seguinte URL em `redirect_uris array`.

```
https://<openshift console access hostname or ipaddress>:<port>/oauth2callback/OIDC
```

4. Atualize o registro do OIDC no IBM Cloud Private para a URL de retorno de chamada do OpenShift aplicando as mudanças feitas no arquivo `platform-oidc-registration.json`.

```
curl -kvv -X PUT -u oauthadmin:$OAUTH2_CLIENT_REGISTRATION_SECRET -H "Content-Type:
application/json" -d @platform-oidc-registration.json https://$MASTER_NODE_IP:
<port>/oidc/endpoint/OP/registration/$WLP_CLIENT_ID
```

5. Configure o OIDC do IBM Cloud Private como o `identityProvider` para o OpenShift.

   1. Atualize o `master-config.yaml` com o conteúdo a seguir na seção `identityProvider`:

```
identityProvider:
...
...
- name: OIDC
challenge: true
login: true
provider:
apiVersion: v1
kind: OpenIDIdentityProvider
ca: icp-ca.crt
clientID: <WLP_CLIENT_ID>
clientSecret: <WLP_CLIENT_SECRET>
claims:
 id:
  - sub
urls:
 authorize: https://<ICP console hostname of ipaddress>:
<port>/idprovider/v1/auth/authorize
 token: https://<ICP console hostname of ipaddress>:<port>/idprovider/v1/auth/token
```

      Nota assegure-se de que as linhas estejam alinhadas adequadamente no arquivo `master-config.yaml`.

   2. Copie o certificado de CA `icp-ca.crt` para a pasta `/etc/origin/master/`:

```
cp icp-ca.crt /etc/origin/master/
```

   3. Reinicie o serviço OpenShift Master API com o comando a seguir:

```
systemctl restart <openShift master api servicename>
For example:
systemctl restart atomic-openshift-master-api.service
```

Efetue login no OpenShift usando o OIDC

1. Ative a URL do console da web do OpenShift em um navegador. Os provedores de `Login with..` são listados no console da web do OpenShift.
2. Clique no provedor OIDC para abrir a página de login do IBM Cloud Private.
3. Insira o log do IBM Cloud Private nas credenciais e pressione **Enter**.

Você está com login efetuado e redirecionado para o console da web do OpenShift.

Recursos no IBM Cloud Private with OpenShift que requerem customização

Ao executar o IBM Cloud Private with OpenShift, algum recurso requer customização para funcionar neste ambiente combinado.

Os recursos a seguir são opcionais para customização:

Horizontal Pod Autoscaler (HPA) em IBM Cloud Private

- Ative o Horizontal Pod Autoscaler (HPA) no recurso do IBM Cloud Private no `kube-controller-manager`, já que o OpenShift possui uma conversão de configuração exclusiva para o gerenciador do controlador Kube:

  ```
  horizontal-pod-autoscaler-use-rest-clients=true
  ```

Problemas conhecidos e limitações para o IBM Cloud Private with OpenShift

Revise os problemas conhecidos para a versão 3.2.0.

Porta de criação de log incorreta

Se você clicar em **Criação de log** na navegação do IBM Cloud Private, irá acessar `https://:8443/kibana/`, em que `8443` é uma porta incorreta. Mude a porta para o número da porta do IBM Cloud Private que está incluído no arquivo de instalação `config.yaml` (`https:// :/kibana/`) para exibir o painel do Kibana corretamente.

Exportador de nó em estado de erro

O exportador de nó pode estar no estado de erro devido ao pull de imagem malsucedido no ambiente do OpenShift.

Como solução alternativa para esse problema na pré-instalação, inclua o conteúdo a seguir no arquivo `config.yaml`:

```
monitoring:
nodeExporter:
  serviceAccount:
    name: "default"
```

Para solução alternativa para esse problema na pós-instalação, use o comando a seguir:

```
  kubectl patch ds/monitoring-prometheus-nodeexporter -n kube-system -p '{"spec":{"template":
{"spec":{"serviceAccount":"default","serviceAccountName":"default"}}}}'
```

Comando cURL incorreto

Você pode encontrar um erro `Connection Refused` devido ao comando cURL incorreto na CLI do IBM Cloud Private.

Para corrigir o erro, substitua `8443` por `5443`.

Erro de certificado após a execução do comando de login cloudctl

Se você executar o comando `cloudctl login` da CLI geral do IBM Cloud Private com o OpenShift instalado, as chamadas **kubectl** poderão receber um erro de certificado.

Para resolver esse problema, siga as etapas apropriadas para um Linux® principal ou um Mac OSx principal:

Em um Linux principal:

1. Substitua `<Cluster Master Host>:<Cluster Master API Port>` pelo terminal principal que está definido em Terminais principais nos seguintes comandos:

```
export OS_CA_CERT=$(openssl s_client -showcerts -connect <Cluster Master Host>:<Cluster Master API Port> </dev/null 2>/dev/null | openssl x509 -outform PEM)

export ICP_CA_CERT=$(kubectl -n kube-system get secret cluster-ca-cert -o yaml | grep 'tls.crt' | cut -d ":" -f 2 | xargs | base64 -d)

echo -e "$ICP_CA_CERT\n$OS_CA_CERT" | base64 | tr -d '\n'
```

2. Copie a saída e substitua os valores `tls.crt` no `cluster-ca-cert`:

```
kubectl -n kube-system edit secret cluster-ca-cert
```

Em um Mac principal:

1. Substitua `<Cluster Master Host>:<Cluster Master API Port>` pelo terminal principal que está definido em Terminais principais nos seguintes comandos:

```
export OS_CA_CERT=$(openssl s_client -showcerts -connect <Cluster Master Host>:<Cluster Master API Port> </dev/null 2>/dev/null | openssl x509 -outform PEM)

export ICP_CA_CERT=$(kubectl -n kube-system get secret cluster-ca-cert -o yaml | grep 'tls.crt' | cut -d ":" -f 2 | base64 -D)

echo -e "$ICP_CA_CERT\n$OS_CA_CERT" | base64 | tr -d '\n'
```

2. Copie a saída e substitua os valores `tls.crt` no `cluster-ca-cert`:

```
kubectl -n kube-system edit secret cluster-ca-cert
```

icp-scc SecurityContextContraints é designado erroneamente a todos os pods em todos os namespaces

Em clusters OpenShift, o recurso SecurityContextContraints `icp-scc` é designado erroneamente a todos os pods em todos os namespaces que usam Deployments, StatefulSets, DaemonSets, Jobs e outros controladores que gerenciam pods. O `icp-scc` SecurityContextContraints é designado, independentemente do ID do usuário que foi usado para criar o recurso ou a conta do serviço designada ao pod.

Para resolver o problema, execute os seguintes comandos `kubectl` no nó principal:

```
kubectl patch scc icp-scc --type='json' -p='[{"op": "remove", "path": "/groups"}]'
kubectl patch scc icp-scc --type='json' -p='[{"op": "add", "path": "/users", "value":
["system:serviceaccount:kube-system:default","system:serviceaccount:istio-system:default",
"system:serviceaccount:icp-system:default","system:serviceaccount:cert-manager:default"] }]'
```

IBM Cloud Private on AWS

É possível implementar um cluster do IBM Cloud Private no Amazon Web Services (AWS) usando o AWS CloudFormation ou o Terraform.

A decisão de usar o AWS CloudFormation ou o Terraform é uma questão de preferência. Há algumas pequenas diferenças no que especificamente é implementado.

Implementando o IBM Cloud Private no AWS usando o AWS CloudFormation

O IBM Cloud Private possui uma Iniciação Rápida que implementa automaticamente o IBM Cloud Private em uma nova nuvem particular virtual (VPC) no AWS Cloud. Uma implementação regular leva aproximadamente 60 minutos para ser concluída, e uma implementação de alta disponibilidade (HA) leva cerca de 75 minutos para ser concluída. A Iniciação rápida inclui os modelos de AWS CloudFormation e um guia de implementação.

Esta Iniciação Rápida é para usuários que desejam explorar a modernização do aplicativo e desejam atingir rapidamente seus objetivos de transformação digital usando ferramentas as IBM Cloud Private e IBM. A Iniciação Rápida ajuda os usuários a implementar rapidamente uma arquitetura de referência do IBM Cloud Private em nível de produção de alta disponibilidade (AD) no AWS.

Para obter todos os detalhes e o guia de implementação, consulte IBM Cloud Private no AWS Quick Start.

Implementando o IBM Cloud Private no AWS usando o Terraform

O IBM Cloud Private pode ser executado na plataforma de nuvem do AWS usando o Terraform. Para implementar o IBM Cloud Private em um ambiente AWS EC2, consulte Instalando o IBM Cloud Private no AWS.

Para obter detalhes sobre as funções do AWS Identity and Access Management (IAM), consulte o projeto cloud-provider-aws.

Há alguns atributos adicionais que podem ser definidos ao criar seu serviço do Kubernetes no AWS que o ajuda com a configuração do Balanceamento de Carga Elástica (ELB) do AWS correspondente. Também há atributos que podem ser analisados para a classe de armazenamento para gerenciar a maneira com que os PVCs são criados. Para obter mais informações, consulte a documentação do Kubernetes no AWS e no AWS Elastic Block Store (EBS).

IBM Cloud Private no Azure

O Microsoft Azure é um provedor de serviço de nuvem que oferece uma plataforma, infraestrutura, aplicativo ou serviços de armazenamento baseados em nuvem.

É possível ativar o Azure como um provedor em nuvem no IBM Cloud Private usando o arquivo `config.yaml` para customizar seu cluster. Assegure-se de atender a todos os requisitos do Azure antes de concluir as configurações para ativar o Azure como um provedor em nuvem e, em seguida, implementar o IBM Cloud Private no Azure.

Também é possível implementar o IBM Cloud Private no Azure usando o Terraform. Para os modelos Terraform, consulte o módulo Implementando o IBM Cloud Private no Azure.

- Requisitos do Azure
- Ativando o Azure como um provedor em nuvem

Requisitos do Azure

Antes de configurar o Azure como um provedor em nuvem e implementar o IBM Cloud Private no Azure, você deve revisar os requisitos do Azure aplicáveis.

Revise e configure, se necessário, os componentes necessários a seguir para o Azure:

- Funções do Azure
- Principal do serviço
- Grupo de recursos
- Rede Virtual e Sub-rede
- Grupo de segurança de rede
- Balanceadores de carga do Azure
- Conjuntos de Disponibilidade
- Zonas de Disponibilidade
- Dimensionamento da instância
- Conta de armazenamento

Funções do Azure

Você deve criar a função do Microsoft® Azure, `Contribuidor`, para criar e gerenciar todos os recursos do Azure. Para obter mais informações, consulte Funções RBAC do Microsoft Azure . Para criar uma designação de função com a interface da linha de comandos do Azure, consulte Designação Criar Função do Azure .

Principal do serviço

Um diretor de serviço é necessário para configurar o Azure. Crie um diretor de serviço exclusivo com permissões para o grupo de recursos Azure no qual o cluster IBM Cloud Private é implementado. O diretor de serviço Azure para seu cluster é usado para criar e gerenciar discos e balanceadores de carga de serviço do Kubernetes para armazenamento persistente. Com um diretor de serviço, é possível concluir as tarefas a seguir:

- Atualizar a tabela de rota do Azure com rotas customizadas para os pods que são executados em um nó específico.
- Criar e modificar os balanceadores de carga Azure para atender às solicitações do `LoadBalancer` de tipo de serviço do Kubernetes.
- Modificar os grupos de segurança para atender às solicitações do `LoadBalancer` de tipo de serviço do Kubernetes.
- Criar e anexar discos e arquivos Azure para atender às solicitações de volume persistente do Kubernetes com solicitações do `Storageclass` do Azure.

Inclua os valores do diretor do serviço no diretório `/etc/cfc/conf/azure.conf` para seus nós do cluster do IBM Cloud Private. Para obter mais informações, consulte Como criar um diretor de serviço Azure .

Para criar um diretor de serviço com a CLI do Azure, execute o comando a seguir:

```
az ad sp create-for-rbac --name icpprovidersp \
   --password <secret> --role contributor \
   --scopes /subscriptions/<subscription>/resourceGroups/<resource-group>
```

O arquivo JSON do Azure pode ser semelhante à saída a seguir:

```
{
  "appId": "<app-id>",
  "displayName": "icpprovidersp",
  "name": "http://icpprovidersp",
  "password": "<secret>",
  "tenant": "<tenant-id>"
}
```

Para obter mais informações, consulte Criar um diretor de serviço Azure com a CLI do Azure.

Nota: se um diretor de serviço não estiver disponível, entre em contato com o proprietário do Azure para ajudar a criar um diretor de serviço que você possa usar. A criação de um diretor de serviço no Azure requer a função de `Administrador de Acesso do Usuário`.

Grupo de recursos

Os grupos de recursos contêm todos os componentes do Azure que formam a implementação para o IBM Cloud Private no Azure. Para obter mais informações, consulte a seção *Grupos de Recursos* a partir da visão geral do Azure Resource Manager para obter mais detalhes.

Rede Virtual e sub-rede

Azure Virtual Network é um recurso do Azure que suporta o isolamento de redes de cluster umas das outras e permite que os recursos na rede virtual se comuniquem entre si e com a Internet.

Você deve criar uma sub-rede no Azure Virtual Network. Especifique a rede do pod para o parâmetro `network_cidr` em seu arquivo `config.yaml`. Para obter mais informações, consulte Azure Virtual Networks.

Grupo de segurança de rede

Os grupos de segurança de rede filtram o tráfego de rede para e a partir de recursos na rede virtual do Azure. Os grupos de segurança de rede definem uma lista de regras para permitir ou negar o tráfego com um valor numérico para indicar prioridade. Para obter mais informações, consulte Azure Virtual Networks.

Balanceadores de carga Azure

É possível usar os balanceadores de carga Azure para criar serviços de alta disponibilidade à medida que você implementa o IBM Cloud Private no Azure. Para obter mais informações, consulte Azure Load Balancer.

Todos os nós não principais devem estar no mesmo grupo de segurança, pois o provedor em nuvem Azure atualiza o grupo de segurança que está especificado no arquivo `config.yaml`. Se os nós de proxy, Vulnerability Advisor (VA), etcd e de gerenciamento tiverem grupos de segurança diferentes, os balanceadores de carga se comunicarão com os nós, mas seus grupos de segurança descartarão o tráfego. As solicitações que são enviadas para seu aplicativo podem falhar. É possível excluir nós do balanceador de carga com a seguinte anotação do nó: `alpha.service-controller.kubernetes.io/exclude-balancer=true`. Para obter mais informações, consulte Kubernetes Load Balancer .

Testando a integração do balanceador de carga

Para testar a integração do Azure Load Balancer, implemente uma carga de trabalho e exponha seu serviço executando os comandos a seguir:

```
kubectl run mynginx --image=nginx --replicas=2 --port=80
kubectl expose deployment mynginx --port=80 --type=LoadBalancer
```

Obtenha o endereço IP para o balanceador de carga Azure ao executar o comando a seguir:

```
kubectl get svc
```

Conjuntos de Disponibilidade

Os conjuntos de disponibilidade são um recurso anti-colocação do Azure que garante que as máquinas virtuais (VM) que desempenham a mesma função em uma solução de alta disponibilidade sejam colocadas em diferentes domínios de falha e de atualização dentro do mesmo data center. Para uma escala de alta disponibilidade, é ideal colocar seus nós principais em um conjunto de disponibilidade para manter a disponibilidade se o hardware falhar em um data center do Azure. Da mesma forma, os nós do trabalhador devem ser colocados em um conjunto de disponibilidade para melhorar a disponibilidade de suas cargas de trabalho do Kubernetes.

O Azure Load Balancer suporta o SKU Básico e Padrão. À medida que você usa o balanceador de carga SKU Básico com os conjuntos de disponibilidade, deve-se colocar todas as MVs não principais no mesmo conjunto de disponibilidade. Se quiser excluir o balanceador de carga Azure, você deve incluir a notação de nó a seguir em todos os nós do trabalhador não principais: `alpha.service-controller.kubernetes.io/exclude-balancecer=true`.

Zonas de disponibilidade

As zonas de disponibilidade são locais físicos exclusivos dentro de uma região do Azure que protegem aplicativos contra falhas do data center. À medida que você usa zonas de disponibilidade com o IBM Cloud Private, deve-se usar o balanceador de carga `SKU Padrão`.

Dimensionamento de instância

Veja os IBM Cloud Private Requisitos do sistema para assegurar que seu ambiente atenda aos requisitos mínimos.

Conta de armazenamento

Veja o IBM Cloud Private Guia de Armazenamento para assegurar que você atenda o requisito mínimo de agrupamento para seu cluster.

Implementação da classe de armazenamento (discos Azure)

Um objeto de classe de armazenamento é necessário para armazenamento persistente para aplicativos em uma implementação do IBM Cloud Private. Conclua as etapas a seguir para definir um objeto de classe de armazenamento padrão para uma implementação do IBM Cloud Private no Azure que está associada a discos Azure:

1. Crie o arquivo `storageclass-azure-disk.yaml` para definir o armazenamento suportado por discos Azure:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
 name: standard
 annotations:
   storageclass.kubernetes.io/is-default-class: "true"
provisioner: kubernetes.io/azure-disk
parameters:
 storageaccounttype: Standard_LRS
 kind: managed
```

2. Crie o objeto de classe de armazenamento executando o comando a seguir:

```
kubectl aplicar -f storageclass-azure-disk.yaml
```

Para obter mais informações, consulte o *Disco do Azure* nos MicrosoftDocs no GitHub.

Implementação de classe de armazenamento (arquivos Azure)

Crie uma implementação de classe de armazenamento que esteja associada aos arquivos Azure.

1. Crie a classe de armazenamento, o arquivo `storageclass-azure-file.yaml` para um arquivo Azure.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
 name: azurefile
```

```
provisioner: kubernetes.io/azure-file
mountOptions:
 - dir_mode=0777
 - file_mode=0777
 - uid=1000
 - gid=1000
parameters:
 skuName: Standard_LRS
```

2. Crie a classe de armazenamento em seu cluster do IBM Cloud Private executando o comando a seguir:

```
kubectl aplicar -f storageclass-azure-file.yaml
```

Para obter mais informações, consulte *Arquivo do Azure* nos MicrosoftDocs no GitHub

IBM Cloud Private ligação de função de cluster e de função de cluster

Para usar os provedores de armazenamento Azure, você deve incluir a função de cluster e o objeto de ligação de função de cluster a seguir:

1. Crie um arquivo denominado `roles-azure-cloud-provider.yaml` inserindo o conteúdo a seguir:

```
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: system:azure-cloud-provider
rules:
  - apiGroups:
      _ ""
    resources:
      - secrets
    verbs:
      - obter
      - criar

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: system:azure-cloud-provider
roleRef:
  kind: ClusterRole
  apiGroup: rbac.authorization.k8s.io
  name: system:azure-cloud-provider
subjects:
  - kind: ServiceAccount
    name: azure-cloud-provider
    namespace: kube-system
```

2. Execute o comando a seguir para incluir as funções em seu cluster:

```
kubectl aplicar -f papéis-azure-cloud-provider.yaml
```

Ativando o Azure como um provedor em nuvem

Para ativar o Azure como um provedor em nuvem, revise os Requisitos do Azure. Em seguida, configure seu cluster do IBM® Cloud Private para atualizar as configurações do Kubernetes, do Calico e do provedor em nuvem.

Atualizar as configurações do Kubernetes e do Calico

1. Abra o arquivo `config.yaml` na pasta `/<installation_directory>/cluster`.

2. Para modificar as configurações do Kuberbetes, atualize o arquivo com os parâmetros a seguir:

```
kube_controller_manager_extra_args:
- --allocate-node-cidrs=true
kubelet_extra_args:
- --enable-controller-attach-detach=true
```

3. Para modificar as configurações do Calico, atualize o arquivo com os parâmetros a seguir:

```
calico_ip_autodetection_method: can-reach=<ip>
calico_ipam_subnet: usePodCidr
calico_ipam_type: host-local
calico_ipip_mode: Always
calico_networking_backend: none
```

A tabela a seguir contém informações sobre os parâmetros Calico:

Parâmetros	Exemplo	Descrição
`calico_ip_autodetection_method`	can-reach=	O método a ser usado para

detectar automaticamente o endereço IPv4 para esse host.

Configurar as configurações do provedor Azure

1. Abra o arquivo `config.yaml` na pasta `/<installation_directory>/cluster`.
2. Inclua a chave `azure` com as seguintes subchaves:

```
azure:
  cloud_provider_conf:
  cloud_provider
```

   - A seção `cloud_provider_conf` descreve as configurações do provedor em nuvem que são passadas para o serviço kubelet e para o apiserver do Kubernetes.
   - A seção `cloud_provider_controller_conf` descreve as configurações que são passadas para o gerenciador do controlador do Kubernetes.
   - A instalação cria o arquivo JSON correspondente com os valores listados e o coloca na pasta `/etc/cfc/conf`.

3. No arquivo `config.yaml`, atualize a chave `cloud_provider_conf` e `cloud_provider_controller_conf` com os parâmetros de configuração do Azure Cloud Provider.

Consulte a documentação de Configuração do provedor em nuvem Azure para obter uma descrição de todos os parâmetros do provedor em nuvem suportados para o provedor Azure.

O arquivo `config.yaml` pode ser semelhante ao conteúdo a seguir:

```
azure:
  cloud_provider_conf:
    resourceGroup: "<name>"
    subscriptionId: "0000000-0000-0000-0000-000000000000"
    tenantId: "0000000-0000-0000-0000-000000000000"
    useManagedIdentityExtension: true
    useInstanceMetadata: true
  cloud_provider_controller_conf:
    aadClientId: "0000000-0000-0000-0000-000000000000"
    aadClientSecret: "0000000-0000-0000-0000-000000000000"
    cloud: "AzurePublicCloud"
    cloudProviderBackoff: false
    location: "eastus"
    resourceGroup: "<name>"
    routeTableName: "<name>"
    securityGroupName: "<name>"
    subscriptionId: "0000000-0000-0000-0000-000000000000"
    subnetName: "<name>"
    tenantId: "0000000-0000-0000-0000-000000000000"
    useManagedIdentityExtension: false
    useInstanceMetadata: true
    vnetName: "<name>"
    vnetResourceGroup: ""
```

O Azure Cloud Provider é ativado.

O IBM Cloud Private no Azure é iniciado incorretamente em um ambiente de alta disponibilidade. Para obter informações adicionais, consulte O IBM Cloud Private no Azure é iniciado incorretamente após a instalação.

Plataformas IBM Cloud Private Cloud Foundry e Cloud Foundry Enterprise Environment

IBM® Cloud Private Cloud Foundry oferece uma versão de pilha completa e uma versão de ambiente corporativo.

A plataforma Cloud Foundry Enterprise Environment é executada diretamente no IBM Cloud Private usando contêineres do Kubernetes. A plataforma IBM Cloud Private Cloud Foundry está disponível para ser executada no VMware vSphere, no AWS e no OpenStack.

A implementação tradicional do Cloud Foundry é feita usando o Bosh. O Bosh se comunica com o IAAS subjacente para implementar e manter a infraestrutura do Cloud Foundry usando máquinas virtuais. Esse método não aproveita a arquitetura de contêiner. O Cloud Foundry Enterprise Environment permite a implementação e a manutenção do Cloud Foundry no Kubernetes, sem a necessidade de Bosh.

- Introdução ao IBM Cloud Private Cloud Foundry e ao Cloud Foundry Enterprise Environment

Introdução ao IBM Cloud Private Cloud Foundry e ao Cloud Foundry Enterprise Environment

O IBM® Cloud Private Cloud Foundry traz o recurso de aplicativo da web do Cloud Foundry para seu data center no VMware vSphere, AWS e OpenStack.

A plataforma Cloud Foundry Enterprise Environment traz o recurso de aplicativo da web do Cloud Foundry para seu data center em contêineres do Kubernetes.

É possível estender o IBM Cloud Private Cloud Foundry e o Cloud Foundry Enterprise Environment usando liberações IBM ou da comunidade enquanto continua recebendo liberações certificadas pela IBM. É possível usar o IBM Cloud Private para gerenciar múltiplas linguagens de programação por meio de buildpacks e combinar aplicativos e serviços no local e baseados em nuvem. O Cloud Foundry suporta o ajuste de escala de aplicativos dinâmicos e usa o conceito de aplicativos simples para implementar e manter seus aplicativos corporativos.

- Visão geral do IBM Cloud Private Cloud Foundry e do Cloud Foundry Enterprise Environment
- O que há de novo no IBM Cloud Private Cloud Foundry e no Cloud Foundry Enterprise Environment Versão 3.2.0
- Considerações sobre plataforma para preparação para o RGPD
- Considerações sobre plataforma para preparação para o PCI

Visão Geral do IBM Cloud Private Cloud Foundry e do Cloud Foundry Enterprise Environment

A plataforma IBM® Cloud Private Cloud Foundry traz o recurso de aplicativo da web Cloud Foundry para seu data center no VMware vSphere, AWS e OpenStack. A plataforma Cloud Foundry Enterprise Environment traz o recurso de aplicativo da web do Cloud Foundry para seu data center em contêineres do Kubernetes.

- Cloud Foundry Enterprise Environment plataforma
- IBM Cloud Private Cloud Foundry plataforma

Plataforma Cloud Foundry Enterprise Environment

Para tirar vantagem de contêineres e Kubernetes, foi criado um novo método de implementação, plataforma Cloud Foundry Enterprise Environment. O Cloud Foundry Enterprise Environment permite a implementação e manutenção do Cloud Foundry no Kubernetes em vez de ambientes IaaS tradicionais. Com o Cloud Foundry Enterprise Environment, é possível executar suas cargas de trabalho do Kubernetes lado a lado do ambiente Cloud Foundry. Também é possível usar quaisquer serviços que estão disponíveis no IBM Cloud Private ou no Kubernetes em aplicativos do Cloud Foundry usando a extensão do broker Open Service.

Nota: A implementação tradicional do Cloud Foundry é feita usando o Bosh; no entanto, a plataforma Cloud Foundry Enterprise Environment não usa o Bosh. Uma limitação dessa abordagem é que não é possível implementar liberações do Bosh criadas pelo usuário nesse ambiente.

O Cloud Foundry Enterprise Environment que está disponível no IBM Cloud Private Versão 3.2.0 oferece o Cloud Foundry Versão 2.7.

Ofertas

É possível implementar o Cloud Foundry Enterprise Environment em dois modos: Developer e Enterprise.

O modo **Developer** usa uma pequena área de cobertura e é uma boa maneira de aprender sobre o produto. Ele ainda é implementado em uma infraestrutura corporativa, mas sua área de cobertura é mínima.

No modo **Enterprise**, você implementa todos os componentes em dois ou mais clusters de nó. Esse modelo de implementação fornece uma segunda camada de proteção, além da proteção que é oferecida pela alta disponibilidade (HA) da infraestrutura como serviço (IaaS), para assegurar que suas cargas de trabalho permaneçam em execução. Após a plataforma HA ser inicializada, é possível expandir ou reduzir seus nós para se ajustar às suas necessidades específicas.

Para obter informações adicionais sobre o Cloud Foundry Enterprise Environment, consulte os seguintes tópicos que são específicos dessa implementação, além das informações comuns ao Cloud Foundry Enterprise Environment e ao IBM Cloud Private Cloud Foundry em toda esta seção do Knowledge Center. As informações que não são aplicáveis ao Cloud Foundry Enterprise Environment incluem a seguinte nota: **IBM® Cloud Private Cloud Foundry:** Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

- Dimensionamento do Cloud Foundry Enterprise Environment
- Parâmetros do Cloud Foundry Enterprise Environment
- Instalando o Cloud Foundry Enterprise Environment
- Aplicativos de Autoscaling no Cloud Foundry Enterprise Environment
- A implementação do Cloud Foundry Enterprise Environment falha
- O Stager está indisponível
- Não é possível desvincular um serviço OSB de um aplicativo Cloud Foundry
- Desinstalando o Cloud Foundry Enterprise Environment

Plataforma IBM Cloud Private Cloud Foundry

É possível estender o IBM Cloud Private Cloud Foundry usando liberações da IBM ou da comunidade enquanto você continua recebendo liberações certificadas pela IBM. Esse recurso oferece a capacidade de usar recursos novos ou especializados que nem sempre estão presentes nas implementações do Cloud Foundry.

O IBM Cloud Private Cloud Foundry, que está disponível no IBM Cloud Private Versão 3.2.0, oferece o Cloud Foundry Versão 4.5.0.

- Recursos
- Ofertas

Recursos

IBM Cloud Private Cloud Foundry

- É construído na arquitetura Diego do Cloud Foundry
- Usa uma ferramenta dockerizada de instalação e atualização
- Oferece várias opções de autenticação, incluindo UAA integrado e LDAP corporativo
- Contém os buildpacks de comunidade do Cloud Foundry para as versões IBM do Node.js, Liberty e Swift
- Permite as customizações do usuário a seguir:
    - Mudar as configurações de implementação do Director ou do Cloud Foundry (modificar configurações e incluir ou remover liberações)
    - Incluir ou remover buildpacks
    - Modificar certificados, segredos e senhas
    - Usar um Stemcell customizado
    - Conectar-se aos seus serviços existentes de criação de log e monitoramento ou usar um serviço IBM Cloud Private do Catalog

Ofertas

É possível implementar o IBM Cloud Private Cloud Foundry de dois modos, Developer e Enterprise. O modo **Developer** usa uma pequena área de cobertura e é uma boa maneira de aprender sobre o produto. Ele ainda é implementado em uma infraestrutura corporativa, mas sua área de cobertura é mínima. No modo **Enterprise**, você implementa todos os componentes em 2 ou 3 clusters do nó. Esse modelo de implementação fornece uma segunda camada de proteção, além da proteção que é oferecida pela

alta disponibilidade (HA) da infraestrutura como serviço (IaaS), para assegurar que suas cargas de trabalho permaneçam em execução. Após a plataforma HA ser inicializada, é possível expandir ou reduzir seus nós para se ajustar às suas necessidades específicas.

Ambas as opções de implementação estão disponíveis como um arquivo binário único a partir do IBM Passport Advantage® para suportar um cenário de isolamento físico. O instalador é pequeno o suficiente para que você possa executá-lo em uma máquina virtual (VM) pequena usando Docker. É possível executar a VM em um computador individual ou em uma infraestrutura corporativa.

- Instalação
- Guia do Operador
- Guia de CLI
- Guia do Desenvolvedor
- Resolução de problemas

O que há de novo no IBM Cloud Private Cloud Foundry e no Cloud Foundry Enterprise Environment versão 3.2.0

Obtenha uma visão geral rápida do que foi incluído, mudado, melhorado ou descontinuado nessa liberação.

- Cloud Foundry Enterprise Environment
- IBM Cloud Private Cloud Foundry

Cloud Foundry Enterprise Environment

A versão 3.2.0 apresenta uma liberação completa do Cloud Foundry Enterprise Environment: A plataforma Cloud Foundry Enterprise Environment apresenta o recurso de aplicativo da web do Cloud Foundry para seu data center em contêineres do Kubernetes. O Cloud Foundry Enterprise Environment usa o Cloud Foundry versão 2.7.

Novos recursos

- O Cloud Foundry Enterprise Environment agora oferece uma extensão que facilita a configuração de um broker de serviço de cluster. O broker de serviço Kubernetes Helm também é configurado para o mercado de Cloud Foundry, que disponibiliza o catálogo do Kubernetes Helm (sob controle de acesso) para aplicativos Cloud Foundry como parte do processo de instalação padrão. Para obter informações adicionais, consulte Extensão do instalador do Open Service Broker (OSB).
- O monitoramento para Cloud Foundry é configurado automaticamente na plataforma Kubernetes do IBM Cloud Private de hosting e cinco painéis Grafana são carregados automaticamente. Consulte Conectando o IBM Cloud Private Cloud Foundry ao Prometheus
- Agora o Cloud Foundry Enterprise Environment inclui capacidade para escalar células Diego para usar mais nós do trabalhador em seu ambiente. Consulte Dimensionamento do Cloud Foundry Enterprise Environment.
- A instalação usa distorção e tolerância para que os contêineres de célula Diego e o plano de controle possam ser divididos em nós do trabalhador específicos. Consulte Instalando Cloud Foundry Enterprise Environment.
- Cloud Foundry Enterprise Environment tem suporte ao ajuste de escala automático integrado para aplicativos Cloud Foundry. Consulte Fazendo o ajuste de escala automático de aplicativos no Cloud Foundry Enterprise Environment.
- Cloud Foundry Enterprise Environment agora suporta integração LDAP. Consulte Configurando a autenticação LDAP para IBM Cloud Private Cloud Foundry.
- O TLS1.2 protege a comunicação externa e de loopback. Consulte Instalando o IBM Cloud Private Cloud Foundry com o Ferramenta de implementação do Cloud Foundry.
- O console de implementação inclui atualizações de segurança e de recursos. Agora, ele é internacionalizado e suporta diversos idiomas. Consulte Usando extensões no IBM Cloud Private Cloud Foundry.

Componentes

Nome da liberação	Versão	IBM
app-autoscaler-release	v1.1.0	sim
capi-release	1.61.0.1.0	sim
cf-syslog-drain-release	7.0+dev.1	sim
cflinuxfs2-release	1.227.0+dev.1	sim
consul-release	195+dev.1	sim
diego-release	2.12.2+dev.1	sim
elk-adapter-release	1.6.16	sim

Nome da liberação	Versão	IBM
garden-runc-release	1.17.2+dev.1	sim
loggregator-agent-release	2.0+dev.1	sim
loggregator-release	103.0+dev.1	sim
nats-release	24+dev.1	sim
postgres-release	26+dev.1	sim
routing-release	0.179.0.1.0	sim
scalerui-release	1581903b	sim
statsd-injector-release	1.3.0	sim
uaa-fissile-release	0.0.1-321-g6c32268	não
uaa-release	60.2.1.0	sim
binary-buildpack-release	1.0.17	sim
go-buildpack-release	1.8.20	sim
java-buildpack-release	4.9	sim
nodejs-buildpack-release	1.6.20	sim
php-buildpack-release	4.3.51	sim
python-buildpack-release	1.6.11	sim
ruby-buildpack-release	1.7.15	sim
staticfile-buildpack-release	1.4.24	sim
liberty-for-java_buildpack	3.29	sim
sdk-for-nodejs_buildpack	3.26	sim
dotnet-core_buildpack	2.2	sim
swift_buildpack	2.0.18	sim

Buildpacks IBM

- O buildpack Swift é atualizado para v2.0.20-20190401-2122
- O buildpack Liberty for Java é atualizado para v3.29-20190223-2128

IBM Cloud Private Cloud Foundry

Aprimoramentos de recurso

- Agora é possível usar uma extensão para modificar o manifest de implementação do IBM Cloud Private Cloud Foundry para usar bancos de dados externos. Para obter informações adicionais, consulte Configurando bancos de dados remotos para o IBM Cloud Private Cloud Foundry.
- É possível atualizar versões secundárias de stemcells existentes para a correção mais recente, configurando a extensão da versão de stemcell no IBM Cloud Private Cloud Foundry. Para obter informações adicionais, consulte Fazendo upgrade de versões secundárias de stemcell.
- O TLS1.2 protege a comunicação externa e de loopback. Consulte Instalando o IBM Cloud Private Cloud Foundry com o Ferramenta de implementação do Cloud Foundry.
- O console de implementação inclui atualizações de segurança e de recursos. Agora, ele é internacionalizado e suporta diversos idiomas. Consulte Usando extensões no IBM Cloud Private Cloud Foundry.

Novos componentes

- `cflinuxfs3` é a pilha padrão para aplicativos. Consulte Usando buildpacks no IBM Cloud Private Cloud Foundry .

O que mudou

Com a introdução do IBM Cloud Private Cloud Foundry versão 3.2.0, as versões do pacote a seguir foram mudadas:

- O Cloud Foundry `cf-deployment` foi atualizado para 7.5.0
- O Cloud Foundry `Stemcell` foi atualizado para 250.9
- O `silk` foi atualizado para 2.20.0
- O `credhub` foi atualizado para 2.1.2
- O `bosh` foi atualizado para 268.6.0
- O `bosh-vsphere-cpi` foi atualizado para 52
- O `capi` foi atualizado para 1.76.0

- O `cf-networking` foi atualizado para 2.20.0
- O `cf-syslog-drain` foi atualizado para 9.0
- O `cflinuxfs2` foi atualizado para 1.260.0
- O `diego` foi atualizado para 2.28.0
- O `garden-runc` foi atualizado para 1.18.2
- O `loggregator` foi atualizado para 104.5
- O `routing` foi atualizado para 0.184.0
- O `uaa` foi atualizado para 69.0

Buildpacks IBM

- Os utilitários do Application Management foram agora descontinuados no SDK para Node.js e são elegíveis para remoção em uma liberação futura. Para obter mais informações, consulte Descontinuação do Separate Application Management no IBM SDK for Node.js Buildpack.
- O Node.js foi atualizado de v3.25.1-20190115-1637 para v3.26-20190313-1440.
- O Swift foi atualizado de v2.0.16-20181214-0434 para v2.1.0-20190404-1206.
- O Liberty foi atualizado da v3.27-20181130-1702 para v3.31-20190423-1354.
- O DotNet foi atualizado da versão v2.1-20181205-1536 para v2.2-20190327-1013. Para obter mais informações sobre o conteúdo da comunidade nesse buildpack, consulte Atualizações mais recentes para o buildpack do ASP.NET Core.

Buildpacks da comunidade

- O buildpack binário foi atualizado da versão 1.0.27 para a versão 1.0.30.
- O buildpack Go foi atualizado da versão 1.8.28 para a versão 1.8.33.
- O buildpack PHP foi atualizado da versão 4.3.61 para a versão 4.3.70.
- O buildpack Java foi atualizado da versão 4.16.1 para a versão 4.17.2.
- O buildpack Ruby foi atualizado da versão 1.7.24 para a versão 1.7.31.
- O buildpack Python foi atualizado da versão 1.6.23 para a versão 1.6.28.
- O buildpack Staticfile foi atualizado da versão 1.4.32 para a versão 1.4.39.
- O buildpack Nginx está na versão 1.0.8.
- O buildpack R está na versão 1.0.3.

IBM® Cloud Private Cloud Foundry Considerações sobre plataforma para preparação GDPR

Aviso

Este documento tem como objetivo ajudá-lo em suas preparações para prontidão do GDPR. Ele fornece informações sobre recursos da plataforma IBM Cloud Private Cloud Foundry que podem ser configurados e os aspectos de uso do produto, que devem ser considerados para ajudar sua organização com prontidão do GDPR. Essas informações não são uma lista exaustiva, devido às muitas formas que os clientes podem escolher e configurar recursos e à grande variedade de maneiras que o produto pode ser usado em si e com aplicativos e sistemas de terceiros.

Os clientes são responsáveis por assegurar sua própria conformidade com várias leis e regulamentações, inclusive com a General Data Protection Regulation da União Europeia. Os clientes são responsáveis apenas por obter aviso de uma consultoria jurídica competente quanto à identificação e interpretação de quaisquer leis e regulamentações relevantes que possam afetar os negócios dos clientes e quaisquer ações que os clientes possam precisar tomar para obedecerem a tais leis e regulamentações.

Os produtos, serviços e outros recursos descritos neste documento não são adequados para todas as situações dos clientes e podem ter disponibilidade restringida. A IBM não fornece aviso jurídico, contábil ou de auditoria nem representa ou garante que seus serviços ou produtos assegurarão que os clientes estejam em conformidade com qualquer lei ou regulamentação.

Índice

GDPR

O General Data Protection Regulation (GDPR) foi adotado pela União Europeia ("EU") e aplica-se a partir de 25 de maio de 2018.

Por que é importante? GDPR

O GDPR estabelece uma estrutura regulamentar de proteção de dados mais forte para processamento de dados pessoais de indivíduos. GDPR traz:

- Novos direitos e aprimorado para indivíduos
- Definição ampliada de dados pessoais
- Novas obrigações para processadores
- Potencial para multas financeiras significativas por não conformidade
- Notificação de violação de dados obrigatórios

Leia mais sobre o GDPR

- Portal de informações do GDPR da EU
- Website ibm.com/GDPR

Configuração do produto - considerações para Prontidão do GDPR

As seções a seguir descrevem aspectos do gerenciamento de dados dentro da plataforma IBM Cloud Private Cloud Foundry e fornecem informações sobre recursos para ajudar os clientes com a prontidão para a GDPR.

Ciclo de Vida de Dados

O IBM Cloud Private Cloud Foundry é uma plataforma de aplicativo para desenvolver e gerenciar aplicativos no local. Ele é um ambiente integrado para gerenciar aplicativos e aplicativos de contêiner, que incluem o Cloud Foundry, um console de gerenciamento e as estruturas de monitoramento.

Dessa forma, a plataforma IBM Cloud Private Cloud Foundry lida principalmente com dados técnicos que estão relacionados à configuração e ao gerenciamento da plataforma, alguns dos quais podem estar sujeitos ao GDPR. A plataforma IBM Cloud Private Cloud Foundry também lida com informações sobre os usuários que gerenciam a plataforma. Esses dados são descritos em todo este documento para ajudar os clientes com a prontidão para a GDPR.

Esses dados são persistidos na plataforma em sistemas de arquivos locais ou remotos como arquivos de configuração ou em bancos de dados. Os aplicativos que são desenvolvidos para serem executados na plataforma IBM Cloud Private Cloud Foundry podem lidar com outras formas de dados pessoais sujeitos ao GDPR. Os mecanismos que são usados para proteger e gerenciar os dados da plataforma também estão disponíveis para aplicativos que são executados na plataforma. Mecanismos adicionais podem ser necessários para gerenciar e proteger dados pessoais que são coletados por aplicativos que são executados na plataforma IBM Cloud Private Cloud Foundry.

Para entender melhor a plataforma IBM Cloud Private Cloud Foundry e seus fluxos de dados, deve-se entender como o Cloud Foundry funciona. Use o Cloud Foundry para hospedar instâncias de aplicativos, que são buildpacks de linguagem de programação compilados. O buildpack contém os componentes de tempo de execução compilador e chave, um contêiner garden é usado como um ambiente de simulação para seu aplicativo e o buildpack, em conjunto com esses componentes, publicam seu aplicativo na plataforma.

O IBM Cloud Private Cloud Foundry inclui uma série de buildpacks comerciais e da comunidade (linguagens). Para visualizar uma lista de todos os buildpacks do IBM Cloud Private Cloud Foundry, consulte O que há de novo no IBM Cloud Private Cloud Foundry. Para obter considerações sobre a GDPR para os buidpacks, consulte a documentação para esses produtos. As informações sobre os pacotes configuráveis do IBM Cloud Private disponíveis, que contêm a plataforma do IBM Cloud Private principal e o software autorizado disponível, estão disponíveis aqui, em IBM Cloud Private Cloud Foundrypacotes configuráveis. Alguns dos buildpacks são software livre. É responsabilidade do cliente determinar e implementar os controles apropriados do GDPR para software livre.

A documentação sobre a plataforma IBM Cloud Private pode ser localizada na Coleção do IBM Cloud Private no IBM Knowledge Center.

Que tipos de dados fluem pela plataforma IBM Cloud Private Cloud Foundry

Como plataforma, o IBM Cloud Private Cloud Foundry lida com várias categorias de dados técnicos que podem ser considerados como dados pessoais, como um ID do usuário administrador e senha padrão, IDs do usuário e senhas de serviço, endereços IP, nomes da organização do Cloud Foundry e nomes de espaço do Cloud Foundry. A plataforma IBM Cloud Private Cloud Foundry também lida com informações sobre os usuários que gerenciam a plataforma. Os aplicativos que são executados na plataforma podem apresentar outras categorias de dados pessoais desconhecidos para a plataforma.

Informações sobre como esses dados técnicos são coletados, criados, armazenados, acessados, protegidos, registrados e excluídos são descritas em seções posteriores deste documento.

Dados pessoais usados para contato on-line com a IBM

Os clientes do IBM Cloud Private Cloud Foundry podem enviar comentários/feedback/solicitações on-line para entrar em contato com a IBM sobre assuntos do IBM Cloud Private Cloud Foundry de várias maneiras, principalmente:

- A comunidade pública IBM Cloud Private-CE (Community Edition) Slack
- Área de comentários públicos nas páginas da documentação do produto IBM Cloud Private no IBM Knowledge Center
- Comentários públicos no espaço do IBM Cloud Private de dW Answers

Geralmente, somente o nome do cliente e o endereço de e-mail são usados para permitir respostas pessoais para o assunto do contato e o uso de dados pessoais em conformidade com o IBM Online Privacy Statement em .

Coleta de Dados

A plataforma IBM Cloud Private Cloud Foundry não coleta dados pessoais sensíveis. Ele cria e gerencia dados técnicos, como um ID do usuário administrador padrão e senha, IDs de usuário e senhas de serviço e endereços IP, que podem ser considerados dados pessoais. A plataforma IBM Cloud Private Cloud Foundry também lida com informações sobre os usuários que gerenciam a plataforma. Todas essas informações são acessíveis somente pelo administrador do sistema por meio de um console de gerenciamento com controle de acesso baseado na função ou pelo administrador do sistema por meio de login em um nó da plataforma IBM Cloud Private Cloud Foundry.

Os aplicativos que são executados na plataforma IBM Cloud Private Cloud Foundry podem coletar dados pessoais.

Ao avaliar o uso dos aplicativos em execução da plataforma IBM Cloud Private Cloud Foundry e sua necessidade de atender aos requisitos de GDPR, deve-se considerar os tipos de dados pessoais que são coletados pelo aplicativo e os aspectos de como esses dados são gerenciados, como:

- Como os dados são protegidos enquanto fluem para/do aplicativo? Os dados são criptografados em trânsito?
- Como os dados são armazenados pelo aplicativo? A dados criptografados em repouso.
- Como as credenciais usadas para acessar o aplicativo são coletadas e armazenadas?
- Como as credenciais usadas pelo aplicativo para acessar origens de dados são coletadas e armazenadas?
- Como os dados coletados pelo aplicativo são removidos conforme necessário?

Esta lista não é uma lista definitiva dos tipos de dados que são coletados pela plataforma IBM Cloud Private Cloud Foundry. Ela é fornecida como um exemplo para consideração. Se você tiver quaisquer perguntas sobre os tipos de dados, entre em contato com a IBM.

Armazenamento de dados

A plataforma IBM Cloud Private Cloud Foundry persiste dados técnicos que estão relacionados à configuração e ao gerenciamento da plataforma em armazenamentos stateful em sistemas de arquivos locais ou remotos como arquivos de configuração ou em bancos de dados. Deve-se considerar assegurar todos os dados em repouso. A plataforma IBM Cloud Private Cloud Foundry permite a criptografia de dados em repouso por meio de suas ferramentas corporativas existentes. Para obter mais informações, consulte Usando as ferramentas de criptografia do cliente como extensões do IBM Cloud Private Cloud Foundry. Outra opção é usar dispositivos SAN, NAS ou vSAN que suportam a criptografia em repouso.

Os itens a seguir destacam as áreas em que os dados são armazenados, que você pode desejar considerar para o GDPR.

- **Dados de configuração da plataforma:** a configuração da plataforma IBM Cloud Private Cloud Foundry pode ser customizada atualizando um arquivo YAML de configuração com propriedades para configurações gerais. Esses dados são usados como entrada para o instalador da plataforma IBM Cloud Private Cloud Foundry para implementação do Cloud Foundry. As propriedades também incluem um ID do usuário administrativo e senha padrão que são usados para autoinicialização. Para obter mais informações, consulte Instalando o Cloud Foundry.

- **Dados de configuração do Cloud Foundry:** são armazenados em um banco de dados Postgres. Para obter mais informações sobre o uso de origens de dados externas, consulte Bancos de dados.
- **Dados de autenticação do usuário, incluindo IDs do usuário e senhas:** o gerenciamento de ID do usuário e senha é manipulado por meio de um diretório LDAP corporativo do cliente. Os usuários que são definidos no LDAP podem ser incluídos em organizações e espaços da plataforma IBM Cloud Private Cloud Foundry e designados a funções de acesso. A plataforma IBM Cloud Private Cloud Foundry armazena o `userid` do LDAP, mas não armazena a senha. A proteção de dados do usuário em repouso no LDAP corporativo deve ser considerada.
- **Dados de autenticação de serviço, incluindo IDs do usuário e senhas:** as credenciais que são usadas pelos aplicativos IBM Cloud Private Cloud Foundry para acessar serviços externos têm seus metadados de serviço vcap criptografados dentro do banco de dados do Cloud Controller. A chave de criptografia é gerada durante a instalação ou pode ser fornecida por você.
- **Dados de serviço:**IBM Cloud Private Cloud Foundry a plataforma inclui um catálogo de serviços externos catalogados por você.
- **Dados de monitoramento:** é possível usar o monitoramento da plataforma IBM Cloud Private Cloud Foundry para monitorar o status de seu Cloud Foundry e aplicativos. Este serviço pode usar o Grafana e o Prometheus para apresentar informações detalhadas sobre nós e contêineres de cluster. Pilhas adicionais de monitoramento podem ser implementadas para monitoramento de aplicativo. Os dados de monitoramento podem ser persistidos usando `PersistentVolumes` do Kubernetes. Para obter informações adicionais, consulte Plug-ins do Prometheus e Plug-in Splunk opcionais.
- **Dados de criação de log:**IBM Cloud Private Cloud Foundry a plataforma usa logs de rolagem do Cloud Foundry padrão. O sistema pode ser apontado para uma pilha ELK externa. ELK é uma abreviação de três produtos, Elasticsearch, Logstash e Kibana, que são construídos pela Elastic e juntos formam uma pilha de ferramentas que podem ser usadas para transmitir, armazenar, procurar e monitorar logs. Para obter mais informações, consulte Integrando o syslog ao ELK.

Acesso de Dados

Os dados da plataforma IBM Cloud Private Cloud Foundry podem ser acessados por meio do conjunto definido de interfaces do produto a seguir.

- Interface com o usuário da Web (o console de gerenciamento)
- CLI do Cloud Foundry

Essas interfaces são projetadas para permitir que você faça mudanças administrativas em sua plataforma IBM Cloud Private Cloud Foundry. O acesso de administração ao IBM Cloud Private Cloud Foundry pode ser assegurado e envolve três estágios lógicos e ordenados quando uma solicitação é feita: autenticação, mapeamento de função e autorização.

Autenticação

A CLI ou o console do IBM Cloud Private Cloud Foundry solicita acesso à API da plataforma. A API direciona a CLI para os servidores User Account and Authentication (UAA). A UAA redireciona a solicitação para o servidor de login. O servidor de login aceita e valida o ID do usuário e a senha com relação ao servidor LDAP configurado. Se a autenticação for bem-sucedida, as funções de acesso serão fornecidas com um token para acesso.

Para todas as solicitações de autenticação subsequentes feitas a partir do console de gerenciamento, o token é usado com a solicitação e validado ao chamar o servidor User Account and Authentication.

A CLI da plataforma IBM Cloud Private Cloud Foundry requer que o usuário forneça credenciais para efetuar login.

Mapeamento de função

A plataforma IBM Cloud Private Cloud Foundry suporta o controle de acesso baseado na função (RBAC). No estágio de mapeamento de função, o nome do usuário que é fornecido no estágio de autenticação está associado a organizações e espaços. O ID do usuário pode ter funções concedidas em múltiplas áreas. O ID do usuário também pode ter funções administrativas concedidas usando a CLI do User Account and Authentication (UAAC).

Autorização

As funções da plataforma IBM Cloud Private Cloud Foundry controlam o acesso aos aplicativos e serviços.

Segurança do Bosh

O Bosh é usado para gerenciar a infraestrutura da plataforma virtual. Para obter mais informações, consulte Comandos do Bosh frequentes.

Processamento de Dados

Os usuários do IBM Cloud Private Cloud Foundry podem controlar a maneira pela qual dados técnicos relacionados à configuração e ao gerenciamento são processados e assegurados por meio da configuração do sistema.

O **Controle de acesso baseado na função** (RBAC) controla quais dados e funções podem ser acessados pelos usuários.

A **segurança do Bosh** é usada para configurar e controlar a infraestrutura virtual.

Dados-em-Trânsito é protegido usando `TLS`. O `HTTPS` (`TLS` subjacente) é usado para a transferência de dados segura entre o cliente do usuário e os dispositivos de proxy de entrada. Os usuários podem especificar os certificados raiz e curinga a serem usados para essa transferência durante a instalação. O TLS pode ser estendido para o GoRouter também por meio de uma customização do IBM Cloud Private Cloud Foundry.

A proteção de **Dados em repouso** é suportada usando as ferramentas de criptografia do cliente como extensões do IBM Cloud Private Cloud Foundry ou criptografando usando recursos de criptografia no nível da infraestrutura.

Períodos de **Retenção de dados** para criação de log (ELK) e monitoramento (Prometheus) são configuráveis e a exclusão de dados é suportada.

Esses mesmos mecanismos de plataforma que são usados para gerenciar e assegurar os dados técnicos da plataforma IBM Cloud Private Cloud Foundry podem ser usados para gerenciar e assegurar dados pessoais para aplicativos desenvolvidos pelo usuário ou fornecidos pelo usuário. É possível desenvolver seus próprios recursos para implementar controles adicionais.

Exclusão de Dados

A plataforma IBM Cloud Private Cloud Foundry fornece comandos, interfaces de programação de aplicativos (APIs) e ações da interface com o usuário para excluir dados que são criados ou coletados pelo produto. Essas funções permitem que os usuários excluam dados técnicos, como IDs de usuário e senhas de serviço, endereços IP ou qualquer outro dado de configuração da plataforma, bem como informações sobre usuários que gerenciam a plataforma.

Áreas da plataforma IBM Cloud Private Cloud Foundry a serem consideradas para suporte de exclusão de dados:

- O período de retenção de dados para dados de criação de log é controlado pelo cliente.
- O período de retenção de dados para dados de monitoramento (Prometheus) é controlado pelo cliente.
- Ao usar o ELK, os dados de criação de log podem ser excluídos da pilha ELK usando APIs do Elasticsearch.
- Ao usar o Prometheus, os dados de monitoramento podem ser excluídos de Prometheus usando as APIs do Prometheus.
- Todos os dados técnicos que estão relacionados à configuração da plataforma podem ser excluídos por meio do console de gerenciamento ou da API do Cloud Foundry.

Áreas da plataforma IBM Cloud Private Cloud Foundry a serem consideradas para suporte de exclusão de dados da conta:

- Todos os dados técnicos que estão relacionados à configuração da plataforma podem ser excluídos por meio da API do Cloud Foundry.

A função para remover dados de ID do usuário e senha que são gerenciados por meio de um diretório LDAP corporativo é fornecida pelo produto LDAP que é usado com a plataforma IBM Cloud Private Cloud Foundry.

Os dados pessoais que são persistidos pela criação de log e monitoramento da plataforma consistem em endereços IP de máquinas virtuais e alguns IDs do usuário. Os aplicativos desenvolvidos pelo usuário ou fornecidos pelo usuário podem incluir outros dados pessoais em seu uso de criação de log e monitoramento. Os mesmos mecanismos que são usados para exclusão de dados de criação de log e monitoramento do sistema podem ser usados para dados de criação de log e monitoramento do aplicativo. Os dados pessoais que são coletados por aplicativos fora desses serviços requerem mecanismos fornecidos pelo aplicativo para excluir dados. Para obter mais informações, consulte

- IBM Cloud Private log
- IBM Cloud Private Cloud Foundry Serviço de monitoramento
- Documentação do Prometheus

Monitoramento de Dados

- Opcional: a plataforma IBM Cloud Private fornece um serviço de monitoramento para monitorar o status de seu IBM Cloud Private Cloud Foundry e aplicativos. Este serviço usa o Grafana e o Prometheus para apresentar informações detalhadas sobre nós do cluster e contêineres. O monitoramento pode ser configurado para gerar alertas ou integrado a provedores de alerta externos. Plataforma de monitoramento é ativada por padrão. Pilhas adicionais de monitoramento podem ser

implementadas para monitoramento de aplicativo. Para obter mais informações, consulte IBM Cloud Private Monitoring Service e IBM Cloud Private Cloud FoundryMonitoramento de cluster.
- É possível criar e exibir seu próprio serviço de monitoramento do Prometheus.
- Opcional: a plataforma IBM Cloud Private fornece um serviço de criação de log que é baseado na pilha ELK para logs de fluxo, de armazenamento, de procura e de monitoramento. A pilha do ELK que é fornecida com a plataforma IBM Cloud Private usa as imagens da pilha do ELK oficial que são publicadas pelo Elastic. Pilhas adicionais de ELK podem ser implementadas para criação de log de aplicativo. Para obter mais informações, consulte Criação de log do IBM Cloud Private.
- É possível criar e exibir seu próprio serviço de criação de log ELK.
- A criação de log é configurada por padrão para coletar logs do sistema para a plataforma IBM Cloud Private Cloud Foundry usando syslog.

Capacidade para restringir o uso de dados pessoais

Usando os recursos que são resumidos neste documento, a plataforma IBM Cloud Private Cloud Foundry permite que um usuário final restrinja o uso de quaisquer dados técnicos dentro da plataforma que é considerada dados pessoais.

Sob o GDPR, os usuários têm direitos para acessar, modificar e restringir o processamento. Consulte as outras seções deste documento para controlar o seguinte:

- Direito de acesso
  - Os administradores da plataforma IBM Cloud Private Cloud Foundry podem usar os recursos da plataforma IBM Cloud Private Cloud Foundry para fornecer aos indivíduos o acesso aos seus dados.
  - Os administradores da plataforma IBM Cloud Private Cloud Foundry podem usar os recursos da plataforma IBM Cloud Private Cloud Foundry para fornecer aos indivíduos as informações sobre quais dados a plataforma IBM Cloud Private Cloud Foundry retém sobre o indivíduo.
- Certo modificar
  - Os administradores da plataforma IBM Cloud Private Cloud Foundry podem usar os recursos da plataforma IBM Cloud Private Cloud Foundry para permitir que um indivíduo modifique ou corrija seus dados.
  - Os administradores da plataforma IBM Cloud Private Cloud Foundry podem usar os recursos da plataforma IBM Cloud Private Cloud Foundry para corrigir os dados de um indivíduo para eles.
- Certo para restringir o processamento
  - Os administradores da plataforma IBM Cloud Private Cloud Foundry podem usar os recursos da plataforma IBM Cloud Private Cloud Foundry para parar o processamento de dados de um indivíduo.

Apêndice - dados registrados pela plataforma do IBM Cloud Private Cloud Foundry

Como plataforma, o IBM Cloud Private Cloud Foundry lida com várias categorias de dados técnicos que podem ser considerados como dados pessoais, como um ID do usuário administrador e senha padrão, IDs de usuário e senhas de serviço, endereços IP e nomes de organizações/espaços. A plataforma IBM Cloud Private Cloud Foundry também lida com informações sobre usuários que gerenciam a plataforma. Os aplicativos que são executados na plataforma pode apresentar outras categorias de dados pessoais que são desconhecidos para a plataforma.

Este apêndice inclui detalhes sobre dados que são registrados pelos serviços da plataforma.

IBM Cloud Private Cloud Foundry segurança

- O dados são registrados
  - ID do usuário e endereço IP de usuários com login efetuado
- Quando os dados são registrados
  - Com pedidos de login
- Onde dados são registrados
  - Nos logs de auditoria no `/var/vcap/sys/log` **padrão**
- Como excluir dados
  - Procure por dados específicos do usuário e exclua o registro do log

IBM Cloud Private Cloud Foundry plataforma da API

- O dados são registrados
  - ID do usuário e endereço IP de usuários com login efetuado
- Quando os dados são registrados
  - Com cada solicitação de API (dependente de nível de log)
- Onde dados são registrados

    - syslog é o padrão
    - Serviço de criação de log externo que você configurou
- Como excluir dados
    - Procure pelos logs no `/var/vcap/sys/log` na máquina virtual `cc_core`
    - Se o ELK estiver sendo usado, procure pelos logs ELK e remova as entradas apropriadas

IBM Cloud Private Cloud Foundry monitorando OPTIONALLY ativado

- O dados são registrados
    - Endereço IP, nome do ambiente, nome da implementação, liberação, stemcell
    - Dados extraídos de aplicativos desenvolvidos pelo cliente podem incluir dados pessoais
- Quando os dados são registrados
    - Quando o Prometheus extrai métricas de destinos configurados
- Onde dados são registrados
    - No servidor Prometheus ou volumes persistentes configurados
- Como excluir dados
    - Procure e exclua dados usando a API do Prometheus

Para obter mais informações, consulte: Documentação do Prometheus e Gerenciamento de logs e métricas para o Prometheus. Você pode usar sua própria Prometheus. Por padrão, nenhum dado de monitoramento é capturado.

IBM Cloud Private Cloud Foundry Cloud Foundry

- O dados são registrados
    - Informações sobre a operação da plataforma
    - Configuração da Plataforma
    - ID do usuário nas tarefas de API, UAA e login
- Quando os dados são registrados
    - Informações em tempo real sobre o funcionamento e a operação do sistema
- Onde dados são registrados
    - O padrão é `/var/vcap/sys/log`
    - O usuário forneceu um serviço de criação de log alternativo
    - O Alternativo é IBM Cloud Private Cloud Foundry ELK
- Como excluir dados
    - Limpe o `/var/vcap/sys/log`
    - Um serviço de criação de log alternativo fornecido pelo usuário foi excluído

IBM Cloud Private considerações de plataforma para preparação de PCI

O Payment Card Industry Data Security Standard (PCI DSS) é uma coleção de objetivos e requisitos correspondentes para proteção de um ambiente de dados do titular do cartão. O ambiente de dados do titular do cartão, conforme definido pelo PCI Security Standards Council, representa "pessoas, processos e tecnologia que armazenam, processam ou transmitem dados do titular do cartão ou que afetam a segurança dos dados do titular do cartão". O DSS é dividido em 6 objetivos de controle e 12 requisitos de nível superior.

A IBM contratou uma empresa QSA de terceiros, Weaver (que trabalha com vários aspectos do IBM Cloud) para revisar a plataforma IBM® Cloud Private (ICP) e desenvolver diretrizes de PCI para usuários do ICP. O resultado é um white paper que descreve considerações e orientações para organizações que estão considerando a plataforma IBM Cloud Private e como ela pode ajudar a suportar a implementação de requisitos do PCI DSS 3.2.1. Cada cliente é responsável por determinar se o ambiente e a configuração do IBM Cloud Private atendem aos requisitos do Payment Card Industry Data Security Standard (PCI DSS) 3.2.1.

Para obter mais informações, consulte o Guia de implementação do IBM Cloud Private Platform PCI DSS 3.2.1.

Tarefas de Instalação

Use as seguintes informações para instalar o Cloud Foundry Enterprise Environment em contêineres do Kubernetes ou para instalar o IBM Cloud Private Cloud Foundry na instância do VMware vSphere, do AWS ou do OpenStack.

- Instalando o Cloud Foundry Enterprise Environment
- Instalando o IBM Cloud Private Cloud Foundry

Instalando o IBM Cloud Private Cloud Foundry

Deve-se preparar seu nuvem antes de instalar o IBM® Cloud Private Cloud Foundry. É possível instalar o IBM Cloud Private Cloud Foundry em nuvens privadas do VMware vSphere, no AWS ou no OpenStack.

- Preparando-se para instalar o IBM Cloud Private Cloud Foundry
- Instalando o IBM Cloud Private Cloud Foundry

Preparando para Instalar o IBM® Cloud Private Cloud Foundry

Deve-se preparar seu nuvem antes de instalar o IBM Cloud Private Cloud Foundry.

É possível instalar o IBM Cloud Private Cloud Foundry em nuvens privadas do VMware vSphere, no AWS ou no OpenStack.

- Componentes do IBM Cloud Private Cloud Foundry
- Requisitos do VMware para o IBM Cloud Private Cloud Foundry
- Requisitos do OpenStack para o IBM Cloud Private Cloud Foundry
- Fornecendo certificados para o IBM Cloud Private Cloud Foundry
- Configurando o DNS para o IBM® Cloud Private Cloud Foundry
- Tamanho da implementação do AWS para instalação corporativa do IBM® Cloud Private Cloud Foundry
- Fazendo upgrade de versões secundárias de stemcell

Requisitos do VMware para IBM Cloud Private Cloud Foundry

É possível instalar o IBM® Cloud Private Cloud Foundry em sua instância do VMware vSphere 5.5 ou 6.X, não importa se instância está conectada ou não à Internet.

- Informações necessárias sobre sua instância do VMware
- Configurando permissões do VMware
- Requisitos de tamanho do VMware para instalação do desenvolvedor do IBM Cloud Private Cloud Foundry
- Requisitos de tamanho do VMware para instalação corporativa do IBM Cloud Private Cloud Foundry

Informações necessárias sobre sua instância do VMware

Enquanto você se prepara para instalar o IBM® Cloud Private Cloud Foundry no VMware, é necessário entender as informações de configuração a seguir sobre as instâncias do VMware.

Informações necessárias do VMware vSphere

- As versões 5.5, 6.0 e 6.5 do vSphere são suportadas.
- Endereço IP do vCenter.
- Nome do usuário do vCenter.
- Senha do vCenter.
- O nome do data center que é exibido no vSphere Client, que é o pai para a implementação.
- O nome do cluster que é exibido no vSphere Client, que é o filho do data center.
- (Opcional) O nome do conjunto de recursos que é exibido no vSphere Client, que é o filho do cluster.
- Os armazenamentos de dados que hospedam stemcells e todas as máquinas virtuais.
- O nome da pasta para discos persistentes.
- O nome do grupo da porta em que o BOSH Director e as máquinas virtuais do Cloud Foundry estão conectados.
- O nome da pasta da máquina virtual que é o pai para as máquinas virtuais do BOSH Director e do Cloud Foundry (visualização Máquinas virtuais e modelos do vSphere Client).

Informações de rede necessárias

- Sub-rede a ser usada, por exemplo `172.12.34.0/24`.
- Gateway da sub-rede especificada.
- Servidores DNS. <! -- É possível listar uma ou mais vírgulas separadas sem espaços) -- >
- Servidores NTP. <! -- (uma ou mais vírgulas separadas sem espaços) -- >

- Endereços IP da sub-rede escolhida. Para instalações que usam o modo Desenvolvedor, são necessários 25 endereços IP. Para instalações que usam o modo corporativo, são necessários 40 endereços IP.

Rotas de rede necessárias

- O instalador deve ser capaz de atingir a instância do vCenter e todos os hosts ESXi no cluster escolhido usando um endereço IP ou o nome completo do domínio (FQDN) que é mostrado no vCenter. Se os hosts ESXi não forem atingíveis, a implementação falhará.
- O instalador implementa um diretor BOSH. O diretor usa as mesmas rotas do vCenter e do ESXi que o instalador.
- Suas implementações requerem rotas adicionais para LDAP, proxies e serviços.

Certificados necessários

Os certificados curinga autoassinados são gerados durante a instalação. Também é possível fornecer seus próprios certificados.

Configurando permissões do VMware

Se você não usar a conta do usuário raiz ou do Administrador, deverá concluir as etapas a seguir para configurar as permissões antes de instalar o IBM® Cloud Private Cloud Foundry no VMware.

1. Crie duas funções.
   - Conceda à primeira função os privilégios a seguir:
     - Armazenamento de dados: **operações de arquivo de baixo nível**
     - Armazenamento de dados: **atualizar arquivos da máquina virtual**
     - vApp: **importar**
   - Conceda à segunda função o privilégio a seguir:
     - Global: **gerenciar atributos customizados**
   - Se você usar a `Rede Virtual Distributed Switch`, conceda à função o privilégio a seguir:
     - Grupo dvPort: **modificar**
2. Crie um usuário do vCenter.
3. Para usar as novas funções, designe ao usuário do vCenter as funções apropriadas para os componentes do vSphere que estão listadas na Tabela 1.
4. Se você usar uma rede vSwitch, designe a função de `administrator` ao grupo de portas apropriado. Assegure-se de que **Propagar para objetos-filhos** não esteja selecionado.
5. Se você usar uma Rede Virtual Distributed Switch (vDS), conclua as etapas a seguir:
   - Coloque o comutador vDS em uma pasta.
   - Designe à pasta pai vDS a função `Read-only` para o novo usuário e selecione **Propagar para objetos-filhos**.
   - Designe a função de `administrator` para o grupo de portas apropriado. Assegure-se de que **Propagar para objetos-filhos** não esteja selecionado.

Tabela 1. Funções e permissões do usuário do VMware

Visualização do Cliente vSphere	Componente do vSphere	Função	Outras
Hosts e Clusters	vCenter	A segunda função definida pelo usuário	Sem propagação
Hosts e Clusters	Data Center	A primeira função definida pelo usuário	Sem propagação
Hosts e Clusters	Grupo	Administrador	Propagada
VMs e Modelos	Pasta da máquina virtual	Administrador	Propagada
Armazenamentos de Dados e Clusters de Armazenamento de Dados	Cada armazenamento de dados	Administrador	Propagada

Requisitos de tamanho do VMware para instalação do desenvolvedor IBM® Cloud Private Cloud Foundry

Para instalar a versão de desenvolvedor do IBM Cloud Private Cloud Foundry no VMware, sua instância do vSphere deve atender aos requisitos de tamanho a seguir.

Os requisitos de armazenamento de dados do vSphere supõem que, no `uiconfig_vmware_template.yml`, o valor do parâmetro `vmware_disk_type` é configurado como `preallocated`. Ao usar o valor padrão do `vmware_disk_type` de `thin`, a implementação inicial precisará de menos espaço de armazenamento de dados. No entanto, a quantia de espaço de armazenamento de dados que é necessária aumenta ao longo do tempo e pode atingir o tamanho especificado.

Cada célula do desenvolvedor padrão usa os recursos a seguir:

```
vCPU: 4
Memory: 32768 MB
Storage: 360.8 GB = 369,424 MB = (32768 MB vSphere swap file + 300,000 MB ephemeral disk + 3072 MB
system disk)x1.10 = (MEM+STORAGE) x OVERHEAD
```

Dimensionamento do modo de desenvolvedor para VMware

Total geral

O armazenamento mínimo total é 2058.258 GB

A memória mínima total é de 142.000 GB

O vCPU total mínimo é 38.

Total geral por implementação

Tabela 1. Total geral por implementação

Implementações	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Gerência de Relações Comerciais e Industriais da IBM Brasil	1	4	8192	51200	102400	27750
Cloud Foundry	18	26	122880	563200	378880	290190
PCP UI	1	2	8192	24576	65536	21401
Compilação do trabalhador	3	6	6144	24288	0	19323
Subtotal	23	38	145408	1202176	546816	358664
Célula Diego	Fornecida pelo Usuário	[Instâncias] x 4	[Instâncias] x 32.000	[Instâncias] x 300.000		
TOTAL geral (subtotal mais total de células)						

Dimensionamento detalhado

Dimensionamento do diretor

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Bosh	1	4	8192	51200	102400	27750
TOTAL	1	4	8192	51200	102400	27750

Dimensionamento do Cloud Foundry

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
nfs_WAL_server	1	2	8192	51200	307200	48230
Nats	1	1	4096	10240	0	8908
adaptador	1	1	4096	10240	0	8908
Banco de dados	1	2	8192	10240	10240	14438
diego-api	1	2	8192	10240	0	13414
uaa	1	1	4096	10240	0	8908
singleton-blobstore	1	2	8192	10240	51200	18534
api	1	2	8192	51200	0	17510
cc-trabalhador	1	1	4096	10240	0	8908

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
roteador	1	1	4096	10240	0	8908
tcp-router	1	1	4096	10240	0	8908
programador	1	1	4096	10240	0	8908
doppler	1	1	4096	10240	0	8908
diego-cell	1	4	32768	307200	0	70144
log-api	1	1	4096	10240	0	8908
credhub	1	1	4096	10240	0	8908
rotate-cc-database-key	1	1	4096	10240	0	8908
Backup-restore	1	1	4096	10240	10240	9932
TOTAL	18	26	122880	563200	378880	290190

Dimensionamento da UI da PCP

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Docker	1	2	8192	24576	65536	21401
TOTAL	1	2	8192	24576	65536	21401

Dimensionamento dos trabalhadores de compilação

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Compilação do trabalhador	3	2	2048	8096	0	6441
TOTAL	3	6	6144	24288	0	19323

Requisitos de tamanho do VMware para a instalação corporativa do IBM Cloud Private Cloud Foundry

Para instalar a versão corporativa do IBM® Cloud Private Cloud Foundry no VMware, sua instância do vSphere deverá atender aos requisitos de tamanho a seguir.

Sob cargas normais de aplicativo, esse modelo de infraestrutura suporta uma memória de célula de até 1 TB.

Além desses requisitos, é preciso fornecer recursos de célula para aplicativos de host para obter uma infraestrutura de alta disponibilidade. É possível modificar esses requisitos de tamanho para atender aos seus requisitos de aplicativos e perfis de hardware.

Os requisitos de armazenamento de dados do vSphere supõem que, no `uiconfig_vmware_template.yml`, o valor do parâmetro `vmware_disk_type` é configurado como `preallocated`. Ao usar o valor padrão do `vmware_disk_type` de `thin`, a implementação inicial precisará de menos espaço de armazenamento de dados. No entanto, a quantia de espaço de armazenamento de dados que é necessária aumenta ao longo do tempo e pode atingir o tamanho especificado.

Cada célula padrão usa os recursos a seguir:

```
vCPU: 4
Memory: 32768 MB
Storage: 360.8 GB = 369,424 MB = (32768 MB vSphere swap file + 300,000 MB ephemeral disk + 3072 MB
system disk)x1.10 = (MEM+STORAGE) x OVERHEAD
```

Dimensionamento do modo corporativo para VMware

Total geral

O armazenamento mínimo total é 2620428 GB.

A memória mínima total é de 210.000 GB.

O vCPU total mínimo é 58.

Total geral por implementação

Tabela 1. Total geral por implementação

Implementações	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Gerência de Relações Comerciais e Industriais da IBM Brasil	1	4	8192	51200	102400	27750
Cloud Foundry	32	42	188416	747520	430080	433130
PCP UI	1	2	8192	24576	65536	21401
Compilação do trabalhador	5	10	10240	40480	0	32205
Subtotal	39	58	215040	1570816	598016	514486
Célula Diego	Fornecida pelo Usuário	[Instâncias] x 4	[Instâncias] x 32.000	[Instâncias] x 300.000		
TOTAL geral (subtotal mais total de células)						

Grande TOTAL

Dimensionamento detalhado

Dimensionamento do diretor

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Bosh	1	4	8192	51200	102400	27750
TOTAL	1	4	8192	51200	102400	27750

Dimensionamento do Cloud Foundry

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
nfs_WAL_server	1	2	8192	51200	307200	48230
Nats	2	1	4096	10240	0	8908
adaptador	2	1	4096	10240	0	8908
Banco de dados	1	2	8192	10240	10240	14438
diego-api	2	2	8192	10240	0	13414
uaa	2	1	4096	10240	0	8908
singleton-blobstore	1	2	8192	10240	102400	23654
api	2	2	8192	51200	0	17510
cc-trabalhador	2	1	4096	10240	0	8908
roteador	2	1	4096	10240	0	8908
tcp-router	2	1	4096	10240	0	8908
programador	2	1	4096	10240	0	8908
doppler	4	1	4096	10240	0	8908
diego-cell	1	4	32768	307200	0	70144
log-api	2	1	4096	10240	0	8908
credhub	2	1	4096	10240	0	8908
rotate-cc-database-key	1	1	4096	10240	0	8908
Backup-restore	1	1	4096	10240	10240	9932
TOTAL	32	42	188416	747520	430080	433130

Dimensionamento da UI da PCP

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Docker	1	2	8192	24576	65536	21401
TOTAL	1	2	8192	24576	65536	21401

Dimensionamento dos trabalhadores de compilação

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Compilação do trabalhador	5	2	2048	8096	0	6441
TOTAL	5	10	10240	40480	0	32205

Requisitos do OpenStack para IBM Cloud Private Cloud Foundry

É possível instalar o IBM® Cloud Private Cloud Foundry em sua instância do OpenStack, independentemente se ela estiver conectada à Internet ou não.

- Informações necessárias sobre sua instância do OpenStack
- Requisitos de tamanho OpenStack para IBM Cloud Private Cloud Foundry desenvolvedor
- Requisitos de tamanho do OpenStack para instalação corporativa do IBM Cloud Private Cloud Foundry

Informações necessárias sobre sua instância do OpenStack

Enquanto você se prepara para instalar o IBM® Cloud Private Cloud Foundry no OpenStack, é necessário entender as informações de configuração a seguir sobre a instância do OpenStack.

Entendendo os parâmetros OpenStack necessários

Para fazer download de informações detalhadas sobre os parâmetros a seguir, use a interface com o usuário do OpenStack. Para versões mais antigas do OpenStack, como Liberty ou Mitaka, selecione **Projeto** > **Calcular** > **Acesso e segurança**

Acesso de API > **Download do OpenStack RC File v3**. Para versões mais recentes do OpenStack, como o Pike, selecione **Projeto** > **Acesso de API** > **Download do OpenStack RC File** > **Identity API v3** ou **Projeto** > **Acesso de API Download do OpenStack RC File** > **Arquivo Clouds.yaml**:

- API versão 3 (obrigatório)
- URL de autenticação de OpenStack
- Nome de usuário do OpenStack
- Senha do OpenStack
- Versão da API
- Nome de domínio de usuário
- Nome do Projeto
- ID do Projeto
- Nome da região
- Interface
- ID da Rede
- Zona de Disponibilidade
- Certificado de CA do OpenStack
- Nome de par de chaves do OpenStack
- Chave privada do par de chaves do Openstack
- Grupos de segurança do Openstack

Configurando tipos necessários

A implementação do IBM Cloud Private Cloud Foundry precisa de tamanhos específicos de CPU, memória e disco para os vários tipos de máquinas virtuais. No OpenStack, esses recursos são definidos como tipos. Por padrão, o IBM Cloud Private Cloud Foundry cria os tipos durante a instalação. Se não for possível criar novos tipos, conclua as etapas a seguir para fornecer um mapeamento para os tipos:

1. Acesse o arquivo `/<installation_directory>/uiconfig_openstack_template.yml`. Localize o **flavors** chave.

```
# flavors:
# icpcf_cfp_ui: <your_flavor>
# icpcf_compilation: <your_flavor>
# icpcf_director: <your_flavor>
# icpcf_minimal: <your_flavor>
# icpcf_small: <your_flavor>
# icpcf_small-highmem: <your_flavor>
```

2. Remova o comentário da linha `# flavors:` e de todas as chaves filhas removendo o sinal de número (#).

3. Para cada uma das chaves de tipo, forneça um mapeamento para um tipo existente do OpenStack que corresponda mais estritamente aos requisitos de CPU, memória e disco para esse tipo de instância.

Para obter mais informações sobre os requisitos de tamanho do OpenStack, consulte Requisitos de tamanho do OpenStack para a instalação de desenvolvedor do IBM Cloud Private Cloud Foundry e Requisitos de tamanho do OpenStack para a instalação corporativa do IBM Cloud Private Cloud Foundry.

Configurando pares de chaves necessários

Na interface com o usuário do OpenStack, selecione o **Domínio** e o **Projeto** adequados:

1. Navegue para **Pares de chaves**
   - Para as versões mais antigas do OpenStack, como Liberty ou Mitaka:
     1. Selecione **Projeto**.
     2. Selecione **Cálculo**.
     3. Selecione **Acesso e Segurança**.
     4. Selecione o **de** guia.
   - Para as versões mais recentes do OpenStack, como Pike:
     1. Selecione **Projeto**.
     2. Selecione **Cálculo**.
     3. Selecione **Pares de chave**.
2. Clique em **Criar Par de Chave** .
3. Nomeie o par de chaves **bosh** e clique em **Criar par de chaves**.
4. Salve o arquivo `bosh.pem`.

AVISO: se você estiver usando o OpenStack versão Liberty ou Mitaka, não crie o par de chaves com o painel do OpenStack Horizon. Isso é devido a um erro do OpenStack. Em vez disso, assegure-se de gerar o par de chaves SSH manualmente. Por exemplo, use o comando ssh-keygen:

```
ssh-keygen -t rsa -b 4096 -C "bosh" -f bosh.key
```

Importe esse par de chaves no OpenStack usando a interface com o usuário do OpenStack. Selecione **Projeto** > **Calcular** > **Acesso e segurança** > **Pares de chaves** > **Importar par de chaves**.

Criando grupos de segurança necessários

Na interface com o usuário do OpenStack, com o **Domínio** e o **Projeto** adequados selecionados, conclua o procedimento a seguir:

1. Navegue para **Grupos de segurança**:
   - Para as versões mais antigas do OpenStack, como Liberty ou Mitaka:
     1. Selecione **Projeto**.
     2. Selecione **Cálculo**.
     3. Selecione **Acesso e Segurança**.
     4. Selecione **Grupos de Segurança**.
   - Para as versões mais recentes do OpenStack, como Pike:
     1. Selecione **Projeto**.
     2. Selecione **Rede**.
     3. Selecione **Grupos de Segurança**.
2. Clique em **Criar Grupo de Segurança**.
3. Nomeie o grupo de segurança **bosh** e inclua a descrição **Grupo de segurança do BOSH**.
4. Clique em **Criar Grupo de Segurança**.
5. Selecione o **Grupo de segurança BOSH** e clique em **Editar regras**.
6. Clique em **Incluir regra**
7. Inclua as regras a seguir no **Grupo de segurança BOSH**:

Regra	Orientação	Tipo Ether	Protocolo IP	Porta ou Intervalo	Remota	Propósito
Regra TCP customizada	Ingresso	IPv4	TCP	22	0.0.0.0/0 (CIDR)	Acesso SSH por meio da CLI
Regra TCP customizada	Ingresso	IPv4	TCP	80	0.0.0.0/0 (CIDR)	acesso a Gorouters
Regra TCP customizada	Ingresso	IPv4	TCP	443	0.0.0.0/0 (CIDR)	acesso a Gorouters

Regra	Orientação	Tipo Ether	Protocolo IP	Porta ou Intervalo	Remota	Propósito
Regra TCP customizada	Ingresso	IPv4	TCP	2222	0.0.0.0/0 (CIDR)	Cloud Foundry SSH
Regra TCP customizada	Ingresso	IPv4	TCP	4222	0.0.0.0/0 (CIDR)	NATS
Regra TCP customizada	Ingresso	IPv4	TCP	4443	0.0.0.0/0 (CIDR)	Console do Cloud Foundry
Regra TCP customizada	Ingresso	IPv4	TCP	6868	0.0.0.0/0 (CIDR)	BOSH Agent para autoinicialização
Regra TCP customizada	Ingresso	IPv4	TCP	8443	0.0.0.0/0 (CIDR)	UAA API
Regra TCP customizada	Ingresso	IPv4	TCP	25250	0.0.0.0/0 (CIDR)	Blobstore
Regra TCP customizada	Ingresso	IPv4	TCP	25555	0.0.0.0/0 (CIDR)	BOSH Director API
Regra TCP customizada	Ingresso	IPv4	TCP	25777	0.0.0.0/0 (CIDR)	Registro
Regra TCP customizada	Ingresso	IPv4	TCP	1-65535	bosh (grupo de segurança)	Gerenciamento e acesso a dados
Regra TCP customizada	Egresso	IPv4	TCP	1-65535	bosh (grupo de segurança)	Gerenciamento e acesso a dados
Todo ICMP	Ingresso	IPv4	ICMP	Qualquer um	bosh (grupo de segurança)	
Regra TCP customizada	Egresso	IPv4	Qualquer um	-	0.0.0.0/0 (CIDR)	
Regra TCP customizada	Egresso	IPv6	Qualquer um	-	::/0 (CIDR)	
Todo UDP	Ingresso	IPv4	UDP	1-65535	0.0.0.0/0 (CIDR)	

Alocando endereços IP flutuantes necessários para o Director, os Gorouters e a interface com o

usuário do console O BOSH Director, o Gorouters e a interface com o usuário do console requerem, cada um, um endereço IP flutuante. Na interface com o usuário do OpenStack, com o **Domínio** e o **Projeto** adequados selecionados, conclua o procedimento a seguir:

1. Navegue para **Grupos de segurança**:
    - Para as versões mais antigas do OpenStack, como Liberty ou Mitaka:
        1. Selecione **Projeto**.
        2. Selecione **Cálculo**.
        3. Selecione **Acesso e Segurança**.
        4. Selecione a guia **IPs flutuantes**.
    - Para as versões mais recentes do OpenStack, como Pike:
        1. Selecione **Projeto**.
        2. Selecione **Rede**.
        3. Selecione **IPs flutuantes**.
2. Selecione **Alocar IP para o Projeto** .
3. Selecione **Externo** no menu **Conjunto**.
4. Clique em **Alocar IP**.
5. Quando o IP é alocado, anote o endereço IP que é exibido na janela pop-up.
6. Repita o processo para alocar o número de endereços IP flutuantes necessários.

Informações de rede necessárias

É necessário fornecer as informações de rede a seguir.

- Sub-rede a ser usada, por exemplo `10.10.25.0/24`. Deve-se fornecer um intervalo de pelo menos 64 endereços IP.
- Gateway da sub-rede especificada
- Servidores DNS <! -- Você pode listar uma ou mais vírgulas separadas sem espaços) -- >
- Servidores NTP <! -- (uma ou mais vírgulas separadas sem espaços) -- >

- Endereços IP da sub-rede escolhida. Você precisa de 25 endereços IP para instalações de desenvolvedor e 40 endereços IP para instalações corporativas.

Entendendo rotas de rede necessárias

- O instalador deve ser capaz de atingir a instância do OpenStack usando um endereço IP ou o nome completo do domínio (FQDN) que é mostrado no vCenter.
- O instalador implementa um diretor BOSH. O diretor usa as mesmas rotas do OpenStack que o instalador.
- Suas implementações requerem rotas adicionais para LDAP, proxies e serviços.

Certificados necessários

Os certificados curinga autoassinados são gerados durante a instalação. Também é possível fornecer seus próprios certificados.

Requisitos de tamanho OpenStack para IBM Cloud Private Cloud Foundry instalação do desenvolvedor

Para instalar uma instalação corporativa do IBM® Cloud Private Cloud Foundry no OpenStack, sua instância do OpenStack deve atender aos requisitos de tamanho a seguir.

Sob cargas normais de aplicativo, esse modelo de infraestrutura suporta uma memória de célula de até 1 TB.

Além desses requisitos, é preciso fornecer recursos de célula para aplicativos de host para obter uma infraestrutura de alta disponibilidade. É possível modificar esses requisitos de tamanho para atender aos seus requisitos de aplicativos e perfis de hardware.

Cada célula padrão usa os recursos a seguir:

```
vCPU: 4 Memória: 32768 MB Armazenamento: 376,2 GB = 385.228 MB = (317.440 MB de disco persistente + 67.788 MB de sobrecarga)
```

Dimensionamento do modo de desenvolvedor para OpenStack

Total geral

O armazenamento mínimo total é 2321.052 GB.

A memória total mínima é 140,605 GB.

O vCPU total mínimo é 42.

Total geral por implementação

Tabela 1. Total geral por implementação

Implementações	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Gerência de Relações Comerciais e Industriais da IBM Brasil	1	4	8192	34816	102400	22732
Cloud Foundry	18	26	117356	729088	378880	239875
PCP UI	1	2	8192	34816	65536	19046
Compilação do trabalhador	5	10	10240	92160	0	20480
Subtotal	25	42	143980	1527808	546816	302133
Célula Diego	Fornecida pelo Usuário	[Instâncias] x 4	[Instâncias] x 32.000	[Instâncias] x 300.000		
TOTAL geral (subtotal mais total de células)						

Dimensionamento detalhado

Dimensionamento do diretor

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Bosh	1	4	8192	34816	102400	22732
TOTAL	1	4	8192	34816	102400	22732

Dimensionamento do Cloud Foundry

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
nfs_WAL_server	1	2	7680	51200	307200	44288
Nats	1	1	3849	20480	0	6281
adaptador	1	1	3849	20480	0	6281
Banco de dados	1	2	7680	24576	10240	11929
diego-api	1	2	7680	24576	0	10905
uaa	1	1	3849	20480	0	6281
singleton-blobstore	1	2	7680	24576	51200	16025
api	1	2	7680	51200	0	13568
cc-trabalhador	1	1	3849	20480	0	6281
roteador	1	1	3849	20480	0	6281
tcp-router	1	1	3849	20480	0	6281
programador	1	1	3849	20480	0	6281
doppler	1	1	3849	20480	0	6281
diego-cell	1	4	32768	307200	0	66764
log-api	1	1	3849	20480	0	6281
credhub	1	1	3849	20480	0	6281
rotate-cc-database-key	1	1	3849	20480	0	6281
Backup-restore	1	1	3849	20480	10240	7305
TOTAL	18	26	117356	729088	378880	239875

Dimensionamento da UI da PCP

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Docker	1	2	8192	34816	65536	19046
TOTAL	1	2	8192	34816	65536	19046

Dimensionamento dos trabalhadores de compilação

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Compilação do trabalhador	5	2	2048	18432	0	4096
TOTAL	5	10	10240	92160	0	20480

Requisitos de tamanho do OpenStack para instalação corporativa do IBM Cloud Private Cloud Foundry

Para instalar uma instalação corporativa do IBM® Cloud Private Cloud Foundry no OpenStack, sua instância do OpenStack deve atender aos requisitos de tamanho a seguir.

Sob carga de aplicativo normal, esse modelo de infraestrutura suporta memória celular de até 1 TB.

Deve-se fornecer recursos da célula para hospedar aplicativos além desses requisitos para a infraestrutura de alta disponibilidade. É possível modificar esses requisitos de tamanho para atender aos seus requisitos de aplicativos e perfis de hardware.

Configuração de célula padrão:

```
vCPU: 4 Memória: 32768 MB Armazenamento: 376,2 GB = 385.228 MB = (317.440 MB de disco persistente +
67.788 MB de sobrecarga)
```

Dimensionamento do modo corporativo para o OpenStack

Total geral

O armazenamento mínimo total é 3101.557 GB.

A memória mínima total é de 200.711 GB.

O vCPU total mínimo é 58.

Total geral por implementação

Tabela 1. Total geral por implementação

Implementações	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Gerência de Relações Comerciais e Industriais da IBM Brasil	1	4	8192	34816	102400	22732
Cloud Foundry	32	42	178904	1050624	430080	344840
PCP UI	1	2	8192	34816	65536	19046
Compilação do trabalhador	5	10	10240	92160	0	20480
Subtotal	39	58	205528	2170880	598016	407098
Célula Diego	Fornecida pelo Usuário	[Instâncias] x 4	[Instâncias] x 32.000	[Instâncias] x 300.000		
TOTAL geral (subtotal mais total de células)						

Dimensionamento detalhado

Dimensionamento do diretor

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Bosh	1	4	8192	34816	102400	22732
TOTAL	1	4	8192	34816	102400	22732

Dimensionamento do Cloud Foundry

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
nfs_WAL_server	1	2	7680	51200	307200	44288
Nats	2	1	3849	20480	0	6281
adaptador	2	1	3849	20480	0	6281
Banco de dados	1	2	7680	24576	10240	11929
diego-api	2	2	7680	24576	0	10905
uaa	2	1	3849	20480	0	6281
singleton-blobstore	1	2	7680	24576	102400	21145
api	2	2	7680	51200	0	13568
cc-trabalhador	2	1	3849	20480	0	6281
roteador	2	1	3849	20480	0	6281
tcp-router	2	1	3849	20480	0	6281
programador	2	1	3849	20480	0	6281
doppler	4	1	3849	20480	0	6281
diego-cell	1	4	32768	307200	0	66764
log-api	2	1	3849	20480	0	6281
credhub	2	1	3849	20480	0	6281
rotate-cc-database-key	1	1	3849	20480	0	6281
Backup-restore	1	1	3849	20480	10240	7305

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
TOTAL	32	42	178904	1050624	430080	344840

Dimensionamento da UI da PCP

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Docker	1	2	8192	34816	65536	19046
TOTAL	1	2	8192	34816	65536	19046

Dimensionamento dos trabalhadores de compilação

Tarefas da VM	Instâncias	vCPU	Memória	Disco da VM (MB)	Disco persistente (MB)	Disco de sobrecarga (MB)
Compilação do trabalhador	5	2	2048	18432	0	4096
TOTAL	5	10	10240	92160	0	20480

Requisitos do AWS para IBM Cloud Private Cloud Foundry

É possível instalar o IBM® Cloud Private Cloud Foundry em sua instância do AWS, independentemente se ela estiver ou não conectada à Internet.

- Tamanho da implementação do AWS para a instalação corporativa do IBM Cloud Private Cloud Foundry

Tamanho da implementação do AWS para a instalação corporativa do IBM Cloud Private Cloud Foundry

A instalação do IBM® Cloud Private Cloud Foundry no AWS cria uma instalação corporativa com os requisitos de tamanho a seguir.

Sob cargas normais de aplicativo, esse modelo de infraestrutura suporta uma memória de célula de até 1 TB.

Além desses requisitos, é preciso fornecer recursos de célula para aplicativos de host para obter uma infraestrutura de alta disponibilidade. É possível modificar esses requisitos de tamanho para atender aos seus requisitos de aplicativos e perfis de hardware.

Cada célula padrão usa os recursos a seguir:

```
AWS AMI: t2.medium, disk size: 300GB
```

Dimensionamento do modo corporativo para o AWS

Total geral por implementação

Implementações	Instâncias	Tipo AMI	Disco da VM (MB)	Disco persistente (MB)
Gerência de Relações Comerciais e Industriais da IBM Brasil	1	m4.xlarge	25000	102400
Cloud Foundry	31		87000	419600
PCP UI	1	t2.medium	30000	65536
Compilação do trabalhador	5	t2.micro	50000	0

Subtotal|38||192000|587536| Célula Diego|Fornecido pelo usuário|[Instâncias] x t2.2xlarge|[Instâncias] x 300.000|0 **TOTAL geral** (subtotal mais o total de células)|| ||||

|**TOTAL**

Dimensionamento detalhado

Dimensionamento do diretor

Tabela 2. Dimensionamento do diretor

Tarefas da VM	Instâncias	Tipo AMI	Disco da VM (MB)	Disco persistente (MB)
Bosh	1	m4.xlarge	25000	102400
TOTAL	1		25000	102400

Dimensionamento do Cloud Foundry

Tabela 3. Dimensionamento do Cloud Foundry

Tarefas da VM	Instâncias	Tipo AMI	Disco da VM (MB)	Disco persistente (MB)
nfs_WAL_server	1	t2.micro	10000	307200
Nats	2	t2.micro	3000	
adaptador	2	t2.micro	10240	
Banco de dados	1	t2.micro	10000	10000
diego-api	2	t2.micro	10000	
uaa	2	t2.micro	3000	
singleton-blobstore	1	t2.micro	10000	102400
api	2	t2.micro	10000	
cc-trabalhador	2	t2.micro	3000	
roteador	2	t2.micro	3000	
tcp-router	2	t2.micro	3000	
programador	2	t2.micro	3000	
doppler	4	t2.micro	3000	
diego-cell	1	t2.2xlarge	300000	
log-api	2	t2.micro	3000	
credhub	2	t2.micro	3000	
rotate-cc-database-key	1	t2.micro	3000	
TOTAL	34	NA	87000	419600

Dimensionamento da UI da PCP

Tabela 4. Dimensionamento da IU do CFP

Tarefas da VM	Instâncias	Tipo AMI	Disco da VM (MB)	Disco persistente (MB)
Docker	1	t2.medium	30000	65536
TOTAL	1		30000	65536

Dimensionamento dos trabalhadores de compilação

Tabela 5. Dimensionamento dos trabalhadores de compilação

Tarefas da VM	Instâncias	Tipo AMI	Disco da VM (MB)	Disco persistente (MB)
Compilação do trabalhador	5	t2.micro	10000	
TOTAL	5		50000	

Fornecendo certificados para IBM Cloud Private Cloud Foundry

Antes de instalar o IBM® Cloud Private Cloud Foundry, é possível fornecer seus próprios certificados.

É possível usar sua própria autoridade de certificação para criar os certificados, usar uma autoridade de certificação externa para gerar os certificados ou criar manualmente os certificados. Os certificados para domínios são compostos de duas partes, um certificado raiz e uma chave.

Se você não usar seus próprios certificados, os certificados serão criados durante a instalação usando os valores para o domínio de aplicativo, `bluemix_app_domain`, e o domínio de ambiente, `bluemix_env_domain`, que você fornece no arquivo `uiconfig.yml`.

Conclua as etapas a seguir para gerar manualmente os certificados.

Nota: como as chaves dos domínios do aplicativo e do ambiente são necessárias, deve-se concluir este processo duas vezes.

- Para o domínio de ambiente, `bluemix_env_domain`, crie este certificado: `*.management.mycompany.com`

- Para o domínio de aplicativo, `bluemix_app_domain`, crie este certificado: `*.apps.mycompany.com`

1. Gerar um certificado raiz.
   - Execute o comando a seguir:

     ```
     openssl genrsa -des3 -out rootCA.key 2048
     ```

   - Forneça uma senha para o certificado.
   - O certificado é armazenado no arquivo `rootCA.key`.
2. Autoassine o certificado:
   - Execute o comando a seguir:

     ```
     openssl req -x509 -new -nodes -key rootCA.key -days 1024 -out rootCA.pem
     ```

   - Quando solicitado, forneça a senha do certificado e as informações sobre sua organização. Esse certificado expira em 1024 dias. O certificado assinado é armazenado no arquivo `rootCA.pem`.
3. Armazene o arquivo de certificado `rootCA.key` e sua senha em um local seguro.
4. Gere uma chave de domínio para o certificado autoassinado. Execute o comando a seguir:

   ```
   openssl req -new -newkey rsa:2048 -nodes -out star_<your_domain>.csr -keyout
   star_<your_domain>.key -subj "/C=<country_code>/ST=<state>/L=<locality>/O=
   <organization_name>/CN=*.<your_domain>"
   ```

   em que `<your_domain>` é o nome de domínio da instância do Cloud Foundry, por exemplo, `inter.mycompany.com`. O parâmetro `subj` contém os dados a seguir:

   - `<country_code>` é um nome de país de duas letras.
   - `<state>` é o nome do estado ou do município.
   - `<locality>` é o nome da localidade, como uma cidade ou município.
   - `<organization_name>` é o nome da organização ou da empresa. Os arquivos `star_<your_domain>.csr` e `star_<your_domain>.key` são criados.
5. Gerar um certificado de domínio. Conclua as etapas a seguir:
   1. Execute o comando a seguir para criar um arquivo de extensão de certificado de domínio `ext_v3`:

      ```
      [ v3_req ]
      subjectAltName=DNS:*.<your_domain>,DNS:<your_domain>
      ```

   2. Gerar o certificado de domínio. Execute o seguinte comando para criar o arquivo `star_<your_domain>.crt`:

      ```
      openssl x509 -req -in star_<your_domain>.csr -CA rootCA.pem -CAkey rootCA.key -
      CAcreateserial -out star_<your_domain>.crt -days 500 -extensions v3_req -extfile ext_v3
      ```

      em que `ext_v3` é o nome do arquivo de extensão do certificado de domínio criado e `<your_domain>` é o nome de domínio da instância do Cloud Foundry.

      Esse certificado expira em 500 dias.
6. Após a execução do script de instalação, deve-se fornecer os valores de certificado para o Cloud Foundry. Para domínios de aplicativo e de ambiente, coloque o conteúdo dos seguintes arquivos na seção `uiconfig` do arquivo `uiconfig<iaas_type>_template.yml` referenciado pelo comando `launch_deployment.sh -c UICONFIG_TEMLATE_FIL`:
   - `star_<your_domain>.crt`
   - `star_<your_domain>.key`
   - `rootCA.pem`

   Para obter informações adicionais sobre o conteúdo do arquivo `uiconfig<iaas_type>_template.yml`, consulte Instalando o IBM® Cloud Private Cloud Foundry.
   1. Inclua manualmente os valores no arquivo `uiconfig<iaas_type>_template.yml`. Por exemplo:

      ```
      uiconfig:
        bluemix_apps_domain_cert: |+
      ```

```
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  bluemix_apps_domain_cert_rsa_key: |+
    -----BEGIN PRIVATE KEY-----
    -----END PRIVATE KEY-----
  bluemix_env_domain_cert: |+
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  bluemix_env_domain_cert_ca: |+
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  bluemix_env_domain_cert_rsa_key: |+
    -----BEGIN PRIVATE KEY-----
    -----END PRIVATE KEY-----
```

Nota: a ordem que é mostrada deve ser mantida. É possível incluir um ou mais conjuntos de valores para ativar vários domínios.

- **bluemix_apps_domain_cert** e **bluemix_env_domain_cert** são os conteúdos de seus arquivos `star_<env_domain/app_domain>.crt`.
- **bluemix_apps_domain_cert_ca** e **bluemix_env_domain_cert_ca** são o conteúdo do arquivo `rootCA.pem`.
- **bluemix_apps_domain_cert_rsa_key** e **bluemix_env_domain_cert_rsa_key** são os conteúdos dos arquivos `star_<env_domain/app_domain>.key`.

2. Forneça os valores a partir da linha de comandos.

    1. Inclua ou atualize os certificados de gerenciamento (ambiente):

    ```
    ./cm bmxconfig add-certificates -c <your_config_path> -m [--key
    star_<your_domain>.key] [--cert star_<your_domain>.crt] [--rootCA rootCA.pem]
    ```

    2. Inclua ou atualize os certificados de aplicativo:

    ```
    ./cm bmxconfig add-certificates -c <your_config_path> -a [--key
    star_<your_domain>.key] [--cert star_<your_domain>.crt] [--rootCA rootCA.pem]
    ```

Configure a resolução de Domain Name Service do IBM® Cloud Private Cloud Foundry

Após a instalação do IBM Cloud Private Cloud Foundry, deve-se configurar a resolução do nome de domínio.

Deve-se incluir os dois domínios que você usa em sua instalação do IBM Cloud Private Cloud Foundry, `bluemix_env_domain` e `bluemix_apps_domain` no arquivo `uiconfig.yml`, para o servidor DNS de sua empresa. Esse processo envolve associar um domínio curinga a um endereço IP. Se você usar um balanceador de carga na frente do IBM Cloud Private Cloud Foundry, designe o domínio curinga para o endereço IP do balanceador de carga. Também é possível configurar um Servidor DNS privado independente.

Para uma instalação do desenvolvedor ou se nenhum balanceador de carga estiver disponível, é possível apontar os endereços de domínio curinga para um dos endereços IP que estão listados para `router_static_ips` no arquivo `uiconfig.yml`.

Requisitos do DNS independente privado:

1. O DNS privado resolverá os dois domínios do IBM Cloud Private Cloud Foundry: gerenciamento e aplicativo padrão.
2. Os recursos de DNS Privado para o DNS Corporativo e, opcionalmente, os serviços DNS da Internet. Isso permite a resolução de nome interno e externo.

Opções de balanceamento de carga DNS no modo corporativo e do desenvolvedor:

- Um balanceador de carga pode ser usado para direcionar o tráfego dos domínios do IBM Cloud Private Cloud Foundry para os endereços IP listados no `router_static_ips`.
- Se o DNS suportar balanceamento de carga, o tráfego de domínios do IBM Cloud Private Cloud Foundry poderá ser resolvido para qualquer um dos endereços IP listados em `router_static_ips`.

Domínios curinga

1. Na maioria dos sistemas de Servidor DNS, um objeto de domínio é criado,

2. Em seguida, um registro `A` é criado no domínio com `*` e com o `IP address` do balanceador de carga ou com um de `router_static_ips`.

Fazendo upgrade de versões secundárias de stemcell

IBM® Cloud Private Cloud Foundry: Estas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

É possível atualizar versões secundárias de stemcells existentes para a correção mais recente, configurando a extensão da versão de stemcell no IBM Cloud Private Cloud Foundry.

Por exemplo, se o stemcell instalado atual for a versão `250.9`, configurando a extensão, é possível fazer upgrade para a versão `250.25`, já que apenas a versão secundária é mudada.

Configure a versão de stemcells antes de executar o script `launch_deployment.sh` durante a instalação do IBM Cloud Private Cloud Foundry (consulte Instalando o IBM Cloud Private Cloud Foundry) ou a qualquer momento após a instalação. A configuração é feita adicionando a extensão incluída `cfp-ext-stemcell-version` na implementação principal do IBM Cloud Private Cloud Foundry. Para obter informações adicionais, consulte Usando extensões no IBM Cloud Private Cloud Foundry. A extensão `cfp-ext-stemcell-version` modifica as propriedades de implementação para o servidor User Account and Authentication (UAA). Um stemcell diferente pode ser configurado para o diretor Cloud Foundry e para a implementação do Cloud Foundry.

Consulte Stemcells para obter uma lista de stemcells oficiais.

Configurando a extensão após a instalação do IBM Cloud Private Cloud Foundry

Se você optar por configurar a extensão após a instalação do IBM Cloud Private Cloud Foundry, será necessário configurar os estados apropriados para `READY` depois de ativar a extensão. Se estiver atualizando a versão do stemcell diretor, os estados `prepare-director` e `deploy-director` devem ser configurados para `READY`. Se estiver atualizando a versão do stemcell de implementação do Cloud Foundry, o estado `prepare-deployment` e todos os estados depois dele devem ser configurados como `READY`.

Valores de Configuração

override_director: Configure o parâmetro como `true` para substituir o stemcell diretor do stemcell predefinido. Configure o parâmetro como `false` para usar o stemcell predefinido. Se configurado como false, não é preciso especificar nenhum parâmetro adicional que esteja relacionado ao diretor.

director_stemcell_version: A versão do stemcell para o diretor que está sendo especificado. Por exemplo, `250.25`.

director_url: Se estiver fazendo download do stemcell da Internet, este parâmetro é a URL para o stemcell a ser usado para o diretor. O arquivo deve ter uma extensão `.tgz`.

director_filename: Se estiver usando uma cópia local de um stemcell, este parâmetro será o nome do arquivo do stemcell a ser usado para o diretor. Coloque o stemcell no diretório `DATA_DIRECTORY/extensions/embedded/cfp-ext-stemcell-version`. O arquivo deve ter uma extensão `.tgz`.

override_cf_deployment: Configure o parâmetro como `true` para substituir o stemcell de implementação do Cloud Foundry do stemcell predefinido. Configure o parâmetro como `false` para usar o stemcell predefinido. Se configurado como false, não é preciso especificar nenhum parâmetro adicional que esteja relacionado à implementação do Cloud Foundry.

cf_deployment_stemcell_version: A versão do stemcell para a implementação do Cloud Foundry que está sendo especificada. Por exemplo, `250.25`.

cf_deployment_url: Se estiver fazendo download do stemcell da Internet, este parâmetro será a URL para o stemcell a ser usado para a implementação do Cloud Foundry. O arquivo deve ter uma extensão `.tgz`.

cf_deployment_filename: Se estiver usando uma cópia local de um stemcell, este parâmetro será o nome do arquivo do stemcell a ser usado para a implementação do Cloud Foundry. Coloque o stemcell no diretório `DATA_DIRECTORY/extensions/embedded/cfp-ext-stemcell-version`. O arquivo deve ter uma extensão `.tgz`.

Configuração de exemplo

Os valores de configuração devem ser especificados como valores-filhos de uma chave **uiconfig**. O exemplo a seguir está configurado para substituir os stemcells para o diretor e a implementação do Cloud Foundry que usa a versão de stemcell `250.25` transferida por download da Internet.

```
uiconfig:
  override_cf_deployment: "true"
  cf_deployment_stemcell_version: 250.25
  cf_deployment_url: https://s3.amazonaws.com/bosh-core-stemcells/250.25/bosh-stemcell-250.25-
vsphere-esxi-ubuntu-xenial-go_agent.tgz
  override_director: "true"
  director_stemcell_version: 250.25
  director_url: https://s3.amazonaws.com/bosh-core-stemcells/250.25/bosh-stemcell-250.25-vsphere-
esxi-ubuntu-xenial-go_agent.tgz
```

Instalando o IBM Cloud Private Cloud Foundry

Conclua as etapas a seguir para fazer download e instalar o IBM® Cloud Private Cloud Foundry.

A instalação requer aproximadamente de 2 a 3 horas. Este tempo de instalação não inclui o tempo que leva para fazer o download dos arquivos de instalação.

1. Faça download do arquivo de instalação completo do Passport Advantage (14 GB).
2. Pode ser necessário instalar o IBM® Cloud Private Cloud Foundry em um data center que não tem uma conexão de Internet. Nessa situação, copie o arquivo de instalação para um servidor que tenha acesso aos ambientes do Docker Community Edition (CE) e do VMware.
3. Se o Docker CE ainda não estiver instalado, instale o Docker CE. Para obter mais informações, consulte Instalar o Docker.
4. Crie um diretório de instalação.
5. Extraia o arquivo da instalação completa para o diretório de instalação.
6. Mude para o diretório de instalação.
7. Execute o comando a seguir para importar imagens do Docker do diretório de instalação.

   ```
   import_images.sh
   ```

8. Crie um diretório para armazenar a configuração de instalação. Deve-se alocar no mínimo 35 GB de espaço para esse diretório, mas podem ser necessários até 100 GB de espaço. Execute o comando a seguir para criar o diretório de configuração de instalação:

   ```
   mkdir -p /home/user/data
   ```

9. Execute o script de instalação. No exemplo a seguir, `/home/user/data` é o diretório de configuração da instalação.

   ```
   ./launch.sh -n IBMCloudPrivate -c /home/user/data -e LICENSE=accept
   ```

10. É possível encontrar um arquivo de modelo para a infraestrutura do IaaS no seguinte local: `/<installation_configuration_directory>/extensions/embedded/cfp-bosh-templates/uiconfig_<environment_type>_template.yml`. A variável `<environment_type>` depende da infraestrutura do IaaS. Copie esse arquivo para seu diretório de instalação e modifique-o. Substitua os valores padrão e de amostra pelos valores reais para seu ambiente específico.
    - Para parâmetros que são comuns ao vSphere, ao OpenStack e ao AWS, consulte Parâmetros comuns.
    - Para parâmetros que são específicos para o vSphere, consulte Parâmetros do vSphere.
    - Para parâmetros que são específicos para o OpenStack, consulte Parâmetros do OpenStack.
    - Para parâmetros que são específicos para o AWS, consulte Parâmetros do AWS.
11. Configure o DNS. Para obter mais informações sobre o DNS, consulte Configurar a resolução do Serviço de Nome de Domínio IBM® Cloud Private Cloud Foundry.
12. (Opcional) Prepare-se para usar um banco de dados externo. Consulte Configurando o serviço do Director para uso de um banco de dados diferente.
13. (Opcional) Prepare-se para usar um banco de dados externo para o IBM Cloud Private Cloud Foundry. Consulte Configurando bancos de dados remotos para o IBM Cloud Private Cloud Foundry.
14. Use um dos seguintes métodos para iniciar sua implementação:

    Interface da linha de comandos (CLI)

    Execute o comando a seguir para iniciar a implementação e exibir os logs:

    ```
    ./launch_deployment.sh -c your-uiconfig.yml
    ```

Se você usar a CLI para iniciar uma implementação, ainda será possível usar a interface com o usuário para monitorar a implementação.

Ferramenta de implementação do Cloud Foundry

Para obter informações adicionais sobre como usar o Ferramenta de implementação do Cloud Foundry para instalar o IBM Cloud Private Cloud Foundry, consulte Instalando o IBM Cloud Private Cloud Foundry com o Ferramenta de implementação do Cloud Foundry.

Especificando Parâmetros Comuns

É possível localizar um arquivo de modelo para o tipo de infraestrutura do IaaS no seguinte local: `/<installation_configuration_directory>/extensions/embedded/cfp-bosh-templates/uiconfig_<environment_type>_template.yml`. Copie esse arquivo para seu diretório de instalação e modifique-o. Substitua os valores padrão e de amostra pelos valores reais de sua infraestrutura do vSphere, OpenStack ou AWS.

Esses parâmetros são comuns para o AWS, o vSphere e o OpenStack. Por exemplo:

```
uiconfig:
    #Infrastructure agnostic parameters.
  developer_mode: "false"
  main_user_name: "admin"
  main_user_password: "mypassword"
  diego_cell_instances: 1
  bluemix_env_domain: "local.bluemix.net"
  bluemix_env_domain_cert: |+
      -----BEGIN CERTIFICATE-----
      -----END CERTIFICATE-----
  bluemix_env_domain_cert_ca: |+
      -----BEGIN CERTIFICATE-----
      -----END CERTIFICATE-----
  bluemix_env_domain_cert_rsa_key: |+
      -----BEGIN PRIVATE KEY-----
      -----END PRIVATE KEY-----
  bluemix_apps_domain: "local.mybluemix.net"
  bluemix_apps_domain_cert: |+
      -----BEGIN CERTIFICATE-----
      -----END CERTIFICATE-----
  bluemix_apps_domain_cert_rsa_key: |+
      -----BEGIN PRIVATE KEY-----
      -----END PRIVATE KEY-----
  ntp_servers: "ntp1,myntp2.com"
  subnet: "100.155.194.129/27"
  address_range: "100.15.194.130-100.15.194.155"
  external_dns:  "100.12.13.14,100.12.13.15,100.12.13.16"
  gateway: "100.15.194.1"
  director_ip: "100.15.194.2"
  console_ip: "100.15.194.5"
  router_static_ips: "100.15.194.3, 100.15.194.4"
  #cloud configuration customization.
  #Uncomment the attribute 'cloud_config_custom' and add your customization in place of the example,
if needed.
  #cloud_config_custom:  |
    #  ---
    #  my_custom_yaml: my_custom_yaml

  #Cloud Foundry customization
      #Uncomment the attribute `cf_custom` and add your customization in place of the example, if
needed.
  #cf_custom:  |
    #  ---
    #  my_custom_yaml: my_custom_yaml

  #Bosh director customization
      #Uncomment the attribute `director_custom` and add your customization in place of the example,
if needed.
  #director_custom: |
    #  ---
    #  my_custom_yaml: my_custom_yaml

  #Backups
```

```
      #Note: Most of the parameters are optional.

  bbr_backup:
      #Set the `customer_nfs_host`, `customer_nfs_path` and `customer_short_name` if you want the
director and
      #deployment backups pushed to your own NFS server.
  #The NFS mount is built as <customer_nfs_host>:/<customer_nfs_path>/<customer_short_name>
      customer_nfs_host: "NFS IP or host name"
      customer_nfs_path: "/bluemix_nfs"
      customer_short_name: "bluemix1"

  #    director_backup: # Optional. Add this parameter if you want to change the default time for
the director backup.
  #                       The default value is 2 AM in the nfs_WAL_server time zone.
  #      scheduled: "* 2 * * *"
  #      enable: "false" # The default value is true, which means that backups are executed.
  #      nb_backups: "10" #Number of backup to keep on the nfs_WAL_server. O valor padrão é 10.
  # max_log_size: "1048576" #Tamanho máximo do log antes de substituir (em bytes). O valor padrão é
1.048.576 bytes (1 MB).
  #      nb_logs: "10" #Maximum number of logs to keep on the nfs_WAL_server. O valor padrão é 10.
  #    deployments_backup:
      #      deployments: #Optional. Add this parameter if you want to change the default time for
the Bluemix deployment backup.
  #                     The default value is 2:30 AM in the nfs_WAL_server time zone.
  #      - name: Bluemix
      #        enabled: "false" # The default value is true, which means that the backup runs.
  #        nb_backups: "10" #Number of backups to keep on the nfs_WAL_server. O valor padrão é 10.
  # max_log_size: "1048576" #Tamanho máximo do log antes de substituir (em bytes). The default value
is 1048576 (1 MB)
      #        nb_logs: "10" #Maximum number of logs to keep on the nfs_WAL_server. O valor padrão é
10.
  #        schedule: "* 6 * * *"

  db_nfs_copy:
      #Set the customer_nfs_host, customer_nfs_path and customer_short_name if you want the UAADB
and
      #CCDB backups pushed to your own NFS server.
  #The NFS mount is built as <customer_nfs_host>:/<customer_nfs_path>/<customer_short_name>
      customer_nfs_host: "NFS server IP address or host name"
      customer_nfs_path: "/bluemix_nfs"
      customer_short_name: "bluemix1"
```

- **developer_mode**: especifica se você deseja implementar componentes de instância única ou componentes em cluster corporativos. Os valores válidos são "true" ou "false".
- **main_user_name**: o nome do usuário usado para efetuar login na CLI e API do Cloud Foundry. Este usuário se torna o administrador.
- **main_user_password**: a senha para o usuário administrador.
- **diego_cell_instances**: especifique o número de células do Diego para implementar. Cada célula do Diego usa 4 vCPUs, 32 GB de RAM e 300 GB de espaço em disco. Para uma implementação corporativa de alta disponibilidade, deve-se usar duas células Diego.
- **bluemix_apps_domain**: o domínio compartilhado padrão no qual os aplicativos são implementados. Deve-se especificar um valor diferente do valor **bluemix_env_domain**.
- **bluemix_apps_domain_cert**: opcional. Certificado de domínio de aplicativo curinga (por exemplo, `*.domain.com`).
- **bluemix_apps_domain_cert_ca**: Opcional. Curinga de domínio raiz certificado.
- **bluemix_apps_domain_cert_rsa_key**: Opcional. Chave de domínio do aplicativo curinga.
- **bluemix_env_domain**: o domínio de API e gerenciamento que é usado em implementações do Cloud Foundry. Deve-se especificar um valor diferente do valor **bluemix_apps_domain**.
- **bluemix_env_domain_cert**: certificado de domínio do sistema curinga. É possível fornecer este valor ou usar dados gerados automaticamente. Para obter mais informações, consulte Fornecendo certificados para o IBM® Cloud Private Cloud Foundry.
- **bluemix_env_domain_cert_ca**: certificado raiz de domínio do sistema curinga. É possível fornecer este valor ou usar dados gerados automaticamente. Para obter mais informações, consulte Fornecendo certificados para o IBM® Cloud Private Cloud Foundry.
- **bluemix_env_domain_cert_rsa_key**: chave de domínio do sistema curinga. É possível fornecer este valor ou usar dados gerados automaticamente. Para obter mais informações, consulte Fornecendo certificados para o IBM® Cloud Private Cloud Foundry.
- **ntp_servers**: os servidores NTP que são usados para sincronizar o tempo da máquina virtual (VM) durante a inicialização das VMs. Forneça os nomes do servidor como uma lista separada por vírgula que não contenha espaços.

- **subnet**: a sub-rede que hospeda Cloud Foundry, como `192.168.52.0/24`

Nota: uma instalação do desenvolvedor requer 25 endereços IP e uma instalação corporativa requer pelo menos 40 endereços IP.

- **gateway**: o endereço IP que o Cloud Foundry usa para rotear o tráfego da sub-rede que você fornece para outras sub-redes ou redes.

- **address_range**: opcional. Especifique os intervalos de endereços IP que estão disponíveis para uso. O gateway é automaticamente considerado indisponível. Se você não especificar um valor, o intervalo total de **sub-rede** ficará disponível.

  Por exemplo, se o gateway for `192.168.52.1` e você desejar usar os endereços IP .50-.75 para instalar uma instalação do desenvolvedor, especifique `192.168.52.50-192.168.52.75`. É possível especificar vários intervalos na mesma sub-rede separando cada intervalo com uma comma, semelhante ao exemplo a seguir: `192.168.52.50-192.168.52.65,192.168.52.70-192.168.52.80`.

- **external_dns**: a lista de servidores DNS que resolve URLs curingas de empresa, Internet e Cloud Foundry. Forneça os nomes do servidor como uma lista separada por vírgula que não contenha espaços.
- **director_ip**: o endereço IP no intervalo disponível para designar ao diretor.
- **router_static_ips**: Os endereços IP no intervalo disponível para designação a Go Routers. O número de IPs estáticos na lista deve corresponder ao número de Go routers configurados. Se `developer_mode=true`, especifique apenas um IP. Se `developer_mode=false`, especifique dois IPs. Esses endereços IP são o ponto de ingresso.
- **console_ip**: o endereço IP no intervalo disponível para designar à interface com o usuário do console. Esse endereço IP é o ponto de ingresso para o qual o domínio curinga aponta.
- **cloud_config_custom**: detalhes da customização da configuração de nuvem YAML.
- **director_custom**: configurações de YAML que são aplicadas ao BOSH Director. Essas informações de configuração são bem mescladas sobre o modelo de configuração padrão. Crie um arquivo YAML com a configuração customizada.
- **cf_custom**: configurações de YAML que são aplicadas à implementação do Cloud Foundry. Essas informações de configuração são bem mescladas sobre o modelo de configuração padrão. Crie um arquivo YAML com a configuração customizada `(deprecated)`.
- **bbr_backup**: define as propriedades (planejamento, número de backups, servidor NFS externo) para os backups de diretor e implementação. Consulte o arquivo `uiconfig-<vmware|openstack>-template.yml` para obter mais detalhes. Por padrão:
  - O backup de diretor é tomado todos os dias às 2h e 10 backups são mantidos no `nfs_WAL_server`.
  - O backup de implementação do Bluemix (somente manifest) é tomado todos os dias às 02h30 e 10 backups são mantidos no `nfs_WAL_server`.
  - Nenhuma transferência é feita para um servidor NFS externo.
- **bbr_backup.customer_nfs_host**: o endereço IP ou o nome do host do servidor NFS externo.
- **bbr_backup.customer_nfs_path**: o caminho do diretório para o servidor NFS.
- **bbr_backup.customer_short_name**: O ambiente `short_name`. Especifique um nome arbitrário para reagrupar todos os backups para um ambiente.
- **bbr_backup.director_backup.schedule**: o planejamento para o backup do Director em um formato crontab. O valor padrão é `2` (todos os dias às 2h).
- **bbr_backup.director_backup.enabled**: Ativa o diretório de backup. O valor-padrão é true. Você pode desejar desativar esse backup se o banco de dados é externalizado e submetido a backup separadamente.
- **bbr_backup.director_backup.nb_backups**: se nenhum servidor NFS externo for fornecido, esse parâmetro especificará o número de backups a manter no `nfs_WAL_server`. O valor padrão é 10.
- **bbr_backup.director_backup.max_log_size**: o tamanho máximo do log antes de ser substituído. O valor padrão é de 1 Mb.
- **bbr_backup.director_backup.nb_logs**: Número de logs para manter.
- **bbr_backup.deployments_backup*.deployments**: lista das implementações para backup. Por padrão, somente a implementação do Bluemix é submetida a backup usando o script da comunidade do Cloud Foundry. Por padrão, esse backup é planejado às 2h30. Atualmente, apenas o manifest é submetido a backup.
- **bbr_backup.deployments_backup.deployments.name**: o nome da implementação para backup.
- **bbr_backup.deployments_backup.deployments.enabled**: se configurado para true, este parâmetro permite o backup da implementação. Esse parâmetro é útil se você deseja desativar o backup padrão (Bluemix).
- **bbr_backup.deployments_backup.deployments.schedule**: usa a sintaxe do crontab `* * * * *` para especificar quando mover os backups para um local externo. Os campos do Crontab são `MIN(0-59) HOURS(0-23) DAY(1-31) MONTH(1-12) WEEKDAY(0-6)`.
- **bbr_backup.deployments_backup.nb_backups**: se nenhum servidor NFS externo é fornecido, esse parâmetro especifica o número de backups a manter no `nfs_WAL_server`. O valor padrão é 10.
- **bbr_backup.deployments_backup.max_log_size**: especifica o tamanho máximo do log antes que ele seja substituído. O valor padrão é de 1 Mb.
- **bbr_backup.deployments_backup.nb_logs**: especifica o número de logs a manter.
- **db_nfs_copy**: define as propriedades (planejamento, número de backups, servidor NFS externo) para os bancos de dados do UAA e do CC. Por padrão, os backups não são transferidos para um servidor NFS externo.

- **db_nfs_copy.customer_nfs_host**: o endereço IP ou nome do host do servidor NFS externo.
- **db_nfs_copy.customer_nfs_path**: o caminho do diretório para o servidor NFS.
- **db_nfs_copy.customer_short_name**: O ambiente `short_name`. Especifique um nome arbitrário para reagrupar todos os backups de um ambiente.

Especificando Parâmetros do vSphere

Durante a instalação, após a execução de `launch.sh` e antes da execução de `launch_deployment.sh`, localize o arquivo de modelo para vSphere: `/<installation_configuration_directory>/extensions/embedded/cfp-bosh-templates/uiconfig_vmware_template.yml`. Copie esse arquivo para seu diretório de instalação e modifique-o. Substitua os valores padrão e de amostra pelos valores reais para seu ambiente específico.

Os parâmetros de exemplo a seguir se aplicam apenas ao vSphere. Também é possível usar parâmetros comuns. Para obter mais informações, consulte Especificando parâmetros comuns.

```
uiconfig:
  #vSphere specific parameters
  vmware_address: "100.204.4.40"
  vmware_username: "vmware_user"
  vmware_password: "vmware_password"
  vmware_disk_type: "thin or preallocated"
  datacenter_name: "MyDataCenter"
  cluster_name: "MyPersistentDataStorePattern"
  resource_pool: "MyResourcePool"
  template_folder: "MyTemplateFolder"
  vm_folder: "MyVMFolder"
  disk_path: "/Disk"
  datastore_pattern: "MyDataStorePattern*"
  persistent_datastore_pattern: "MyPersistentDataStorePattern"
  portgroup: "myPortGroup"
```

- **vmware_address**: o nome do host ou endereço IP para a instância do vCenter.
- **vmware_username**: o nome do usuário do VMware que está autorizado a acessar o vCenter para Cloud Foundry.
- **vmware_password**: a senha para o usuário do VMware.

- **vmware_disk_type**: o tipo de disco a ser provisionado para máquinas virtuais (VMs). Os valores válidos são `thin` ou `preallocated` e o valor padrão é `thin`. Se o valor de parâmetro é `thin`, VMware não aloca o espaço em disco até que ele seja necessário e seus discos possam ser supercomprometidos. Quando você usa discos `thin` provisioned, o VMware grava novos arquivos mais lentamente e você pode ficar sem espaço devido ao armazenamento supercomprometido.

  Se você especificar `preallocated`, o VMware alocará todo o espaço em disco quando você criar uma VM. O uso desse valor requer mais espaço em disco no momento da implementação, mas assegura que o espaço esteja sempre disponível.

- **datacenter_name**: o data center do vSphere que hospeda os recursos de VMware do Cloud Foundry.
- **cluster_name**: o cluster do vSphere que hospeda os hosts ESXi do Cloud Foundry.
- **resource_pool**: Opcional. O conjunto de recursos que hospeda as máquinas virtuais. Se um não estiver disponível, configure o valor como `Resources`.
- **template_folder**: a pasta vSphere (máquinas virtuais e modelos) que contém todas as máquinas virtuais `stemcell`.
- **vm_folder**: a pasta vSphere (máquinas virtuais e visualização de modelos) que contém todas as máquinas virtuais.
- **disk_path**: o diretório no armazenamento de dados que hospeda discos persistentes da máquina virtual (VMDK). Se ele não existir, o diretório será criado.
- **datastore_pattern**: o padrão que determina quais armazenamentos de dados hospedam as máquinas virtuais do Cloud Foundry.
- **persistent_datastore_pattern**: o padrão que determina quais armazenamentos de dados hospedam discos persistentes da máquina virtual. É possível usar o mesmo valor que o **datastore_pattern**.
- **portgroup**: o grupo da porta vSphere com o qual todas as máquinas virtuais são provisionadas.

Especificando Parâmetros do OpenStack

Durante a instalação, após a execução de `launch.sh` e antes da execução de `launch_deployment.sh`, localize o arquivo de modelo para OpenStack: `/<installation_configuration_directory>/extensions/embedded/cfp-bosh-templates/uiconfig_openstack_template.yml`. Copie esse arquivo para seu diretório de instalação e modifique-o. Substitua os valores padrão e de amostra pelos valores reais para seu ambiente específico.

Os parâmetros de exemplo a seguir se aplicam apenas ao OpenStack. Também é possível usar parâmetros comuns. Para obter mais informações, consulte Especificando parâmetros comuns.

Use a interface com o usuário do OpenStack para fazer download de descrições de parâmetros. Para as versões mais antigas do OpenStack, como Liberty ou Mitaka, selecione `Project > Compute > Access & Security > API Access > Download OpenStack RC File v3`. Para versões mais recentes do OpenStack, como Pike, selecione `Projeto > Acesso à API > Download de arquivo do OpenStack RC > Identity API v3` ou `Projeto > Acesso à API > Download de arquivo do OpenStack RC > Arquivo Clouds.yaml`. Isso faz download de descrições de parâmetros que são mostrados no exemplo a seguir:

```
uiconfig:
  openstack_key_pair_name:  "my_icp_cf_key_name"
  openstack_key_pair_private: |+
      -----BEGIN RSA PRIVATE KEY-----
      -----END RSA PRIVATE KEY-----
  openstack_availability_zone: "nova"
  openstack_security_groups:  "CF-Sec1,CF-Sec2"
  openstack_cacert: |+
      -----BEGIN CERTIFICATE-----
      -----END CERTIFICATE-----

  openstack:
        auth:
          auth_url: "https://hostname:5000/v3"
          project_name: "myopenstack_project_name"
          project_id: "e16asdfasdf7ab80ac9b4234a74d"
          user_domain_name: "myopenstack_domain_name"
          username: "myopenstack_user"
          password: "myopenstack_password"
      identity_api_version: 3
      interface: "public"
      region_name: "RegionOne"

      #Uncomment this section and overwrite with your flavor if needed.
#   flavors:
      #     icpcf_cfp_ui: <your_flavor>
      #     icpcf_compilation: <your_flavor>
      #     icpcf_director: <your_flavor>
      #     icpcf_minimal: <your_flavor>
      #     icpcf_small: <your_flavor>
      #     icpcf_small-highmem: <your_flavor>
```

- **openstack_key_pair_name**: o nome do par de chaves que é usado para acessar as máquinas virtuais do OpenStack. Este é o nome do par de chaves importado para o OpenStack. Na interface com o usuário do OpenStack, selecione `Projeto > Calcular > Acessar & Segurança > Pares de chaves` para ver os nomes de pares de chaves disponíveis.

- **openstack_key_pair_private**: a chave privada do par de chaves que é usada para acessar as máquinas virtuais do OpenStack. **AVISO**: se você estiver usando uma versão do OpenStack mais antiga, como Liberty ou Mitaka, não crie o par de chaves com o painel do OpenStack Horizon. Em vez disso, certifique-se de gerar o par de chaves SSH manualmente. Por exemplo, use o comando ssh-keygen:

  ```
  ssh-keygen -t rsa -b 4096 -C "bosh" -f bosh.key
  ```

  Em seguida, importe esse par de chaves para o OpenStack usando a interface com o usuário do OpenStack, selecione `Projeto > Calcular > Acessar & Segurança > Pares de chaves > Importar par de chaves`. Isso é devido a um erro do OpenStack.

- **openstack_availability_zone**: a zona de disponibilidade.
- **openstack_security_groups**: os grupos de segurança do OpenStack que são conectados a máquinas virtuais. Forneça os nomes de grupo de segurança como uma lista separada por vírgula que não contenha espaços. Por exemplo, `padrão, bosh`.

- **openstack_cacert**: os certificados a serem usados para conectar-se ao OpenStack. O certificado é armazenado em `/data/openstack-cacert.pem`. Para adquirir o certificado a partir da API OpenStack, execute o seguinte:

  ```
  openssl s_client -connect <OpenStack IP>:443 -showcerts |  sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p'
  ```

  Se múltiplas seções de certificado forem retornadas, use somente o último certificado.

- **openstack**: O conteúdo desse atributo está armazenado em `/data/home/.config/openstack/clouds.yaml`, que permite que `openstack --os-cloud cf <sub-command>` seja executado a partir do contêiner de concepção.
- **openstack.auth.auth_url**: A URL de autenticação de OpenStack. Por exemplo, `https://<hostname>:5000/v3`. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_AUTH_URL`.
- **openstack.auth.project_name**: o nome do projeto a ser usado para a instalação do IBM® Cloud Private Cloud Foundry. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_PROJECT_NAME`.
- **openstack.auth.project_id**: o ID do projeto a ser usado para a instalação do IBM® Cloud Private Cloud Foundry. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_PROJECT_ID`.
- **openstack.auth.user_domain_name**: o nome do domínio a ser usado para a instalação do IBM® Cloud Private Cloud Foundry. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_USER_DOMAIN_NAME`.
- **openstack.auth.username**: o usuário do OpenStack a ser usado para implementações do IBM® Cloud Private Cloud Foundry. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_USERNAME`.
- **openstack.auth.password**: a senha do OpenStack a ser usada para implementações do IBM® Cloud Private Cloud Foundry. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_PASSWORD`.
- **openstack.identity_api_version**: deve ser configurado como 3. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_IDENTITY_API_VERSION`.
- **openstack.interface**: os valores válidos são `public` ou `private`. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_INTERFACE`.
- **openstack.region_name**: a região a ser usada para implementações do IBM® Cloud Private Cloud Foundry. Recupere esse valor da variável de ambiente do OpenStack RC, `OS_REGION_NAME`.
- **flavors**: o tipo a ser usado para cada tarefa do IBM® Cloud Private Cloud Foundry. Se o usuário fornecido tiver a autoridade para criar tipos, eles serão criados automaticamente. Se não, será possível configurar seus próprios tipos para cada tarefa usando este parâmetro. Para obter mais informações, consulte Requisitos de tamanho do OpenStack para instalação de desenvolvedor do IBM® Cloud Private Cloud Foundry.

Especificando parâmetros do AWS

Se você estiver usando a infraestrutura do AWS, configure os parâmetros necessários durante a instalação do IBM® Cloud Private Cloud Foundry.

Antes de iniciar: siga as etapas iniciais no Amazon Web Services para configurar seu ambiente antes de iniciar a instalação.

É possível localizar um arquivo de modelo para o AWS no seguinte local: `/<installation_configuration_directory>/extensions/embedded/cfp-bosh-templates/uiconfig_aws_template.yml`. Copie esse arquivo para seu diretório de instalação e modifique-o. Substitua os valores padrão e de amostra pelos valores reais para seu ambiente específico.

Os parâmetros de exemplo a seguir se aplicam apenas ao AWS. Também é possível usar parâmetros comuns. Para obter mais informações, consulte Especificando parâmetros comuns.

```
uiconfig:
  aws:
    access_key_id: "ABCDE..." #Provided by your AWS account manager
    secret_access_key: "12345..." #Provided by your AWS account manager
    default_key_name: "BOSH" #User defined in AWS
    default_security_groups: [bosh] #User defined in AWS
    region: "us-east-2" #Chosen from AWS regions
    availability_zone: "us-east-2a" #Chosen based on your AWS region
    subnet_id: "subnet-01124214214..." #Defined in AWS for the availability_zone picked above
    key_pair_private:
      -----BEGIN RSA PRIVATE KEY-----
      -----END RSA PRIVATE KEY-----
```

- **access_key_id**: chave de acesso da conta do Amazon.
- **secret_access_key**: chave secreta usada para a conta do AWS.
- **default_key_name**: nome da chave SSH criado para uso com o IBM CF AMI's.
- **default_security_groups**: o grupo de segurança usado para cada AMI para controlar o tráfego de rede.
- **region**: o AWS Region que você deseja usar.
- **availability_zone**: a zona de disponibilidade singleton.
- **subnet_id**: o ID de sub-rede do VDC que será usado para o IBM CF AMI's para se conectar com os `Recursos`.
- **key_pair_private**: chave privada que foi criada com o `default_key_name` acima. Permite acesso SSH ao AMI's.

Instalando o IBM Cloud Private Cloud Foundry com

Ferramenta de implementação do Cloud Foundry

O Ferramenta de implementação do Cloud Foundry é instalado automaticamente no contêiner de concepção quando você executa o script `launch.sh`. É possível usar o Ferramenta de implementação do Cloud Foundry para gerenciar extensões, fazer upload de suas configurações e iniciar e monitorar implementações.

Por padrão, o Ferramenta de implementação do Cloud Foundry é acessível a partir de todos os endereços IP e usa a porta 30100 para se comunicar. É possível usar os parâmetros a seguir para restringir o acesso do cliente:

- Use o parâmetro `-ih` para configurar endereços IP de origem desejados. O padrão, 0.0.0.0, é aberto para todas as origens.
- Use o parâmetro `-pui` para ativar as portas desejadas para o Ferramenta de implementação do Cloud Foundry. O padrão é 30100.
- Use o parâmetro `-psslui` para ativar a porta desejada para acesso HTTPS ao Ferramenta de implementação do Cloud Foundry. O padrão é 30102.
- Use o parâmetro `-dc` para especificar o caminho para seu próprio certificado de domínio.
- Use o parâmetro `-dk` para especificar o caminho para sua própria chave RSA.
- Use o parâmetro `-hn` para especificar o domínio do certificado. O padrão é `localhost`.
- Use o parâmetro `-rc` para renovar os certificados ou se desejar migrar de um certificado gerado automaticamente para seu próprio certificado.

  Nota: Se os parâmetros `-dc` e `-dk` não forem fornecidos, será criado um certificado autoassinado válido para o domínio fornecido. Consulte Fornecendo certificados para o IBM Cloud Private Cloud Foundry para obter informações adicionais.

Por exemplo:

```
./launch.sh -n IBMCloudPrivate -b ./BOM.yml -c /home/user/data -e LICENSE=accept -ih 10.10.1.2 -pui
9090
```

Definindo as Configurações do Ferramenta de implementação do Cloud Foundry

Conclua as etapas a seguir para configurar o Ferramenta de implementação do Cloud Foundry:

1. A partir do diretório de instalação, execute o seguinte comando:

   ```
   ./cm api
   ```

   Copie o `Token`. É preciso o token para configurar o Ferramenta de implementação do Cloud Foundry.

2. Em um navegador da web, navegue para `http[s]://<IP>:<Port>`, em que IP é o endereço IP do servidor que está executando o contêiner de concepção (geralmente o servidor a partir do qual você insere o comando `launch.sh`). As portas padrão são 30100 (HTTP) e 30102 (HTTPS) ou as portas especificadas nos parâmetros `-pui` e `-psslui` do comando `launch.sh`.

   Nota: Se você gerou automaticamente ou autoassinou certificados, seu navegador deverá confiar na URL do Ferramenta de implementação do Cloud Foundry.

3. Clique em `Configurações` e preencha os campos a seguir:

   - **Terminal da API do gerenciador de configuração** `http[s]://<IP>:<port>` em que IP é o endereço IP do servidor que está executando o contêiner de concepção e a porta é a porta padrão 30101 (HTTP) e 30103 (HTTPS) ou a porta especificada nos parâmetros `-p` e `-pssl` do comando `launch.sh`.

     Nota: Se você usar o certificado gerado automaticamente ou autoassinado, seu navegador deve confiar na URL de terminal da API do gerenciador de configuração.

   - **Token** este é o token que foi copiado na etapa 1.

4. Clique em `Enviar`.

Gerenciando extensões

A IBM fornece extensões prontas para serem usadas. Também é possível incluir suas próprias extensões customizadas. Para obter mais informações sobre como usar extensões, consulte Usando extensões no IBM® Cloud Private Cloud Foundry.

É possível visualizar, registrar e cancelar o registro de extensões.

- Para visualizar todas as extensões disponíveis e para registrar ou cancelar o registro de extensões customizadas, selecione `Registrar Novas Extensões` no menu. A partir da mesma página, também é possível incluir e remover uma extensão no arquivo de estados.
- Para incluir ou remover extensões do arquivo de estados de implementação, selecione `Estados` no menu.
- Monitore implementações e visualize os logs de implementação a partir da tabela `Estados`. Para visualizar logs para um estado de implementação específico, clique no ícone de doc que está associado ao estado no qual você está interessado.

Configurações

É possível fazer upload, criar, editar ou visualizar e implementar suas configurações.

- Para fazer upload de novas configurações para implementações e extensões principais, selecione `Fazer upload da configuração` no menu.
- Para criar uma nova configuração, selecione o tipo de configuração e clique no lápis. Agora é possível editar todos os parâmetros da configuração. Em seguida, clique em `Salvar` ou em `Salvar e sair` para salvar a configuração.
- Depois que a configuração é criada ou transferida por upload, as definições de configuração são validadas. Caso não sejam encontrados erros, clique em `Iniciar implementação` para implementar a configuração. A janela `Estados` é aberta e permite visualizar o progresso da implementação.

Estados

Na página `Estados`, é possível acompanhar até a implementação.

- Para visualizar os logs de um estado, clique no ícone `documento` próximo do estado.
- Para mudar o status de um estado ou de diversos estados, selecione os estados para os quais você deseja mudar o status. Clique no lápis e, em seguida, selecione o status que você deseja configurar.
- Para reconfigurar todos os estados para `READY`, clique no menu e selecione `Reconfigurar Status`.
- Para incluir uma extensão, clique em `Incluir Extensão`.
- O ícone de marcação na coluna `Próxima Execução` indica o estado correspondente que é executado na próxima implementação. Para cada mudança no status de um estado, essa coluna é atualizada com base nas dependências de estado. Por exemplo, se `Implementar Diretor` estiver configurado como pronto, um ícone de marcação aparecerá ao lado de `Preparar Diretor` para indicar que ele será executado na próxima implementação.

Logs do Servidor

No menu `Logs do servidor`, é possível ver os logs do `Gerenciador de configuração`.

Etapas de pós-instalação para IBM Cloud Private Cloud Foundry

Essas etapas são opcionais depois de concluir a instalação do IBM® Cloud Private Cloud Foundry. Conclua as etapas a seguir se você deseja criptografar seus dados de configuração de instalação do IBM® Cloud Private Cloud Foundry em repouso.

1. Pare o contêiner de incepção:

   ```
   docker stop inception-NAME
   ```

   em que `NAME` foi especificado com `-n` quando executou `launch.sh`.

2. Decida o que você deseja fazer com os arquivos no **Diretório de configuração de instalação**. Você tem duas opções:

   1. Deixe os arquivos no local.

   2. Mova os arquivos para um local ou dispositivo corporativo obrigatório.

3. Criptografe o diretório e o conteúdo.
   O exemplo a seguir usa `encryptfs` como `root`.

   Instale `encryptfs` na máquina em que o contêiner de concepção está em execução:

   ```
   # Ubuntu
   sudo apt-get install ecryptfs-utils
   encryptfs-setup-private
   ```

Depois de concluir o procedimento de configuração, é possível mover o conteúdo do **Diretório de configuração de instalação** para `~/Private`. Os arquivos agora são criptografados em repouso. Na próxima vez em que você executar `launch.sh -c`, use o novo local do diretório `~/Private`. O procedimento deve ser feito como `root` para certificar-se de que o contêiner possa acessar o diretório `~/Private`. Caso contrário, o conteúdo poderá precisar ser movido temporariamente para um local que não seja criptografado enquanto o contêiner de concepção estiver em execução.

Instalando o Cloud Foundry Enterprise Environment

Siga as instruções para instalar o Cloud Foundry Enterprise Environment nos contêineres do Kubernetes.

- Preparando-se para instalar o Cloud Foundry Enterprise Environment
- Instalando o Cloud Foundry Enterprise Environment

Preparando para Instalar o Cloud Foundry Enterprise Environment

Deve-se preparar seu nuvem antes de instalar o Cloud Foundry Enterprise Environment.

- Dimensionamento do Cloud Foundry Enterprise Environment
- Parâmetros do Cloud Foundry Enterprise Environment

Dimensionamento do Cloud Foundry Enterprise Environment

Para instalar o Cloud Foundry Enterprise Environment, sua instância deve atender aos seguintes requisitos de tamanho.

Nós do trabalhador da instância da célula

Cada nó do trabalhador usado como uma célula do Cloud Foundry Enterprise Environment é distorcido para reservar a quantidade máxima de recursos estritamente para uso da célula do Cloud Foundry. Entretanto, há pods da infraestrutura essencial do IBM® Cloud Private que devem ser executados em cada nó do trabalhador como parte dos serviços de gerenciamento para o Kubernetes.

Para visualizar os nós que são rotulados para uso pelo Cloud Foundry Enterprise Environment, execute o seguinte comando:

```
kubectl get nodes -L bcf.type
```

Para visualizar os pods que estão em execução em um nó e ver os que são obrigatórios para a infraestrutura do IBM Cloud Private, execute o seguinte comando:

```
kubectl describe node <node name>
```

Cada pod no namespace kube-system faz parte da infraestrutura. Cada pod no UAA e no namespace Cloud Foundry faz parte da infraestrutura do Cloud Foundry Enterprise Environment.

Geralmente, os pods da infraestrutura do IBM Cloud Private consomem aproximadamente de 256 MB a 1 GB em cada nó do trabalhador, dependendo do número de serviços de gerenciamento do IBM Cloud Private ativados. Ao dimensionar os nós do trabalhador, use as seguintes diretrizes para uma configuração mínima:

Tabela 1. Configuração mínima

Nó bcf.type	vCPU	Memória	Armazenamento
diego-cell	4	16 GB	160 GB
de acesso	4	16 GB	160 GB

Ao dimensionar os nós do trabalhador, use as seguintes diretrizes para uma configuração de produção padrão:

Tabela 2. Configuração de produção padrão

Nó bcf.type	vCPU	Memória	Memória
diego-cell	4	32 GB	250 GB
de acesso	8	16 GB	160 GB

Proporção de instâncias de plano de controle para instâncias da célula

Os pods da célula Diego são responsáveis pelo planejamento de contêineres de aplicativo do Cloud Foundry no ambiente do Cloud Foundry Enterprise Environment. O tempo de execução Diego do Cloud Foundry supõe que ele tenha controle exclusivo sobre a CPU, a memória e os recursos de disco do nó do trabalhador. Etapas de configuração adicionais são necessárias para assegurar que os nós do trabalhador sejam dedicados para uso exclusivo pelos pods da célula Diego implementados. Para obter informações adicionais, consulte Instalando o Cloud Foundry Enterprise Environment. O implementador deve identificar, rotular e distorcer nós do trabalhador que executam pods da célula Diego no ambiente do Cloud Foundry Enterprise Environment. O número de nós do trabalhador que executam pods da célula Diego deve ser igual ao número de células Diego no Cloud Foundry Enterprise Environment que você está configurando.

Conforme o número de instâncias de célula aumenta, é necessário aumentar o número de instâncias de plano de controle. A tabela a seguir fornece a proporção recomendada:

Tabela 3. Proporção recomendada de instâncias de célula para instâncias de plano de controle

Controle	Célula Diego	Notes
1	1	Apenas para modo de desenvolvimento
2	2-6	Modo de produção
3	6-9	Modo de produção

Nota: a redução não é suportada.

Parâmetros do Cloud Foundry Enterprise Environment

Durante a instalação, especifique os parâmetros a seguir para configurar a implementação.

Parâmetros do Cloud Foundry

- **Ação do instalador**: Tipo de ação do instalador a ser executada. As opções válidas são:
  - **install**: Instalar uma nova implementação.
  - **upgrade**: Fazer upgrade de uma implementação existente. Essa ação não é suportada quando o `nome da classe de armazenamento do Kubernetes` é `local`.
  - **scale**: Aumentar o número de instâncias de célula ou de instâncias de plano de controle em uma implementação existente.
  - **uninstall**: Excluir a implementação existente.

    NOTA: Antes de iniciar a implementação, certifique-se de que a página de configurações do Ferramenta de implementação do Cloud Foundry indique o status `Pronto` para cada etapa que você deseja executar.
- **Modo de desenvolvedor**: se marcado, então um ambiente com tamanho reduzido será implementado. **O número de instâncias do plano de controle** é configurado como 1 e o **Número de instâncias de célula** é configurado como 1.
- **Senha do administrador**: a senha do administrador (admin) do Cloud Foundry.
- **Senha do administrador do UAA**: a senha do administrador UAA (admin) do Cloud Foundry.
- **Número de instâncias de célula**: o número de células deve estar entre 1 e 9. Duas ou mais são necessárias para alta disponibilidade.
- **Número de instâncias do plano de controle**: o número de instâncias do plano de controle para a api, uaa, nats, etc. Duas ou mais são necessárias para alta disponibilidade.
- **Porta do console de gerenciamento**: Porta HTTPS externa para o console de gerenciamento.

Nota: o total entre o `Número de instâncias de célula` e o `Número de instâncias do plano de controle` deve ser menor ou igual ao número de nós do trabalhador do Kubernetes.

- Opções do banco de dados externo:
  - **Usar banco de dados Postgres externo**: Quando desmarcada, o ambiente configura um banco de dados Postgres na implementação. Quando marcada, os seguintes parâmetros são obrigatórios:
    - **Nome do host do banco de dados**: O nome do banco de dados Postgres a ser usado.
    - **Porta do banco de dados**: A porta do banco de dados do Postgres para conexão.
    - **Usuário do banco de dados**: ID do usuário do Postgres. Deve ter privilégios para criar tabelas.
    - **Senha do banco de dados**: Senha para o usuário do Postgres.
    - **Nome do banco de dados**: Nome do banco de dados Postgres padrão a ser usado. Se ficar em branco, 'compose' será usado.

- **Excluir tabelas**: Eliminar quaisquer tabelas de banco de dados Postgres existentes durante a instalação.

Parâmetros do Kubernetes

- **IBM Cloud Private username** : o nome do usuário para IBM® Cloud Private. Padrão: `admin`.
- **IBM Cloud Private password** : a senha para o usuário do IBM Cloud Private .
- **Nome da classe de armazenamento do Kubernetes**: o nome da classe de armazenamento em disco do Kubernetes para uso de dados persistentes. O nome `local` é reservado e só deve ser usado para ambientes de `modo de desenvolvedor` de não produção. O nome `storageclass` já deverá existir, a menos que o valor tenha sido especificado como `local`. Padrão: `local`.
- **Nome do host do Docker**: o nome do host ou o endereço IP para o repositório do Docker do qual puxar imagens do contêiner. Esse parâmetro é opcional.
- **Nome do usuário do Docker**: o nome de usuário para o repositório do Docker do qual puxar imagens do contêiner. Esse parâmetro é opcional.
- **Senha do Docker**: a senha para o repositório do Docker do qual puxar as imagens de contêiner. Esse parâmetro é opcional.
- **Organização do Docker**: a organização para o repositório de Docker do qual puxar as imagens do contêiner. Esse parâmetro é opcional.

Parâmetros de domínio curinga e certificados

- **Domínio de aplicativo**: domínio de aplicativo padrão do Cloud Foundry.

- **Segredo de chave privada e de certificado de domínio de aplicativo**: o nome secreto do Kubernetes (TLS) que contém a chave privada e o certificado de domínio de aplicativo Cloud Foundry. Esse parâmetro é opcional. Para obter informações adicionais sobre como criar um certificado de domínio, consulte Fornecendo certificados para o IBM Cloud Private Cloud Foundry. O objeto secreto deve estar no mesmo namespace no qual o gráfico do Helm do Cloud Foundry Enterprise Environment está implementado.

  Para criar o segredo de Kubernetes, por exemplo, use o seguinte código em que `${APP_DOMAIN}` é o `domínio de aplicativo`:

  ```
  kubectl create secret tls star.${APP_DOMAIN} --key star_${APP_DOMAIN}.key --cert
  star_${APP_DOMAIN}.crt
  ```

- **Segredo de certificados de contêiner de aplicativos confiáveis**: o nome secreto do Kubernetes que contém certificados de Autoridade de Certificação confiáveis para instalação em contêineres de aplicativos do Cloud Foundry. O objeto secreto deve estar no mesmo namespace no qual o gráfico do Helm do Cloud Foundry Enterprise Environment está implementado.
- **Atualize o Kubernetes DNS (kube-dns) com domínio**: se a resolução de DNS não for fornecida externamente para o domínio listado e o kube-dns precisar resolver esse domínio, deixe esse parâmetro marcado.
- **Endereço IP para entrada DNS do Kubernetes (kube-dns)**: Endereço IP do ingresso do IBM Cloud Private usado por aplicativos Cloud Foundry Enterprise Environment. Se **Atualizar DNS Kubernetes (kube-dns) com o domínio** estiver marcada, este endereço IP será usado para atualizar kube-dns. Para que os aplicativos Cloud Foundry se comuniquem uns com os outros, use um DNS voltado para o público, por exemplo, o endereço IP externo do proxy ou o endereço IP do balanceador de carga. Caso contrário, é possível usar o endereço IP interno do proxy.

Instalando o Cloud Foundry Enterprise Environment

A instalação do Cloud Foundry Enterprise Environment é um processo com várias etapas.

- Namespace de Destino
- Instale o IBM Certified Container
- Nós do Trabalhador
- Portas de Entrada
- Criar armazenamento persistente para o Ferramenta de implementação do Cloud Foundry
- Implementar a liberação do Helm
- Implementar o Cloud Foundry Enterprise Environment usando o Ferramenta de implementação do Cloud Foundry
- Implementar o Cloud Foundry Enterprise Environment usando a CLI do config-manager
- IBM Cloud Private Cloud Foundry console de gerenciamento

Espaço de Nomes de Destino

É possível usar o namespace `default` ou um namespace de destino de sua opção. Deve-se destinar o mesmo namespace ao instalar o IBM Certified Container e implementar o gráfico do Helm, já que o script `load_cloudpak.sh` carrega imagens no registro privado e todas as imagens que são carregadas no registro privado incluem o namespace de destino em seus nomes.

A menos que indicado de outra forma, quaisquer recursos que forem criados para o gráfico do Helm que pertencerem a um namespace, como segredos do Kubernetes, deverão ser criados no namespace de destino.

Um número de políticas e privilégios específicos são necessários pelo gráfico do Helm para implementar o Cloud Foundry Enterprise Environment. Se você não estiver usando o namespace `default`, assegure-se de configurar as permissões a seguir.

1. O namespace de destino deve ter uma política de segurança de pod que permita que os pods sejam executados com qualquer usuário, como `ibm-anyuid-psp`. Se você estiver usando volumes hostPath para uma instalação de demonstração, use uma política que permita o hostPath, como `ibm-anyuid-hostpath-psp`. É possível ligar uma política de segurança de pod a um namespace ao criá-lo usando a console de gerenciamento. Os comandos `kubectl` a seguir mostram como é possível ligar uma política a um namespace chamado `cfee` usando a CLI.

   ```
   kubectl create namespace cfee
   kubectl -n cfee create rolebinding ibm-anyuid-clusterrole-rolebinding --clusterrole=ibm-anyuid-clusterrole - group=system:serviceaccounts:cfee
   ```

2. A conta de serviço padrão para o namespace de destino deve ter privilégios de administrador de cluster. Essa permissão é necessária para o Ferramenta de implementação do Cloud Foundry para determinar o endereço IP do nó do proxy e a porta para o serviço de ingresso de gerenciamento do IBM Cloud Private e para o Open Service Broker para implementar os gráficos do Helm. É possível executar o comando a seguir para criar um ClusterRoleBinding que conceda essas permissões a um namespace não padrão. No exemplo a seguir, o namespace de destino é chamado `cfee`.

   ```
   kubectl create clusterrolebinding cfee-serviceaccount-cluster-admin  --clusterrole=cluster-admin --serviceaccount=cfee:default
   ```

Instalar o IBM Certified Container

Conclua as etapas a seguir para fazer download e instalar o gráfico do Cloud Foundry Enterprise Environment IBM Certified Container.

1. Faça download do gráfico do IBM Certified Container a partir do IBM Passport Advantage®
2. Prepare-se para Instalar o software IBM no IBM Cloud Private, mas não execute a etapa, `cloudctl catalog load-archive`. Siga as etapas restantes nesta página:

3. Descompacte o IBM Certified Container usando o comando a seguir:

   ```
   tar xvf <IBM Cloud Private binary download>.tgz
   ```

4. Carregue o IBM Certified Container no IBM Cloud Private:

   ```
   scripts/load_cloudpak.sh -n <namespace> -c <ICP hostname> -u <ICP User> -a ./ibm-cfee-installer-archive-3.2.0-*.tgz
   ```

   Exemplos padrão: -n default -c mycluster.icp -u admin

Nós do Trabalhador

Deve haver um mínimo de quatro nós do trabalhador em seu cluster. Todos os nós do trabalhador devem conter pelo menos quatro núcleos cada. Cada nó do trabalhador pode ser usado por uma instância do plano de controle ou por uma instância de célula. A localização dos nós do plano de controle é determinada pelo usuário. A localização das instâncias de célula é determinada pelo usuário. O número máximo de instâncias de célula e instâncias de plano de controle é limitado pelo número de nós do trabalhador.

Todos os nós do trabalhador para instâncias de plano de controle devem ser modificados para assegurar a operação adequada. Faça as seguintes mudanças em cada nó do trabalhador que é uma instância do plano de controle:

- Rotule os nós do trabalhador que são usados para executar os pods da instância de controle como `bcf.type=control` executando o seguinte comando:

  ```
  kubectl label node <node-name> bcf.type=control
  ```

Todos os nós do trabalhador para instâncias de célula devem ser modificados para assegurar a operação adequada. Faça as seguintes mudanças em cada nó do trabalhador que é uma instância de célula:

- Rotule os nós do trabalhador que são dedicados a executar pods de célula diego como `bcf.type=diego-cell` usando o seguinte comando:

  ```
  kubectl label node <node-name> bcf.type=diego-cell
  ```

- Marque os nós do trabalhador que são dedicados a executar pods de célula diego como `dedicated=diego-cell` executando o seguinte comando:

  ```
  kubectl taint node <node-name> dedicated=diego-cell:NoSchedule
  ```

- Mude as configurações do grub nos nós do trabalhador que executam instâncias de célula diego para assegurar que não haverá problemas com o limite de troca de cgroup enquanto o Docker estiver em execução. Sem essa modificação, você pode ver as mensagens de erro a seguir:

  ```
  AVISO: seu kernel não suporta o limite de troca do cgroup. AVISO: seu kernel não suporta recursos de limite de troca. Limitação descartada.
  ```

  ou

  ```
  memory.memsw.limit_in_bytes: permissão de permissão negada
  ```

Para cada nó trabalhador que é uma instância de célula em seu ambiente, conclua as seguintes etapas:

1. SSH para o nó do trabalhador. **Nota**: você pode precisar executar um SSH para o nó principal primeiro e, em seguida, para os nós do trabalhador a partir do principal.
2. Verifique `/etc/default/grub` para assegurar que a linha a seguir exista:

   ```
   GRUB_CMDLINE_LINUX = "cgroup_enable = memory swapaccount= 1"
   ```

3. Os nós do trabalhador que são designados como `célula diego` devem ser atualizados com configurações do grub. Se você fizer mudanças em `/etc/default/grub`, atualize o grub com o comando a seguir:

   ```
   sudo update-grub
   ```

4. Se o grub foi atualizado, reinicialize o nó do trabalhador. Siga o procedimento padrão do Kubernetes Manutenção em um nó ou o procedimento que é usado por sua organização. Por exemplo:

   - Marque o nó do trabalhador como não planejável

     ```
     kubectl cordon < worker node >
     ```

   - Drene o nó do trabalhador

     ```
     kubectl dreno < nó do trabalhador >
     ```

   - No nó do trabalhador:

     ```
     sudo reboot -f
     ```

   - Quando o nó do trabalhador estiver em execução, ative-o para planejamento

     ```
     kubectl uncordon < worker node >
     ```

5. Execute estas ações em cada nó trabalhador que é uma instância de célula. Se você incluir um novo nó do trabalhador, execute as mesmas ações.

Portas de Entrada

Assegure-se de que as portas a seguir tenham acesso de entrada para o ambiente do Kubernetes:

- 2222
- 2793
- Se desejar usar o Ferramenta de implementação do Cloud Foundry para fazer a instalação, você terá que abrir quatro portas no intervalo de 30000 a 32767 (os padrões são 31080, 32080, 31442, 32443). Em seguida, conecte o Ferramenta de implementação do Cloud Foundry da rede do usuário ao ambiente IBM Cloud Private.

Por exemplo, no OpenStack, em que o tráfego de entrada é restrito, execute as tarefas a seguir para criar o grupo de segurança necessário para que o controlador de ingresso permita o tráfego de entrada nas portas necessárias. No console de gerenciamento do OpenStack, com o **Domínio** e o **Projeto** adequados selecionados, conclua o procedimento a seguir:

1. Navegue para **Grupos de segurança**:
    - Para as versões mais antigas do OpenStack, como Liberty ou Mitaka:
        - Selecione **Projeto** > **Cálculo** > **Acesso & Segurança** > **Grupos de Segurança**.
    - Para as versões mais recentes do OpenStack, como Pike:
        - Selecione **Projeto** > **Rede** > **Grupos de Segurança** .
2. Clique em **Criar Grupo de Segurança**.
3. Nomeie o grupo de segurança **icp-cfee** e inclua a descrição **Grupo de Segurança CFEE do ICP**.
4. Clique em **Criar Grupo de Segurança**.
5. Selecione o **Grupo de Segurança CFEE do ICP** e clique em **Editar Regras**.
6. Clique em **Incluir regra**
7. Inclua as regras a seguir no **Grupo de Segurança CFEE do ICP**:

Regra	Orientação	Tipo Ether	Protocolo IP	Porta ou Intervalo	Remota	Propósito
Regra TCP customizada	Ingresso	IPv4	TCP	2222	0.0.0.0/0 (CIDR)	UAA CFEE
Regra TCP customizada	Ingresso	IPv4	TCP	2793	0.0.0.0/0 (CIDR)	CFEE diego-access
Regra TCP customizada	Egresso	IPv4	Qualquer um	-	0.0.0.0/0 (CIDR)	
Regra TCP customizada	Egresso	IPv6	Qualquer um	-	::/0 (CIDR)	

Criar armazenamento persistente para Ferramenta de implementação do Cloud Foundry

Volume persistente para a liberação do Helm

- O administrador deve criar um volume persistente. A classe de armazenamento do volume persistente é usada para a solicitação de volume persistente do gráfico do Helm.
- O exemplo em Instalar o gráfico usa `hostPath`, mas é recomendado usar um volume persistente em um sistema de arquivos de rede (NFS), no GlusterFS ou em outra infraestrutura compartilhada. O `hostPath` pode ser usado apenas para demonstração.
- O volume persistente deve ter pelo menos 10 GB disponíveis para a ferramenta de implementação.
- O volume persistente deve ser configurado como `Reter` para a política de solicitação de volume persistente para manter os dados de implementação no caso de o aplicativo ser removido temporariamente.

Volume persistente para Cloud Foundry Enterprise Environment

Você precisa de armazenamento persistente separado para o Cloud Foundry Enterprise Environment. O nome da classe de armazenamento é necessário quando você usa a Ferramenta de implementação do Cloud Foundry no campo **Nome da classe de armazenamento do Kubernetes**. O nome `local` é reservado e deve ser usado apenas para ambientes de não produção. O nome da classe de armazenamento já deve existir, exceto se o valor for especificado como `local`.

1. Em IBM Cloud Private console de gerenciamento, abra o Catálogo.

2. Localize e selecione o gráfico `ibm-cfee-installer` .

3. Crie um volume persistente (PV) que pode ser um Network File System (NFS), ou outro tipo de PV com uma classe de armazenamento específica. A capacidade de armazenamento precisa ter pelo menos 10 GB. O código a seguir é uma definição de volume persistente de amostra que pode ser usada apenas para propósitos de demonstração ou de prova de conceito.

```
kubectl create -f - <<EOF
kind: PersistentVolume
apiVersion: v1
metadata:
  name: ibm-cfee-installer-data
spec:
  capacity:
    storage: 10Gi
  storageClassName: ibm-cfee-installer-storage
  accessModes:
    - "ReadWriteOnce"
  persistentVolumeReclaimPolicy: Retain
  hostPath:
    path: /tmp/icp/cfee/data
    type: DirectoryOrCreate
---
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
```

```
  name: ibm-cfee-installer-storage
provisioner: kubernetes.io/no-provisioner
EOF
```

Implementar a liberação do Helm

1. Em IBM Cloud Private console de gerenciamento, abra o Catálogo.
2. Localize e selecione o gráfico `ibm-cfee-installer` .
3. Assegure-se de criar um volume persistente conforme mostrado no arquivo leia-me do gráfico do Helm.
4. Revise as instruções fornecidas e selecione **Configurar**.
5. Forneça um nome de liberação e selecione um namespace. No exemplo, no gráfico do Helm, o nome da liberação é `cfee-inception` e o namespace é `default`.
6. Revise e aceite a licença ou as licenças.
7. Forneça o nome da classe de armazenamento. No exemplo no gráfico do Helm, a classe de armazenamento é `ibm-cfee-installer-storage-storage`.
8. Selecione **Instalar** para concluir a instalação do Helm.

Implemente o Cloud Foundry Enterprise Environment usando o Ferramenta de implementação do Cloud Foundry

Quando o gráfico estiver instalado, execute as ações a seguir para acessar o Ferramenta de implementação do Cloud Foundry e iniciar a implementação do Cloud Foundry.

1. Na IBM Cloud Private do IBM Cloud Private , abra **Cargas de trabalho** > **Liberações de Helm** .
2. Localize e selecione o gráfico `ibm-cfee-installer` que você instalou.
3. Em **Liberação do Helm** , selecione **Ativar** > **deployment-tool** . Uma nova guia é aberta com a página de configurações da ferramenta de implementação do Cloud Foundry. Os dois valores de configuração que você precisa podem ser obtidos executando comandos `kubectl`. Os comandos a serem executados são listados na seção **Notas** da liberação do Helm.
4. Execute os dois comandos que foram gerados quando o gráfico do Helm foi implementado. Para ver esses comandos, navegue para o gráfico do Helm implementado e role para baixo. Esses comandos são necessários para obter a chave de API e a URL da API para o Instalador do Cloud Foundry Enterprise Environment. Copie os valores para o campo `Terminal da API do gerenciador de configuração` no Ferramenta de implementação do Cloud Foundry.
5. Execute o comando listado em `3. Obtenha o token executando estes comandos:`. Copie o valor para o campo `Token` no Ferramenta de implementação do Cloud Foundry.
6. No Ferramenta de implementação do Cloud Foundry, selecione **Enviar**.
7. Quando a página Configuração for aberta, clique em **Selecionar um tipo de configuração** e escolha **Kubernetes** no menu. Selecione o ícone de lápis. Insira os parâmetros necessários. Consulte Especificando parâmetros comuns para o Cloud Foundry Enterprise Environment.
8. Selecione **Salvar e Sair** .
9. A configuração é verificada. Selecione **Iniciar implementação** . A página `Estados` mostra o status de implementação e os arquivos de log.

Implemente o Cloud Foundry Enterprise Environment usando a CLI do gerenciador de configuração

Pré-requisito

1. O `ibm-cfee-installer` deve ser implementado com um nome do host que seja diferente do valor padrão `localhost` para poder acessar esse nome do host a partir de qualquer outra máquina.
2. O nome do host deve estar no diretório `/etc/hosts` ou registrado em um DNS.

Acessando a CLI do gerenciador de configuração

Quando o `ibm-cfee-installer` estiver instalado, execute as seguintes ações para acessar a CLI do gerenciador de configuração (CM) e iniciar a implementação do Cloud Foundry.

1. No diretório no qual o IBM Certified Container é descompactado, execute o comando a seguir:

   ```
   scripts/setup_client.sh -n <namespace> -hr <helm_release_name> -pn
   <ibm_cfee_inception_pod_name> -c <ICP hostname> -u <ICP User>`
   ```

   Este comando faz download dos modelos de configuração e da CLI do CM. Ele também configura a CLI do CM para acessar o gerenciador de configuração integrado no contêiner `ibm-cfee-inception`.

2. Escolha seu idioma e copie o modelo para um novo arquivo com o nome de sua escolha (extensão: .yml)
3. Ative `scripts/launch_deployment.sh -c <your_configuration_file>`.

IBM Cloud Private Cloud Foundry console de gerenciamento

O Ferramenta de implementação do Cloud Foundry instala uma liberação do Helm que fornece o console de gerenciamento do IBM Cloud Private Cloud Foundry.

1. No console do painel do IBM Cloud Private, abra **Cargas de Trabalho** > **Liberações do Helm**.
2. Localize e selecione a liberação do Helm. O nome da liberação corresponde ao nome que você escolheu para `ibm-cfee-installer` com o `-console` anexado. Por exemplo, se você usou `cfee`, a liberação para o console de gerenciamento do IBM Cloud Private Cloud Foundry será `cfee-console`. O nome do gráfico é `ibm-cf-ui`.
3. Na liberação do Helm, selecione **Ativar** para abrir o console de gerenciamento do IBM Cloud Private Cloud Foundry.

Guia do operador do IBM Cloud Private Cloud Foundry

Este guia contém as tarefas diárias para gerenciar sua implementação do IBM® Cloud Private Cloud Foundry. Como um operador, é possível customizar o ambiente, usar os recursos de criação de log e de monitoramento e gerenciar a segurança e a autenticação.

- Fazendo upgrade do IBM Cloud Private Cloud Foundry
- Desinstalando o IBM Cloud Private Cloud Foundry
- Controle de acesso baseado na função
- Customizando seu ambiente para o IBM Cloud Private Cloud Foundry
- Gerenciador de configuração (CM) - guia de referência rápida
- Criação de Log e Monitoramento
- Trabalhando com serviços
- Fazendo upgrade de versões secundárias de stemcell

Atualizando IBM Cloud Private Cloud Foundry

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

Conclua as etapas a seguir para fazer download e upgrade do IBM® Cloud Private Cloud Foundry.

O processo de upgrade requer aproximadamente de 2 a 4 horas. Essa estimativa não inclui o tempo que leva para fazer download dos arquivos de instalação.

O processo de upgrade consiste em três partes. É possível concluir as partes 1 e 2 simultaneamente. Conclua a parte 3 depois de concluir as partes 1 e 2.

- Fazendo upgrade do IBM Cloud Private Cloud Foundry
  - Antes de iniciar
  - Parte 1: preparar a nova mídia de instalação
  - Parte 2: verificar a instalação existente para o upgrade
  - Parte 3: fazer upgrade para a nova versão

Antes de iniciar

- Certifique-se de que todas as máquinas virtuais para todas as suas implementações estejam `em execução` e em um estado funcional. No contêiner de concepção, execute as verificações a seguir:

  1. Verifique todas as máquinas virtuais implementadas pelo BOSH. Assegure-se de que todas as máquinas virtuais estejam `em execução`.

     ```
     bosh -e IBMCloudPrivate vms
     ```

  2. Verifique os processos de `monitoramento` em todas as máquinas virtuais BOSH implementadas. Assegure-se de que todos os processos estejam `em execução`.

     ```
     bosh -e IBMCloudPrivate instâncias -- ps
     ```

  3. Use o BOSH para verificar os problemas de implementação. Assegure-se de que problemas zero sejam relatados.

     ```
     bosh -e IBMCloudPrivate -d Bluemix cck
     ```

  Nota: a migração da versão 2.1.0.3 ou anterior do IBM Cloud Private Cloud Foundry não é suportada. Uma nova instalação é necessária.

- Anote o diretório de dados para sua instalação existente. Esse diretório é o diretório que você especificou originalmente para o argumento `-c` no script `launch.sh`. Você precisa deste diretório nas partes 2 e 3. Substitua-o onde aparecer `<installation_configuration_directory>` nos comandos de exemplo.

  Além disso, inspecione o arquivo `uiconfig.yml` no `<previous_installation_directory>` para verificar se `developer_mode: false` está configurado. A migração não é suportada para o modo de desenvolvedor. Se você estiver usando o modo de desenvolvedor, crie uma nova implementação.

Parte 1: preparar a nova mídia de instalação

1. Faça download do arquivo de instalação completa no Passport Advantage (8,1 GB).
2. Para fazer upgrade do IBM Cloud Private Cloud Foundry, copie o arquivo de instalação para um servidor que tenha acesso ao Docker CE e ao ambiente IaaS.
3. Se o Docker Community Edition (CE) ainda não estiver instalado, instale o Docker CE. Para obter mais informações, consulte Instalar o Docker.
4. Crie um diretório a ser usado para o upgrade, referido como `<installation_directory>` posteriormente nas instruções.
5. Extraia o arquivo de instalação completo para o diretório.
6. Mude para o diretório.
7. Importe as imagens do Docker do diretório.

   ```
   import_images.sh
   ```

Parte 2: verificar a instalação existente para o upgrade

1. Mude para o diretório de instalação da versão anterior, referido como `<previous_installation_directory>` nos comandos.

2. Faça uma cópia da configuração para sua instalação anterior que você modificará para o processo de upgrade:

   ```
   cp < installation_configuration_directory> /uiconfig.yml < previous_installation_directory> /uiconfig_upgrade.yml
   ```

3. Inspecione o arquivo `uiconfig_upgrade.yml` e verifique se `developer_mode: false` está configurado. Se o `developer_mode: true` estiver configurado, não continue a migração. Você deve criar uma nova implementação.

Parte 3: fazer upgrade para a nova versão

1. Mude para o diretório da nova versão, referido como `<installation_directory>`.

2. Execute o script `launch.sh` para iniciar o contêiner do instalador. Substitua `<environment-name>` pelo mesmo nome usado para o contêiner na parte 2. `<installation_configuration_directory>` é o diretório de dados original da instalação anterior.

   ```
   ./launch.sh -n <environment_name> -c <installation_configuration_directory> -e LICENSE=accept
   ```

3. Faça uma cópia da nova versão do modelo `uiconfig`. O arquivo de modelo `uiconfig` está disponível depois de executar o comando `launch.sh`. Verifique o arquivo

`<installation_configuration_directory>/extensions/embedded/cfp-bosh-templates/uiconfig_<environment_type>_template.yml` em que `<environment_type>` depende da infraestrutura do IaaS (`vmware/openstack/aws`).

Por exemplo, se você usar o vSphere:

```
cp < installation_configuration_directory> /extensions/embedded/cfp-bosh-templates/uiconfig_vmware_template.yml < installation_directory> /uiconfig_upgrade.yml
```

4. Localize as chaves e valores que estão especificados nos conteúdos do arquivo de configuração da versão anterior, `<previous_installation_directory>/uiconfig_upgrade.yml`, que precisam ser preenchidos na versão mais recente e copie-os para o arquivo `<installation_directory>/uiconfig_upgrade.yml`. Conclua o valor para quaisquer novas chaves que forem necessárias pelo 3.2.0. Consulte Instalando o IBM Cloud Private Cloud Foundry para obter mais informações sobre os parâmetros de configuração.
    - **Mudanças de YAML customizadas** Como há mudanças na configuração entre o cf-release e o cf-deployment do IBM Cloud Private Cloud Foundry, quaisquer seções YAML customizadas que estão listadas no arquivo `uiconfig_upgrade.yml` podem precisar ser modificadas. Por exemplo, `cloud_config_custom`, `cf_custom`, e `director_custom`.
5. Faça upgrade para a nova versão.

   ```
   ./cm engine reset ./launch_deployment.sh -c uiconfig_upgrade.yml
   ```

   Esse comando ativa todas as etapas de implementação e inicia o upgrade do IBM Cloud Private Cloud Foundry. Se você tiver problemas, consulte Resolução de problemas do IBM Cloud Private Cloud Foundry.
6. Limpeza (opcional): no contêiner de concepção, execute o comando a seguir para remover quaisquer stemcells, liberações ou discos órfãos que você não está usando mais:

   ```
   bosh -e IBMCloudPrivate clean-up -- all
   ```

Desinstalando o IBM Cloud Private Cloud Foundry

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

Para desinstalar o IBM® Cloud Private Cloud Foundry, primeiro remova as implementações BOSH e, em seguida, remova o contêiner de instalação.

1. Efetue login no BOSH. Consulte Interfaces da linha de comandos para IBM® Cloud Private Cloud Foundry.
2. Remova a implementação do console do IBM Cloud Private Cloud Foundry .

   ```
   bosh -e IBMCloudPrivate -d Bluemix-cfp-ui delete-deployment --force
   ```
3. Remova a implementação do IBM Cloud Private Cloud Foundry .

   ```
   bosh -e IBMCloudPrivate -d Bluemix delete-deployment --force
   ```
4. Remova liberações, stemcells, tarefas e discos órfãos.

   ```
   bosh -e IBMCloudPrivate clean-up -- all
   ```
5. Remova o diretor BOSH.
    - Para VMware:

      ```
      bosh -e IBMCloudPrivate delete-env /data/gen-vmware_micro_boshinit.yml
      ```
    - Para OpenStack:

      ```
      bosh -e IBMCloudPrivate delete-env /data/gen-openstack_micro_boshinit.yml
      ```
6. Se você não planeja reinstalar o IBM Cloud Private Cloud Foundry, remova o contêiner de concepção do host de instalação.
    1. Obtenha o nome do contêiner de incepção. O nome do contêiner de concepção começa com `inception-`.

       ```
       docker ps
       ```
    2. Remova o contêiner de concepção.

```
docker rm <inception-name>
```

7. (Opcional) Desinstale o Docker.

8. Exclua o diretório no qual você armazenou os arquivos de configuração de instalação. Você especificou esse diretório como o argumento da opção `-c` quando executou o comando de instalação `launch.sh`.

Ações específicas do vSphere

Se não for possível concluir o procedimento de desinstalação, é possível que algumas partes da implementação do IBM® Cloud Private Cloud Foundry não tenham sido limpas corretamente. Use o procedimento a seguir para certificar-se de que o vSphere seja limpo adequadamente.

1. Conecte-se ao console do vSphere.
2. Localize a implementação. Dependendo de sua configuração, as máquinas virtuais poderão estar em um conjunto de recursos ou em um cluster.
3. Selecione todas as máquinas virtuais (VMs) no conjunto de recursos ou cluster. Clique com o botão direito do mouse e clique em **power** > **off** .
4. Selecione as VMs novamente. Clique com o botão direito e clique em **Excluir do disco**.
5. Se você não precisar mais do `cluster`, do `resource pool` e das `Virtual Machine Folders` que o IBM Cloud Private Cloud Foundry usou, será possível excluí-los.

Ações específicas do OpenStack

Se não for possível concluir o procedimento de desinstalação, é possível que algumas partes da implementação do IBM® Cloud Private Cloud Foundry não tenham sido limpas corretamente. Use o procedimento a seguir para certificar-se de que o OpenStack seja limpo adequadamente.

1. Conecte-se ao console do OpenStack.
2. Selecione `Project > Compute > Instances` e localize as instâncias que são usadas em sua implementação.
3. Selecione todas as instâncias que você deseja excluir.
4. Selecione `Excluir Instâncias`.
5. Selecione `Project > Compute > Volumes` e localize os volumes que são usados em sua implementação.
6. Selecione todos os volumes que você deseja excluir.
7. Selecione `Excluir Volumes`.
8. Selecione `Projeto > Computar > Imagens` e localize as imagens que são usadas em sua implementação.
9. Selecione todas as imagens que você deseja excluir.
10. Selecione `Excluir imagens`.

IBM Cloud Private Cloud Foundry controle de acesso baseado em função (RBAC)

O IBM® Cloud Private Cloud Foundry suporta várias funções. Sua função determina as ações que você pode fazer.

Para obter informações detalhadas sobre como gerenciar a segurança e o acesso à sua plataforma IBM® Cloud Private, consulte o guia de segurança do IBM Cloud Private.

Tabela 1. Funções RBAC para IBM Cloud Private Cloud Foundry

Função	Escopo	Ações

Função	Escopo	Ações
Administrador	Organização	• Designar funções do usuário • Visualizar usuários e funções • Criar e designar planos de cota de organização • Visualizar planos de cota da organização • Criar organizações • Visualizar organizações • Visualizar organizações das quais o usuário é um membro • Editar, renomear e excluir organizações • Suspender ou ativar uma organização
	Espaço	• Crie • Visualizar • Edição • Excluir • Renomear
	Instâncias	• Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso
	Domínios compartilhados e privados	• Incluir • Compartilhar com outras organizações
	Aplicativos	• Implementar • Execute o • Gerenciar • Instanciar e ligar serviços • Associar rotas • Contagens de Instância • Alocação de • Limites de disco • Renomear • Criar e gerenciar grupos de segurança de aplicativos

Função	Escopo	Ações
	Segmentos de Isolamento	• Crie • Atualizar • Excluir • Listar segmentos de isolamento para uma organização • Intitulação • Revogar • Listar organizações intituladas • Designar um padrão a uma organização • Listar e gerenciar • Listar segmentos de isolamento autorizados para um espaço • Listar quais apps estão em execução
	Eventos de uso	• Listar aplicativo • Listar Serviço
Administrador somente leitura	Organização	• Visualizar usuários • Visualizar funções • Visualizar planos de cota da organização • Visualizar organizações • Visualizar organizações das quais o usuário é um membro
	Espaço	• Espaços de Visualização • Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso de aplicativos
	Segmentos de Isolamento	• Listar segmentos de isolamento para uma organização • Listar organizações intituladas • Listar espaços autorizados • Listar quais aplicativos são executados em segmentos de isolamento
	Eventos de uso	• Listar aplicativo • Listar Serviço

Função	Escopo	Ações
Auditor global	Organização	• Visualizar usuários • Visualizar funções • Visualizar planos de cota da organização • Visualizar organizações • Visualizar organizações das quais o usuário é um membro
	Espaço	• Espaços de Visualização • Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso de aplicativos
	Segmentos de Isolamento	• Listar espaços para uma organização • Listar organizações intituladas • Listar espaços autorizados • Listar quais aplicativos são executados nos espaços
Gerente da organização	Organização	• Designar funções do usuário • Visualizar usuários • Visualizar funções • Visualizar planos de cota da organização • Visualizar organizações • Visualizar organizações das quais o usuário é um membro • Edição • Renomear
	Espaço	• Criar e designar planos de cota • Crie • Visualizar • Edição • Excluir • Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso de apps
	Domínios	• Incluir domínios particulares • Incluir domínios compartilhados quando eles tiverem acesso a todas as organizações

Função	Escopo	Ações
	Segmentos de Isolamento	• Crie • Atualizar • Excluir • Listar segmentos de isolamento para uma org • Intitulação • Revogar • Listar organizações intituladas • Designar um padrão a uma organização • Listar e gerenciar • Listar segmentos de isolamento autorizados para um espaço • Listar quais apps estão em execução
auditor da organização	Organização	• Visualizar usuários • Visualizar funções • Visualizar planos de cota da organização • Visualizar organizações das quais o usuário é um membro
	Segmentos de Isolamento	• Listar segmentos de isolamento para uma organização • Listar organizações intituladas
Gerenciador de faturamento da organização	Organização	• Visualizar usuários • Visualizar funções • Visualizar planos de cota da organização • Visualizar organizações das quais o usuário é um membro
	Segmentos de Isolamento	• Listar segmentos de isolamento para uma organização • Listar organizações intituladas
Gerenciador de Espaços	Espaço	• Designar funções do usuário • Visualizar usuários e funções • Visualizar planos de cota da organização • Visualizar organizações das quais o usuário é um membro • Visualizar • Edição • Renomear • Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso de aplicativos

Função	Escopo	Ações
	Segmentos de Isolamento	• Listar segmentos de isolamento para uma organização • Listar organizações intituladas • Listar segmentos de isolamento autorizados para um espaço • Listar quais apps estão em execução
Desenvolvedor de espaço	Espaço	• Visualizar usuários e funções • Visualizar planos de cota da organização • Visualizar organizações das quais o usuário é um membro • Visualizar • Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso de aplicativos
	Aplicativos	• Implementar • Execute o • Gerenciar • Instanciar serviço • Serviço de Ligação • Associar rotas • Contagens de Instância • Alocação de • Limite de disco • Renomear
	Segmentos de Isolamento	• Listar segmentos de isolamento para uma organização • Listar organizações intituladas • Listar segmentos de isolamento autorizados para um espaço • Listar quais apps estão em execução
	Eventos de uso	• Listar aplicativo • Listar Serviço

Função	Escopo	Ações
Auditor de Espaço	Espaço	• Visualizar usuários e funções • Visualizar planos de cota da organização • Visualizar organizações das quais o usuário é um membro • Visualizar • Visualizar status • Número de instâncias • Ligações de serviços • Uso de recurso de aplicativos
	Segmentos de Isolamento	• Listar segmentos de isolamento para uma organização • Listar organizações intituladas • Listar segmentos de isolamento autorizados para um espaço • Listar quais apps estão em execução
	Eventos de uso	• Listar aplicativo • Listar Serviço

Customizando seu ambiente do IBM Cloud Private Cloud Foundry

Customize seu ambiente para IBM Cloud Private Cloud Foundry.

- Configurando a autenticação para o IBM® Cloud Private Cloud Foundry
- Implementando o banco de dados Open Service Broker no IBM Cloud Private IBM® Cloud Private Cloud Foundry
- Extensão do instalador do Open Service Broker
- Registrando o IBM Cloud e plataformas adicionais do Cloud Foundry com o console
- Usando os serviços do IBM Cloud no IBM Cloud Private Cloud Foundry
- Configurando o Director para usar um banco de dados diferente
- Configurando bancos de dados remotos para o IBM Cloud Private Cloud Foundry
- Configurando a rede do contêiner
- Configurando backups para o IBM Cloud Private Cloud Foundry
- Configurando zonas de disponibilidade
- Configurando segmentos de isolamento no IBM Cloud Private Cloud Foundry
- Aumentando o número de células do Diego
- Configurar certificados de confiança para aplicativos para o IBM Cloud Private Cloud Foundry
- Usando extensões no IBM Cloud Private Cloud Foundry

Configurar autenticação para IBM® Cloud Private Cloud Foundry

É possível configurar a autenticação para o IBM Cloud Private Cloud Foundry antes ou depois da instalação.

- Configurando a autenticação LDAP para o IBM Cloud Private Cloud Foundry
- Configurando a autenticação da UAA para o IBM Cloud Private Cloud Foundry
- Gerenciando permissões de usuário para organizações e espaços

Configurando a autenticação LDAP para IBM Cloud Private Cloud Foundry

É possível usar a autenticação LDAP para autenticação do usuário no IBM® Cloud Private Cloud Foundry e no Cloud Foundry Enterprise Environment.

É possível seguir estas instruções para configurar a autenticação LDAP a qualquer momento antes ou depois de instalar o IBM Cloud Private Cloud Foundry (consulte Instalando o IBM Cloud Private Cloud Foundry) ou o Cloud Foundry Enterprise Environment (consulte Instalando o Cloud Foundry Enterprise Environment). A configuração é feita incluindo a extensão `cfp-ext-ldap` adicionada na implementação principal do IBM Cloud Private Cloud Foundry. Consulte Usando extensões no IBM® Cloud Private Cloud Foundry. A extensão `cfp-ext-ldap` modifica as propriedades de implementação para o servidor User Account and Authentication (UAA).

Antes de configurar a autenticação LDAP, deve-se criar os usuários em seu domínio do LDAP ou do Active Directory. É possível usar grupos para filtrar o acesso ao ambiente do IBM Cloud Private Cloud Foundry.

Preparando sua Configuração

A configuração LDAP é armazenada em um arquivo formatado em YAML, referido como `<ldap_config_file>` nestas instruções. Crie o arquivo, por exemplo, `ldapconfig.yml` e coloque os detalhes de configuração na seção `uiconfig:`, conforme mostrado na configuração de exemplo.

Se você preferir usar o Ferramenta de implementação do Cloud Foundry, a extensão `cfp-ext-ldap` oferecerá a edição orientada dos valores de configuração. Na página `Estados`, clique em `Incluir extensão` e selecione `cfp-ext-ldap` e, em seguida, conclua o diálogo. Acesse a página `Configuração` e localize `cfp-ext-ldap` na lista de extensões. Selecione um tipo de configuração, como `Ligação simples`, `Procurar e ligar`, `Mapa de grupos para escopos` ou `Grupos como escopos`. O Ferramenta de implementação do Cloud Foundry exibe os valores de configuração obrigatórios e opcionais para o cenário selecionado. O Ferramenta de implementação do Cloud Foundry fornece descrições, valores de amostra e validação dos valores de configuração. Para obter informações sobre extensões, consulte Gerenciando extensões e Configurações.

Para configurar a autenticação LDAP, deve-se fazer as seguintes opções:

- Método de ligação LDAP
  - *simple-binding*

    Se o formato do nome distinto (DN) para usuários for conhecido e contiver o identificador que você deseja usar como o nome do usuário, use a configuração *simple-binding* e forneça um ou mais padrões de DN para construção. O UAA constrói os DNs e, em seguida, tenta autenticar-se com a senha fornecida pelo usuário. As linhas aplicáveis estão marcadas na configuração de exemplo.

  - *search-and-bind*

    Se o formato do DN para usuários for desconhecido ou variar, será possível usar a configuração *search-and-bind* que procura pelo LDAP de um usuário correspondente. Uma razão comum para usar o *search-and-bind* seria se você deseja usar os endereços de e-mail, já que os nomes de usuário e os nomes comuns (CNs) para registros do usuário em seu servidor LDAP não contêm os endereços de e-mail. Nesse caso, é necessário fornecer o nome de usuário e a senha para um usuário LDAP que tenha acesso (de preferência somente leitura) ao servidor para recuperar registros e procurar por uma correspondência para o usuário. Se houver exatamente uma correspondência com relação ao filtro de procura, então o UAA tentará autenticar-se com a senha fornecida pelo usuário. As linhas aplicáveis estão marcadas na configuração de exemplo. Teste seus valores de configuração substituindo-os no comando `ldapsearch -H <url> -D <userDN> -w <userPassword> -b <searchBase> <searchFilter>` e substituindo '{0}' no valor `<searchFilter>` por um nome do usuário real.

- Mapeamento de grupos
  - Por padrão, os usuários LDAP possuem os seguintes escopos de privilégio:

```
cloud_controller.read
cloud_controller.write
cloud_controller_service_permissions.read
notification_preferences.read
notification_preferences.write
oauth.approvals
openid
password.write
profile
roles
scim.me
uaa.offline_token
uaa.user
user_attributes
```

Se desejar conceder escopos de privilégios adicionais a usuários LDAP com base em sua associação ao grupo, há duas opções:

- *groups-map-as-scopes*

  Use a CLI `uaac` para gerenciar o mapeamento entre grupos LDAP e escopos de privilégio. Os escopos são recalculados em cada login de usuário, portanto, é possível remover privilégios removendo o mapeamento de grupos usando `uaac`.

- *grupos-como-escopos*

  Especifique os escopos de privilégio no valor de um atributo do grupo LDAP. Isso requer acesso administrativo ao seu servidor LDAP.

- Método de ativação
  - Por padrão, todos os usuários que podem ser autenticados usando o método de ligação LDAP escolhido são automaticamente ativados quando efetuam login pela primeira vez no IBM Cloud Private Cloud Foundry. Se preferir permitir o login somente para usuários que são ativados manualmente, configure `add_shadow_user_on_login` como `false`.

Nota: Se estiver incluindo LDAP para o Cloud Foundry Enterprise Environment após uma implementação inicial, deve-se selecionar `upgrade` para a ação do Instalador.

Configuração de exemplo

```
uiconfig:
  enabled: true
  url: 'ldap://ldap.local.bluemix.net'  # LDAP server URL (space-separated list permitted)
  add_shadow_user_on_login: true        # Enable to allow any LDAP user to log in; otherwise,
accounts
                                        #   must be created manually (default value is true)

# BEGIN Include and customize the following lines to use simple-binding
            profile_type: simple-bind
            userDNPattern: 'cn={0},dc=local,dc=bluemix,dc=net;cn={0},dc=example,dc=org'
# FIM

# BEGIN Include and customize the following lines to use search-and-bind
            profile_type: search-and-bind
            userDN: 'cn=admin,dc=local,dc=bluemix,dc=net'
            userPassword: '**********'
  searchBase: 'dc=local,dc=bluemix,dc=net'  # Specify if only part of the directory should be
searched
  searchFilter: 'cn={0}'                    # User name is inserted into this filter in place of {0}
# END

# BEGIN Include the following lines to customize which LDAP attributes are used to populate the UAA
user record
  mailAttributeName: emailAddress       # Attribute containing email address of user (default is
'mail')
  mailSubstitute: 'generated-{0}@ldap'  # Form for generated email address if none found in LDAP
record
                                        #   (default is {0}@user.from.ldap.cf)
  mailSubstituteOverridesLdap: false    # Enable to always use the value from mailSubstitute as user
email
  attributeMappings:
    family_name: lastName               # LDAP attribute to use for family name (default value is
sn)
    given_name: preferredName           # LDAP attribute to use for given name (default value is
givenName)
# FIM

# BEGIN Include and customize the following lines to use groups-map-to-scopes or groups-as-scopes
#         (Requires using the search-and-bind binding method)
  groups:
    groupSearchFilter: 'member={0}'           # Used to find the groups a user (or group for nested
search) belongs to
    searchBase: 'dc=local,dc=bluemix,dc=net'  # Specify if only part of the directory should be
searched for groups
                                              #   Can be set to 'memberOf' if using Active Directory
to skip group search
                                              #   and use calculated 'memberOf' field on user
```

```
records
    maxSearchDepth: 2                          # Number of levels to search groups (default is 1 - no
nested search)
    searchSubtree: true                        # Enable to search below the search base (default
value is true)

    # BEGIN Include the following line to use groups-map-to-scopes
              profile_type: groups-map-to-scopes
    # END

    # BEGIN Include and customize the following lines to use groups-as-scopes
              profile_type: groups-as-scopes
              groupRoleAttribute: cloudFoundryScope  # Name of the LDAP attribute that holds a list
(comma-separated)
                                                     #   of scope names applied to members of the
group
    # END

# END

  ssl:
              skipverification: false  # Enable to skip server certificate validation when using
LDAPS
              tls: none                # Use value 'simple' to enable StartTLS (default value is
none)
            sslCertificate: |+         # Self-signed server certificate to be trusted if using LDAPS
              -----BEGIN CERTIFICATE-----
              <BASE64_ENCODED_CERT>
              -----END CERTIFICATE-----
```

Essas instruções e a configuração de exemplo abrangem as opções mais comuns para autenticação LDAP. Para obter a lista completa de propriedades válidas, consulte as propriedades uaa.ldap para a Tarefa uaa do BOSH. Para obter detalhes adicionais, consulte Integração LDAP de conta e autenticação do usuário.

Ativando a extensão `cfp-ext-ldap` usando a CLI do gerenciador de configuração

1. Envie por push o seu arquivo de configuração para o contêiner de concepção. `<ldap_config_file>` é o nome do arquivo de configuração LDAP que foi criado:

   ```
   ./cm extension -e cfp-ext-ldap save -c <ldap_config_file>
   ```

2. Insira a extensão na implementação principal do IBM Cloud Private Cloud Foundry ou do Cloud Foundry Enterprise Environment. O comando a seguir inclui a implementação da extensão LDAP antes da implementação dos componentes IBM Cloud Private Cloud Foundry ou Cloud Foundry Enterprise Environment. A extensão LDAP é inserida automaticamente no local correto no arquivo de estados.

   ```
   ./cm states insert -i cfp-ext-ldap
   ```

3. Reconfigure o mecanismo:

   ```
   ./cm engine reset
   ```

Implementando o Cloud Foundry com a extensão `cfp-ext-ldap`

Depois de ativar e configurar a extensão `cfp-ext-ldap` usando o Ferramenta de implementação do Cloud Foundry ou a CLI do gerenciador de configuração, deve-se reativar a implementação do Cloud Foundry para atualizar os componentes afetados.

- Para o IBM Cloud Private Cloud Foundry, siga as instruções na seção anterior para ativar a extensão cfp-ext-ldap. Reinicie o processo de implementação do IBM Cloud Private Cloud Foundry. `<cf_config_file>` é seu arquivo de configuração principal para a implementação do IBM Cloud Private Cloud Foundry, como `uiconfig.yml`:

  ```
  ./launch_deployment.sh -c <cf_config_file>
  ```

  É possível usar o Ferramenta de implementação do Cloud Foundry clicando no botão `Iniciar implementação` na página `Estados`.

- Para o Cloud Foundry Enterprise Environment, use o Ferramenta de implementação do Cloud Foundry para mudar a `ação do Instalador` para `Fazer upgrade` na configuração principal. Clique em `Iniciar implementação` na página `Estados`.

Ativando usuários

Se optar por ativar manualmente os usuários, para cada usuário que precisa acessar o IBM Cloud Private Cloud Foundry, execute o seguinte comando. `<username>` é um ID do usuário LDAP que corresponde à propriedade `userDNPattern` ou `searchFilter` na configuração LDAP, dependendo do método de ligação selecionado:

```
cf create-user < username> -- origin ldap
```

Revogando o acesso de usuário

Se optar por ativar os usuários manualmente, é possível revogar o acesso para um usuário excluindo sua conta:

```
cf delete-user <username>
```

Caso contrário, além de excluir o usuário usando a CLI do `cf`, para revogar o acesso de usuário, o usuário deve ser removido do LDAP. Ou a associação ao grupo do usuário deve ser mudada de forma que ele não atenda mais aos critérios, conforme definido pela configuração de ligação LDAP.

Designando Usuários a Organizações e Espaços

Os usuários LDAP não podem ter acesso concedido automaticamente a organizações ou espaços específicos, mas é possível conceder permissão antes do primeiro login, primeiro ativando o usuário com o comando `cf create-user <username> --origin ldap` e, em seguida, designando funções de organização e de espaço de forma usual. Para obter mais informações, consulte Gerenciando permissões de usuários para organizações e espaços. Essas ações podem ter o script definido se você tiver muitos usuários LDAP que deseja designar a determinadas organizações e espaços, com base nas informações em seus registros LDAP.

Mapeando grupos de LDAP para escopos

Se você escolheu usar `groups-map-to-scopes` para designar automaticamente os escopos de privilégio extras para usuários LDAP com base na associação ao grupo, é possível usar a CLI `uaac` para gerenciar os mapeamentos.

1. Instale o gem Ruby `uaac` usando o seguinte comando:

   ```
   gem instalar cf-uaac
   ```

2. Direcione o servidor da UAA de sua instalação. Substitua `<bluemix_env_domain>` pelo domínio de ambiente usado para sua instalação. Inclua a opção `--skip-ssl-validation` no comando se o seu certificado de domínio do ambiente estiver autoassinado.

   ```
   uaac target https://uaa.<bluemix_env_domain>
   ```

3. Localize o segredo do cliente administrador de UAA.

   - Para o IBM Cloud Private Cloud Foundry, efetue login com a CLI `uaac` localizando o valor `uaa_admin_client_secret` no arquivo `<data_directory>/CloudFoundry/deployment-vars.yml`.
   - Para o Cloud Foundry Enterprise Environment, obtenha o valor `<uaa_admin_client_secret>` executando o seguinte comando:

     ```
     kubectl get -n uaa secret secrets -o jsonpath='{.data.uaa-admin-client-secret}' | base64 --decode
     ```

   Substitua o segredo do cliente administrador de UAA no seguinte comando:

   ```
   uaac token client get admin -s <uaa_admin_client_secret>
   ```

Agora é possível gerenciar mapeamentos de grupo emitindo comandos, conforme mostrado no exemplo a seguir:

```
uaac group mappings                        # List all group mappings
uaac group map --name <scope> <groupDN>  # Create a new group mapping by specifying the privilege
scope
                                           #   and distinguished name (DN) of the LDAP group
uaac group unmap <scope> <groupDN>       # Delete a group mapping by specifying the privilege scope
                                           #   and distinguished name (DN) of the LDAP group
uaac user get <username>                 # Retrieve user information to verify which groups have
been mapped
```

Configurando a autenticação UAA para IBM® Cloud Private Cloud Foundry

O User Account & Authentication (UAA) é o método de autenticação padrão para o IBM Cloud Private Cloud Foundry.

Ao instalar o IBM Cloud Private Cloud Foundry, use os parâmetros `main_user_name` e `main_user_password` para definir um nome de usuário administrativo e senha.

Criando usuários com a CLI cf

1. Instale a CLI cf. Para obter mais informações sobre a instalação, consulte Interfaces da linha de comandos para o IBM® Cloud Private Cloud Foundry.
2. Efetue login no IBM Cloud Private Cloud Foundry e execute os comandos a seguir:

   ```
   cf api https://api.<my_domain>
   cf login
   ```

   em que `<my_domain>` é o domínio da API.

   Quando solicitado, insira as credenciais do administrador.
3. Crie um usuário:

   ```
   cf create-user <username> <password>
   ```

   em que `<username>` é o nome do usuário e `<password>` é a senha associada.
4. Conceda as permissões de usuários para suas organizações e espaços. Para obter mais informações, consulte Gerenciando permissões de usuários para organizações e espaços.

Excluindo usuários

Execute o comando a seguir para excluir um usuário do UAA interno:

```
cf delete-user <username> [-f]
```

em que `<username>` é o nome do usuário. A opção `-f` exclui o usuário sem confirmação da CLI.

Gerenciando permissões de usuário para organizações e espaços

Depois de criar usuários, será possível conceder permissão a eles para acessar suas organizações e espaços.

Antes de conceder aos usuários permissões para uma organização ou espaço, execute as ações a seguir:

- Crie os usuários. Consulte Configurar autenticação para IBM® Cloud Private Cloud Foundry.
- Assegure-se de que o objeto ao qual você planeja conceder acesso exista. Execute `cf orgs` ou `cf spaces` e confirme se a organização ou o espaço existe. (requer `cf target -o ORG`)
- Assegure-se de que o usuário tenha efetuado login no IBM Cloud Private Cloud Foundry. Se ele não efetuou, não será possível conceder acesso a ele para uma organização ou espaço.

Concedendo acesso a uma organização

Para conceder a um usuário acesso a uma organização, execute este comando:

```
cf set-org-role <username> <org> <role>
```

Em que `<username>` é o nome do usuário, `<org>` é a organização Cloud Foundry e `<role>` é uma das seguintes funções:

- **OrgManager**: o usuário pode convidar e gerenciar usuários, selecionar e mudar planos e configurar limites de gastos.
- **BillingManager**: o usuário pode criar e gerenciar as informações da conta de cobrança e de pagamento.
- **OrgAuditor**: o usuário tem acesso somente leitura a informações da organização e relatórios.

Concedendo acesso a um espaço

Nota: deve-se executar os comandos a seguir usando o Cloud Foundry CLI versão 6.13 ou anterior.

Para conceder a um usuário acesso a um espaço, execute este comando:

```
cf set-space-role <username> <org> <space> <role>
```

Em que `<username>` é o nome do usuário, `<org>` é a organização Cloud Foundry e `<role>` é uma das seguintes funções:

- **SpaceManager**: o usuário pode convidar e gerenciar usuários e ativar recursos para o espaço.
- **SpaceDeveloper**: o usuário pode criar e gerenciar aplicativos e serviços e visualizar logs e relatórios.
- **SpaceAuditor**: o usuário pode visualizar logs, relatórios e configurações para o espaço.

Removendo permissões para organizações e espaços

Para remover permissões de um usuário para uma organização, execute o comando a seguir:

```
cf unset-org-role <username> <org> <role>
```

Para remover permissões de um usuário para um espaço, execute o comando a seguir:

```
cf unset-space-role <username> <org> <space> <role>
```

Implementando o banco de dados Open Service Broker no IBM Cloud Private Cloud Foundry

O IBM® Cloud Private Cloud Foundry fornece um gráfico Helm chamado `ibm-osb-database`. É possível instalar o `ibm-osb-database` para ser implementado como um broker de serviço de cluster no IBM Cloud Private. O broker de serviço de cluster é construído para a especificação da API do Open Service Broker (OSB).

O broker de serviço oferece gráficos Helm de banco de dados do IBM Cloud Private como serviços. É possível registrar o broker de serviço no IBM® Cloud Private Cloud Foundry para provisionar e desprovisionar instâncias de serviço e ligar instâncias de serviço aos aplicativos.

Para obter informações sobre a extensão do Cloud Foundry Enterprise Environment que facilita a configuração do broker de serviço de cluster, consulte Extensão do instalador do Open Service Broker.

Pré-requisitos

- Ambiente com o IBM Cloud Private Versão 3.1 ou mais recente.
- O IBM Cloud Private Cloud Foundry deve ser instalado, já que o gráfico Helm `ibm-osb-database` é exportado durante a instalação.
- Instale os comandos gerais da CLI do IBM Cloud Private (cloudctl) e a CLI kubectl. Efetue login em seu cluster. A CLI kubectl é configurada automaticamente ao efetuar login usando cloudctl. Para obter informações sobre a instalação da CLI, consulte Instalando a CLI do IBM Cloud Private.
- Instale e configure a CLI do Docker para usar com o cluster. Para obter informações sobre a instalação da CLI, consulte Configurando a autenticação para a CLI do Docker.

Carregando o archive de gráfico

Depois de instalar o IBM Cloud Private Cloud Foundry, é possível localizar o archive de gráfico no diretório `<data_directory>/IBMCloudPrivate` no sistema onde foi executado o instalador do IBM Cloud Private. `<data_directory>` é o diretório fornecido para o script `launch.sh` usando a opção `-c`.

O archive de gráfico é denominado `ibm-osb-database-1.0.0-archive.tgz` e contém o gráfico Helm e uma imagem necessária. Se você instalou e configurou as CLIs (conforme descrito na seção de pré-requisitos) em um sistema diferente, copie o archive para esse sistema. Assegure-se de ter efetuado login em seu cluster do IBM Cloud Private e de que sua CLI do Docker esteja com login efetuado no registro de imagem privado para seu cluster. Em seguida, execute o comando a seguir:

```
cloudctl catalog load-archive -- archive ibm-osb-database-1.0.0-archive.tgz
```

Por padrão, o comando carrega o gráfico no repositório do Helm `local-charts` e a imagem no registro de imagem privado, no qual ele é acessível apenas por gráficos que estão instalados no namespace de destino atual. Para obter informações sobre comandos do catálogo da CLI, consulte IBM Cloud PrivateComandos do catálogo CLI. Para obter informações sobre como gerenciar imagens, consulte Gerenciando imagens.

Criando segredos

Deve-se criar um segredo do Kubernetes que contenha o nome do usuário e a senha de autenticação do broker de serviço no namespace no qual o gráfico do Helm do broker de serviço está instalado. Por exemplo, crie um arquivo YAML que seja denominado `cf-osb-broker-secret.yaml` com o conteúdo a seguir. Substitua o nome do usuário do broker de serviço e a senha do broker de serviço pelo nome do usuário e a senha codificados em base-64.

```
apiVersion: v1
kind: Secret
metadata:
  name: cf-osb-broker-secret
type: Opaque
data:
  username: service broker user name
  password: service broker password
```

Para codificar uma sequência em base-64, é possível usar o comando a seguir, por exemplo.

```
$ echo -n 'stringToEncode' | openssl base64
```

Em seguida, execute o comando a seguir para criar o segredo do Kubernetes.

```
$kubectl create -f cf-osb-broker-secret.yaml
```

Deve-se criar outro segredo do Kubernetes que contenha o nome do usuário e a senha de login do IBM Cloud Private no namespace no qual o gráfico do Helm do broker de serviço está instalado. Essa credencial é usada pelo broker de serviço para provisionar instâncias de serviço. Por exemplo, crie um arquivo YAML que seja denominado `cf-osb-icp-secret.yaml` com o conteúdo a seguir. Substitua o nome do usuário do IBM Cloud Private e a senha do IBM Cloud Private pelo nome do usuário e senha codificados por base-64. Você deve ser capaz de efetuar login no IBM Cloud Private usando esse nome do usuário e senha para visualizar e instalar os gráficos do Helm.

```
apiVersion: v1
kind: Secret
metadata:
  name: cf-osb-icp-secret
type: Opaque
data:
  username: IBM Cloud Private user name
  password: IBM Cloud Private password
```

Para codificar uma sequência em base-64, é possível usar o comando a seguir, por exemplo.

```
$ echo -n 'stringToEncode' | openssl base64
```

Em seguida, execute o comando a seguir para criar o segredo do Kubernetes.

```
$kubectl create -f cf-osb-icp-secret.yaml
```

Instalando o gráfico

IBM Cloud Private Gerenciador

Localize e clique no gráfico `ibm-osb-database` no catálogo. A visão geral contém informações detalhadas sobre todos os parâmetros de configuração do gráfico. Conclua as etapas a seguir para configurar seu gráfico:

1. Alterne para a guia `Configuração` ou clique em `Configurar`.
2. Insira um nome exclusivo para o `Nome da liberação do Helm`.
3. Selecione o namespace de destino.
4. `Aceite` a licença.
5. Forneça valores necessários para os parâmetros do aplicativo.
6. Clique em `Instalar` para concluir sua configuração.

CLI do Helm

Se você preferir usar a CLI do Helm, consulte Instalando as CLI do Helm (helm) para obter instruções sobre a instalação da CLI do Helm. O Helm é configurado automaticamente quando você efetua login usando `cloudctl`.

Execute o comando a seguir para instalar o gráfico.

```
helm install local-charts/ibm-osb-database-1.0.0.tgz --name <release_name> --namespace <namespace> --tls
```

Use uma das opções a seguir para configurar valores para os campos:

- Use a opção `--set` para configurar esses valores. Por exemplo: `-- set brokerconfig.userToken="YWRtaW4 = ", brokerconfig.password="YWRtaW4 ="`
- Crie um arquivo YAML contendo os valores e use a opção `--values` para fornecer o arquivo YAML.

Forneça os valores necessários para os campos a seguir.

```
brokerconfig.servicebrokersecret="<Kubernetes secret object name that contains the service broker's user name and password. Defaults to 'cf-osb-broker-secret'>"
brokerconfig.icpsecret="<Kubernetes secret object name that contains IBM Cloud Private's user name and password. Defaults to 'cf-osb-icp-secret'. > "
brokerconfig.externalClusterIp = "" brokerconfig.namespace = ""
```

Expondo o broker de serviço para acesso externo

Conclua as etapas a seguir para expor o broker de serviço para acesso externo:

1. Liste o nome do serviço interno do broker de serviço.

   ```
   $kubectl get services
   ```

2. Crie um `NodePort` para expor o broker fora do cluster.

   ```
   kubectl expose deployment <helm_release_name>-ibm-osb-database --name <helm_release_name>-ibm-osb-database-external --type=NodePort --port=80 --target-port=8080
   ```

   Nota: O TLS é desativado por padrão. Se você ativou o TLS, execute então o seguinte comando:

   ```
   kubectl expose deployment <helm_release_name>-ibm-osb-database --name <helm_release_name>-ibm-osb-database-external --type=NodePort --port=443 --target-port=8443
   ```

3. Verifique a porta exposta e obtenha o número da porta.

   ```
   $kubectl get services
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
   dbservicebroker1-ibm-osb-database ClusterIP 10.0.0.248 <none> 443/TCP 3m
   dbservicebroker1-ibm-osb-database-external NodePort 10.0.0.196 <none> 443:32055/TCP 5s
   kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 47d
   ```

   Nesse caso, o número da porta do nó externo é 32055. É necessário o número da porta para registrar o broker de serviço no IBM Cloud Private Cloud Foundry.

Extensão do instalador do Open Service Broker

O Cloud Foundry Enterprise Environment fornece uma extensão que facilita configurar o broker de serviço de cluster.

Pré-requisitos

Ambiente com o IBM® Cloud Private 3.2.0 ou mais recente.

Implementando a extensão

1. Abra o Ferramenta de implementação do Cloud Foundry.
2. Acesse **Menu** > **Configuração**.
3. Selecione **Configuração do instalador do broker de serviço** no menu.
4. Clique no botão de lápis para editar o arquivo de configuração.
5. Preencha todos os campos obrigatórios no arquivo de configuração.
6. Clique em **Salvar e Sair**.
7. É possível verificar os valores de configuração clicando no botão de papel.
8. Acesse **Menu** > **Estados**.

9. Clique em `cfp-ext-osb-installer`.
10. Selecione a primeira etapa e clique em **Iniciar esta extensão**.
11. Se a implementação falhar, clique no botão de papel para ver o log, que tem sugestões sobre como recuperar a implementação.

Registrando o IBM Cloud e plataformas adicionais do

Cloud Foundry com o console

Use o console do IBM Cloud Private Cloud Foundry para criar e gerenciar seus aplicativos e serviços de limite do IBM Cloud Private Cloud Foundry.

Efetuar login no IBM Cloud Private Cloud Foundry console

1. Na janela de login , use as credenciais **Nome do usuário administrador** e **Senha do administrador** que estão no arquivo `./uiconfig.yml`:

   ```
   main_user_name main_user_password
   ```

2. Selecione **Terminais** na barra de navegação.
3. Selecione o sinal de mais para registrar um novo terminal.
4. Especifique os valores a seguir para registrar um terminal do IBM Cloud Private Cloud Foundry (implementação):
   - **Endereço de terminal**: especifique a URL apontando para a API do IBM Cloud Private Cloud Foundry `https://api.BLUEMIX_ENV_DOMAIN`, por exemplo: `https://api.cf.ibm.com`.
   - **Nome**: especifique um rótulo arbitrário que representa o terminal do IBM Cloud Private Cloud Foundry que está sendo registrado.
   - **Ignorar a validação de SSL para o terminal**: selecione somente quando você estiver usando certificados autoassinados com o IBM Cloud Private Cloud Foundry.
5. Depois de ter especificado todos os campos obrigatórios, clique em **Registrar**.
6. Na janela de registro de terminal do IBM Cloud Private Cloud Foundry, clique no menu overflow e selecione **Conectar**.
7. Na caixa de diálogo **Fornecer credenciais**, especifique o **Nome do usuário administrador** e a **Senha do administrador**.
8. Clique em **Conectar**.

Seu console do IBM Cloud Private Cloud Foundry agora está pronto para uso com o terminal.

Configurando o serviço do Director para usar um banco de dados diferente

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

Por padrão, o serviço do Director possui um banco de dados Postgres interno. Esse banco de dados é uma instância única de uma origem de dados não em cluster. É possível configurar o serviço do Director para usar um banco de dados Postgres ou MySQL externo diferente.

No servidor de banco de dados externo, crie os bancos de dados a seguir:

- Bosh
- credhub
- uaa

Para usar um banco de dados diferente, crie um arquivo `director-ext-db-vars.yml` com o conteúdo a seguir no diretório de configuração de instalação, por exemplo, `/home/user/data`. A existência do arquivo `director-ext-db-vars.yml` aciona a configuração de serviço do Director com o banco de dados externo.

Para obter mais informações sobre o diretório de configuração de instalação, consulte Instalando o IBM Cloud Private Cloud Foundry.

```
db_host: <The IP address or fully qualified domain name (FQDN) for the Postgres or MySQL server.>
db_port: <The database port number.>
db_user: <The user name to connect to the database.>
db_password: <The password for the specified database user.>
db_adapter: <postgres or mysql>
db_scheme: <postgresql or mysql>
```

Como uma confirmação, procure pela mensagem `Configurar o Director com um banco de dados externo` no `prepare-director.log`.

Para obter mais informações sobre como configurar o banco de dados do Director, consulte Conectando o Director a um banco de dados Postgres externo.

Configurando bancos de dados remotos para o IBM Cloud Private Cloud Foundry

É possível usar uma extensão para modificar o manifest de implementação do IBM® Cloud Private Cloud Foundry para usar bancos de dados externos.

Antes de iniciar

Preparar bancos de dados externos

Deve-se preparar bancos de dados externos antes de aplicar a extensão. Todos os bancos de dados necessários podem ser criados em servidores de banco de dados diferentes. O tipo de servidor de banco de dados e o número da porta de conexão devem ser os mesmos. MySQL e Postgres são suportados atualmente.

1. Efetue login no servidor ou servidores de banco de dados e crie os seguintes bancos de dados externos. Opcionalmente, é possível nomear esses bancos de dados de forma diferente. Anote os nomes e use-os posteriormente quando configurar a extensão.
   - cloud_controller
   - credhub
   - diego
   - Medalhão
   - network_connectivity
   - network_policy
   - routing-api
   - uaa
2. Se estiver usando um banco de dados Postgres, conecte-se a cada banco de dados e instale a extensão `citext` usando a seguinte SQL:

   ```
   create extension citext;
   ```

3. Certifique-se de que a porta de conexão esteja aberta no firewall e que a conexão remota seja permitida para esses bancos de dados.

Clone o repositório `cfp-cf-ext-db-extension`

Aplique o `cfp-cf-ext-db-extension` antes de sua implementação inicial. Se você aplicar essa extensão após a implementação inicial, ela requer reimplementação, o que resulta em perda de dados.

Criando o arquivo .zip de extensão

Clone o repositório Git `https://github.com/ibm-cloud-architecture/cfp-cf-ext-db-extension`. Na raiz do projeto, crie o arquivo .zip da extensão usando o seguinte comando:

```
zip -r ../cfp-cf-ext-db-extension.zip *
```

Registrando a extensão

1. Copie o arquivo .zip para o diretório do instalador do IBM Cloud Private Cloud Foundry da VM de concepção.
2. Execute o seguinte comando para registrar a extensão. Certifique-se de que o caminho do arquivo .zip de extensão esteja correto.

   ```
   ./cm extension -e cfp-cf-ext-db-extension register -p ./cfp-cf-ext-db-extension
   ```

3. Configure a extensão usando a linha de comandos ou usando o Ferramenta de implementação do Cloud Foundry.

Configurando a extensão usando a linha de comandos

1. Crie o arquivo `ext-db-uiconfig.yml` no diretório do instalador do IBM Cloud Private Cloud Foundry usando o seguinte conteúdo como um exemplo. Substitua os exemplos pelos valores reais para sua implementação.

```
YAML
uiconfig:
 bbs_db_host: 9.21.107.54
 bbs_db_name: diego
 bbs_db_password: postgres
 bbs_db_user: postgres
 cc_db_host: 9.21.107.54
 cc_db_name: cloud_controller
 cc_db_password: postgres
 cc_db_user: postgres
 configuration_name: externaldb
 credhub_db_host: 9.21.107.54
 credhub_db_name: credhub
 credhub_db_password: postgres
 credhub_db_user: postgres
 db_port: 5432                # This port number is used to connect to all databases.
 db_type: postgres        # This database type is used for all databases. Valid values are
postgres or mysql.
 locket_db_host: 9.21.107.54
 locket_db_name: locket
 locket_db_password: postgres
 locket_db_user: postgres
 policy_server_db_host: 9.21.107.54
 policy_server_db_name: network_policy
 policy_server_db_password: postgres
 policy_server_db_user: postgres
 routing_api_db_host: 9.21.107.54
 routing_api_db_name: routing-api
 routing_api_db_password: postgres
 routing_api_db_user: postgres
 silk_controller_db_host: 9.21.107.54
 silk_controller_db_name: network_connectivity
 silk_controller_db_password: postgres
 silk_controller_db_user: postgres
 uaa_db_host: 9.21.107.54
 uaa_db_name: uaa
 uaa_db_password: postgres
 uaa_db_user: postgres
```

2. No diretório do instalador do IBM Cloud Private Cloud Foundry, execute o seguinte comando para salvar a configuração de extensão:

```
./cm extension -e cfp-cf-ext-db-extension save -c ext-db-uiconfig.yml
```

3. Insira a extensão na implementação principal:

```
./cm states insert -i cfp-cf-ext-db-extension
```

Configurando a extensão usando o Ferramenta de implementação do Cloud Foundry

1. Efetue login no Ferramenta de implementação do Cloud Foundry.
2. No menu principal, clique em `1. Registrar novas extensões`.
3. Localize `cfp-cf-ext-db-extension` e clique em `+` para inserir a extensão na implementação principal.
4. No menu principal, clique em `2. Configuração`.
5. Escolha a configuração do `Banco de dados externo` para `cfp-cf-ext-db-extension` e clique no lápis para editar a configuração.
6. Forneça todos os valores necessários em cada guia do banco de dados.
7. Salve e saia.

Agora a extensão faz parte das principais etapas de implementação para implementar o IBM Cloud Private Cloud Foundry com bancos de dados externos.

Nota: se tiver problemas ao carregar mudanças na extensão com o mesmo nome, cancele o registro da extensão usando o seguinte comando e, em seguida, registre a extensão novamente:

```
./cm extension -e cfp-cf-ext-db-extension unregister
```

Configurando a rede do contêiner

A Rede de contêiner para contêiner, introduzida no IBM Cloud Private 3.2.0, reutiliza uma rede de sobreposição para gerenciar a comunicação entre as instâncias de app, sem passar pelo Gorouter. Para obter uma descrição dos recursos de rede do contêiner e mais detalhes, consulte Entendendo a Rede contêiner para contêiner.

Antes de iniciar

Assegure-se de que IBM® Cloud Private Cloud Foundry está instalado.

Nota: a rede de contêiner é ativada por padrão.

Configuração

O recurso Rede contêiner para contêiner também fornece um endereço IP exclusivo para cada contêiner de app e fornece alcance de endereço IP direto entre as instâncias de app.

A Rede de contêiner para contêiner de sua implementação requer que você crie políticas para comunicação entre as instâncias de app usando a linha de comandos `cf`. Por padrão, os contêineres de aplicativo não são visíveis entre si. As políticas que você cria especificam um app de origem, o aplicativo de destino, o protocolo e a porta para que as instâncias de app possam se comunicar diretamente sem passar pelo Gorouter, por um balanceador de carga ou por um firewall. A Rede de contêiner para contêiner suporta UDP e TCP e é possível configurar políticas para múltiplas portas. Essas políticas aplicam-se imediatamente sem reiniciar o aplicativo.

Configuração Opcional

Os parâmetros `optional` a seguir podem ser ajustados para corresponder a seus requisitos de implementação. Edite o arquivo `your-uiconfig.yml` para ajustar os valores a seguir para customizar a Rede contêiner para contêiner em sua implementação:

1. `cf_neworking.disable`: quando configurado como `true`, desativa a rede de contêiner completamente. O valor padrão é `false`.
2. `cf_networking.network_cidr`: insira um intervalo de endereços IP para a rede de sobreposição. O CIDR deve especificar um intervalo de RFC 1918. Se você não configurar um intervalo customizado, a implementação usará `10.255.0.0/16`.
3. `iptables_logging`: configure o sinalizador como `true` para ativar a criação de log da tabela de endereço IP do kernel. O valor padrão é `false`.
4. Altere o estado da implementação. Execute o comando a seguir para mudar o status para READY:

   ```
   ./cm state -s prepare-cf set --status READY
   ./cm state -s deploy-cf set --status READY
   ```

5. Reative a implementação executando o comando a seguir:

   ```
   ./launch_deployment.sh -c your-uiconfig.yml
   ```

Configurando backups para IBM® Cloud Private Cloud Foundry

Os backups são enviados para um servidor NFS (Network File System) hospedado pelo cliente ou os backups permanecem no `nfs_WAL_server`, dependendo da configuração. É possível gerenciar os backups que são hospedados em NFS em termos de espaço disponível. Também é possível criptografar os backups se isso faz parte de suas regras de segurança.

Sobre backups

Os backups do UAADB, CCDB, Director e as implementações são planejados. Os backups são tomados e postados localmente no `nfs_WAL_server`. A configuração pode especificar se os backups precisam ser movidos para um servidor NFS hospedado pelo cliente.

Veja Componente: servidor User Account and Authentication (UAA) e Componente: Cloud Controller para obter mais informações. Também será possível fazer a sua própria implementação se você implementar os scripts que são descritos em Backup e restauração do BOSH.

UAADB/CCDB backups

Os backups são tomados a cada hora e mantidos por 6 horas no `nfs_WAL_server`. Configure o atributo `db_nfs_copy` no arquivo `uiconfig` para copiar backups em um servidor NFS hospedado pelo cliente. Consulte Instalando o IBM Cloud Private Cloud Foundry para obter detalhes. O processo de cópia é executado a cada hora e os backups de 7 dias são mantidos no NFS hospedado pelo cliente.

Director backups

Configure o atributo `bbr_backup` no arquivo `uiconfig` para configurar o planejamento de backup. Consulte Instalando o IBM Cloud Private Cloud Foundry para obter detalhes. O atributo `bbr_backup` possui valores padrão. Eles podem ser movidos para um servidor NFS hospedado pelo cliente se fornecidos no atributo `bbr_backup`. Após o atributo ser transferido para o servidor NFS hospedado pelo cliente, ele é removido do `nfs_WAL_server`.

Se nenhum servidor NFS hospedado pelo cliente for fornecido, ou se o servidor NFS hospedado pelo cliente não estiver disponível, vários backups, que são definidos no atributo `bbr_backup`, serão mantidos no `nfs_WAL_server`.

É possível verificar se o backup está funcionando verificando o diretório `/var/vcap/store/bbr_backup/director` ou seu servidor NFS se você configurar um para o backup de diretor.

Restaure seu procedimento:

1. Assegure-se de que você tenha o arquivo `.tgz` de backup. Se o backup está no `nfs_WAL_server`, transfira-o para o contêiner de concepção.
2. Exclua a máquina virtual diretor.
3. Renomeie o `/data/gen-vmware_micro_boshinit-state.json` arquivo.
4. Reinstale o diretor. Execute o comando a seguir:

```
./cm engine reset
./cm states set-status-by-range -f deploy-director --status SKIP #This set all states after
deploy-director to SKIP
launch_deployment.sh...
```

5. Ative a restauração, da concepção ou do `nfs_wal_server`. A ativação da concepção será útil se seu backup não estiver mais no nfs_WAL_server.

    - Para ativar a restauração do `nfs_wal_server`, conclua o procedimento a seguir:

        1. Efetue login no `nfs_wal_server` remotamente com o seguinte comando: `ssh vcap@<nfs_WAL_ip>` ou `bosh -e IBMCloudPrivate -d Bluemix ssh <nfs_WAL_server_name>`. Execute `bosh -e IBMCloudPrivate vms` para localizar o `<nfs_WAL_ip>` e o `<nfs_WAL_server_name>`.
        2. Execute `tar xvf <file>` para descompactar o backup a partir do diretório nfs `/local_backup/director` ou do disco local `/var/vcap/store/cfp-backup/director`.
        3. Para restaurar, execute o comando a seguir: `/var/vcap/packages/bbr-1.2.0/bbr director --host 192.168.247.2 --username director_backup --private-key-path /var/vcap/sys/run/cfp-backup/director_backup.key restore --artifact-path <uncompressed file>`.

    - Para ativar a restauração da concepção, conclua o procedimento a seguir:

        1. Copie seu backup para a concepção usando, por exemplo, `docker cp <backup_file> cfp-inception-<environment_name>:/tmp`
        2. Conecte-se à concepção com o seguinte comando: `./connect.sh -n <your environment name>`.
        3. Execute o comando a seguir para instalar o bbr 1.2.0:

```
wget https://github.com/cloudfoundry-incubator/bosh-backup-and-
restore/releases/download/v1.2.0/bbr-1.2.0.tar
tar xvf bbr-1.2.0.tar
chmod +x releases/bbr
```

        4. Execute o comando a seguir para recuperar a chave privada:

```
/data/CloudFoundry/certificates.yml gato
```

        5. Procure o `director_backup` grupo de atributos.
        6. Copie em um arquivo a chave->conteúdo de dados Assegure-se de que cada linha seja iniciada na coluna um.

7. Execute `tar xvf <file>` para descompactar o backup a partir do diretório onde foi copiado o backup.
8. Para restaurar, execute o comando a seguir:

```
<BBR_INSTALLATION_PATH>/bbr director --host 192.168.247.2 --username director_backup
--private-key-path <BACKUP_KEY_PATH>/director_backup.key restore --artifact-path
<uncompressed file>.
```

Backups de implementação

Implementação do Bluemix

Como o backup de diretor, o backup do Bluemix (planejamento, servidor NFS hospedado pelo cliente e número de backup para manter) pode ser configurado no atributo `bbr_backup`. Somente o manifest do Bluemix é submetido a backup.

É possível verificar se o backup está funcionando verificando o diretório `/var/vcap/store/bbr_backup/deployments/Bluemix` ou seu servidor NFS, se você configurar um servidor NFS para o backup de diretor.

A implementação customizada

Se sua implementação implementar o Backup e restauração do BOSH, será possível configurar o atributo `cf_custom` para fazer backup de forma automática e periódica de sua implementação.

```
  cf_custom: |+
    properties:
      bbr_backup:
        deployment_backup:
          deployments:
          - (( grab bmxconfig.bbr_backup.deployments_backup.deployments.0 )) #This to continue to
backup the Bluemix
          - name: YourDeployment
            schedule: "* 3 * * *"
            nb_backups: 15
            nb_logs: 15
            max_log_size: 10000
```

É possível verificar se o backup está funcionando verificando o diretório `/var/vcap/store/bbr_backup/deployments/<YourDeploymentName>` ou seu servidor NFS, se você configurou um para o backup de diretor.

Configurando Zonas de Disponibilidade para IBM Cloud Private Cloud Foundry

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

As Zonas de Disponibilidade são segmentos de infraestrutura de rede designados pelo operador e funcionalmente independentes. As Zonas de disponibilidade permitem tolerância a falhas e alta disponibilidade de uma implementação do IBM® Cloud Private Cloud Foundry.

Para obter mais informações sobre Zonas de Disponibilidade e sua implementação, consulte Alta disponibilidade no Cloud Foundry e Zonas de Disponibilidade.

Antes de iniciar

Para ambientes de produção (ambientes nos quais o `uiconfig.developer_mode` não é especificado ou está configurado como `false`), quatro zonas de disponibilidade são ativadas e implementadas por padrão. Todas essas zonas usam as mesmas propriedades de nuvem que são obtidas de seu arquivo de configuração `uiconfig.yml`. Em outras palavras, ao usar o VMware, todas as zonas de disponibilidade são configuradas para serem implementadas nos mesmos data center, cluster e conjunto de recursos. Ao usar o OpenStack ou o AWS, todas as zonas de disponibilidade são configuradas para serem implementadas na mesma zona de disponibilidade do OpenStack ou do AWS.

Se desejar customizar as zonas de disponibilidade, é melhor fazer isso antes da primeira implementação do Cloud Foundry. Caso contrário, se suas mudanças requererem que as MVs sejam reimplementadas, poderá ser necessário disponibilizar endereços IP adicionais e mudar as designações de IP estático para concluir a configuração de suas Zonas de Disponibilidade.

Por padrão, uma implementação tenta distribuir de maneira uniforme o número de instâncias especificadas para cada grupo de instâncias em todas as Zonas de Disponibilidade de maneira round-robin.

Customizando sua configuração

1. Se você ainda não implementou o Cloud Foundry, deve-se primeiramente gerar a configuração de base. Execute o comando a seguir ou use a IU de implementação para configurar todos os estados após **Gerar configuração** como o status **SKIP**.

   ```
   ./cm states set-status-by-range --from-state gen-bmxconfig --status SKIP
   ```

   Ative a implementação.

   ```
   ./launch_deployment.sh -c uiconfig.yml
   ```

   Após a implementação ser concluída, continue com as instruções.

2. Em seu diretório de configuração de instalação, localize o arquivo `CloudFoundry/bmxconfig.yml` e abra-o para edição. Localize o campo `bmxconfig.default_azs_definition`. Modifique `cloud_properties` em cada definição de zona de disponibilidade para seu ambiente. É recomendado manter o número padrão de zonas de disponibilidade e seus nomes padrão editando apenas os valores de cada campo `cloud_properties`.

   Na maioria dos casos, também é necessário incluir sub-redes para uso com suas Zonas de Disponibilidade. Localize a seção `bmxconfig.networks`. Usando a entrada de `sub-rede` existente que é gerada a partir de sua configuração principal como um guia, inclua sub-redes adicionais, conforme necessário, para suas zonas de disponibilidade. Certifique-se de que cada zona de disponibilidade esteja especificada na sub-rede correspondente correta.

   Um exemplo de uma definição de Zona de Disponibilidade é mostrado para cada IaaS suportado. Customize os valores de amostra exibidos em maiúscula na seção `cloud_properties`.

   VMware

   ```
   bmxconfig:
     default_azs_definition:
       azs:
       - name: z1
         cloud_properties:
           datacenters:
           - name: DATACENTER_NAME_1
             clusters: [{CLUSTER_NAME_1: {resource_pool: RESOURCE_POOL_NAME_1}}]
   ```

   Nota: para VMware, as Zonas de Disponibilidade padrão em `bmxconfig.yml` contêm um erro de sintaxe. A chave `clusters` aparece em seu próprio item da lista em vez de como um irmão para a chave `name`. Substitua o caractere – que precede os `clusters:` com um espaço. Ao verificar os conteúdos de `CloudFoundry/cloud-config.yml` na próxima etapa, esteja ciente de que a ordem das chaves pode mudar, mas somente a primeira chave em `datacenters` será precedida por um caractere –.

   OpenStack

   ```
   bmxconfig:
     default_azs_definition:
       azs:
       - name: z1
         cloud_properties:
           availability_zone: OPENSTACK_AVAILABILITY_ZONE_1
   ```

   AWS

   ```
   bmxconfig:
     default_azs_definition:
       azs:
       - name: z1
         cloud_properties:
           availability_zone: AWS_AVAILABILITY_ZONE_1
   ```

   Se você mudou o nome de qualquer zona de disponibilidade ou mudou o número de zonas de disponibilidade, certifique-se de substituir todas as referências às zonas originais no arquivo `bmxconfig.yml` por referências às novas zonas.

3. Execute o comando a seguir ou use a IU de implementação para configurar o estado **Gerar configuração** como o status **READY**.

```
./cm state -s gen-bmxconfig set --status READY
```

Ative a implementação novamente. Os valores modificados em `CloudFoundry/bmxconfig.yml` são mesclados em `CloudFoundry/cloud-config.yml`. Visualize este arquivo para confirmar se as mudanças estão corretas. Se você mudou o nome das zonas de disponibilidade, precisará atualizar o valor para `compilation.az` para referir-se a uma de suas zonas de disponibilidade.

4. Configure o estado **Gerar configuração** para o status **SKIP** usando a UI de implementação ou o seguinte comando.

```
./cm state -s gen-bmxconfig set --status SKIP
```

Nota: Se for necessário fazer outras mudanças na configuração para uma reimplementação futura, as mudanças em `CloudFoundry/bmxconfig.yml` e `CloudFoundry/cloud-config.yml` serão perdidas quando esse estado for executado. Deve-se executar apenas esse estado, restabelecer suas modificações e, em seguida, configurar esse estado como **SKIP** e executar o restante da implementação.

5. Se você mudou o nome de alguma zona de disponibilidade ou mudou o número de zonas de disponibilidade, ou se desejar mudar a distribuição de instâncias e a localização de sua zona de disponibilidade, também deve seguir as instruções nos **Exemplos de zona de disponibilidade e de instância** para atualizar as referências da zona no arquivo `cf-deploy.yml`. Pule para a seção **Exemplos de Zona de Disponibilidade e de Instância** agora.

6. Por último, se você não mudou nomes e o número de zonas de disponibilidade, configure os estados restantes após **Gerar configuração** para **READY** usando a UI de implementação ou o seguinte comando.

```
./cm states set-status-by-range --from-state gen-bmxconfig --status READY
```

Reative a implementação para aplicar suas mudanças:

Zona de Disponibilidade e Exemplos de Instância

1. Para mudar a distribuição das instâncias e seus posicionamentos na Zona de Disponibilidade, mude o arquivo `CloudFoundry/cf-deploy.yml` no diretório de configuração de instalação de sua implementação.

Nota: a mudança da distribuição e do posicionamento na Zona de Disponibilidade não é geralmente necessária, já que o número de instâncias é balanceado para uma implementação de produção típica.

2. Para se preparar, deve-se primeiro executar a implementação no estado em que o arquivo `cf-deploy.yml` é gerado e ignorar o estado **Gerar configuração**. Execute os seguintes comandos ou use a UI de implementação para assegurar que **Preparar Cloud Foundry** tenha o status **READY** e para configurar todos os estados após **Preparar Cloud Foundry** para o status **SKIP**.

```
./cm state -s gen-bmxconfig set --status SKIP
./cm states set-status-by-range --from-state gen-bmxconfig --status READY
./cm states set-status-by-range --from-state prepare-cf --status SKIP
```

Ative a implementação e espere sua conclusão antes de editar o arquivo `cf-deploy.yml`.

3. Cada grupo de instâncias aceita uma lista de zonas de disponibilidade (`azs:`) que definem uma ou mais zonas nas quais uma instância é colocada no arquivo `cf-deploy.yml`. Se você mudou o nome das Zonas de Disponibilidade padrão, assegure-se de que todas as referências de zona nesse arquivo usem seus novos nomes.

Consulte o procedimento a seguir e exemplos:

- É possível isolar instâncias para uma Zona de Disponibilidade, que é `z2` no exemplo a seguir.

```
...
instance_groups:
- name: consul
  instances: 3
  azs: [z2]
  migrated_from:
  - { name: consul }
...
```

- É possível fazer com que o BOSH implemente uma instância em todas as Zonas de Disponibilidade, por exemplo, `z1`, `z2` e `z3`.

```
...
- name: consul
  instances: 3
```

```
  azs: [z1, z2, z3]
  migrated_from:
  - { name: consul }
...
```

- As instâncias são colocadas em Zonas de Disponibilidade em um estilo round-robin. Se houver mais zonas especificadas do que instâncias, nem todas as zonas serão usadas. Os comandos a seguir implementam uma instância em `z1` e uma instância em `z2`. Uma instância não é implementada na zona de disponibilidade `z3`.

```
...
- name: consul
  instances: 2
  azs: [z1, z2, z3]
  migrated_from:
  - { name: consul }
...
```

4. Em seguida, configure os estados de implementação concluídos para **SKIP** e os estados restantes até e incluindo **Preparar interface com o usuário** para **READY** usando os seguintes comandos ou a UI de implementação.

```
./cm states set-status-by-range --to-state deploy-cf --status SKIP
./cm states set-status-by-range --from-state prepare-cf --to-state deploy-cfp-ui --status READY
```

   Ative a implementação e aguarde até que ela seja concluída.

5. Edite o `CloudFoundry/cfp-ui-deploy.yml` para mudar a referência de Zona de Disponibilidade para uma de suas novas Zonas de Disponibilidade.

6. Por último, configure os estados de implementação concluídos para **SKIP** e reative os estados de implementação restantes executando os seguintes comandos ou usando a UI de implementação para configurar todos os estados após **Preparar interface com o usuário** para o status **READY**.

```
./cm states set-status-by-range --to-state deploy-cfp-ui --status SKIP
./cm states set-status-by-range --from-state prepare-cfp-ui --status READY
```

   Ative a implementação.

   Nota: Os estados de implementação **Preparar Cloud Foundry** ou `prepare-cf` e os estados **Preparar interface com o usuário** ou `prepare-cfp-ui` geram novamente os arquivos `cf-deploy.yml` e `cfp-ui-deploy.yml` sempre que eles são executados. Deve-se repetir esse procedimento para configurar as Zonas de Disponibilidade corretamente cada vez que precisar ativar a implementação do Cloud Foundry.

Configurando Segmentos de Isolamento em IBM Cloud Private Cloud Foundry Implementações

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

Em geral, os negócios precisam isolar aplicativos que são implementados no IBM Cloud Private Cloud Foundry.

Este requisito vai além do fornecimento de outra implementação do IBM Cloud Private Cloud Foundry para cada segmento necessário. Os requisitos regulamentares do governo estipulam que os recursos de cálculo para um aplicativo sejam isolados de outros aplicativos. A rede que transita os dados que são consumidos e produzidos por esses aplicativos deve também ser isolada de outros dados e aplicativos.

Os segmentos de isolamento podem ser usados para separar aplicativos como se estivessem em implementações do IBM Cloud Private Cloud Foundry diferentes sem a necessidade de complexidade redundante de gerenciamento e de rede.

Preparando o BOSH para suportar segmentos de isolamento

Para ativar segmentos de isolamento para uma implementação, deve-se designar as propriedades `placement_tags` para instâncias em suas implementações. Cada célula do Diego que você deseja incluir em um segmento de isolamento deve incluir um `placement_tags` com o nome do segmento de isolamento. Os nomes de segmentos de isolamento não fazem distinção entre maiúsculas e minúsculas e devem ser exclusivos. Por exemplo, em seu manifesto `./data/CloudFoundry/diego-deploy.yml`, é possível marcar uma célula do Diego como `segment_1`:

```
- instances: 1
  name: cell
```

```
networks:
- name: default
properties:
  diego:
   rep:
     [...]
      placement_tags:
      - segment_1
```

Implementando um novo manifesto

Execute os comandos a seguir para efetuar login no contêiner de concepção e implementar um novo manifesto:

```
$ docker exec -it <inception_container_name> bash $ bosh deploy /data/CloudFoundry/diego-deply.yml
```

Gerenciando segmentos de isolamento com a interface da linha de comandos do

Cloud Foundry

Criando segmentos de isolamento

Use o comando `$cf create-isolation-segment segment_1` da CLI do Cloud Foundry para criar segmentos de isolamento no IBM Cloud Private Cloud Foundry. O comando retorna o resultado a seguir:

```
$ cf create-isolation-segment segment_1
Creating isolation segment segment_1 as johndoe...
OK
```

Nota: o nome do segmento de isolamento que é usado no comando da CLI do Cloud Foundry deve corresponder ao valor que é especificado na seção `placement_tags` do arquivo de manifesto do Diego. Se os nomes não corresponderem, o Cloud Foundry falhará ao colocar aplicativos nos segmentos de isolamento quando os aplicativos são iniciados ou reiniciados nos espaços que são designados aos segmentos de isolamento.

Visualizando informações sobre segmentos de isolamento

Os administradores e usuários podem usar os comandos `cf` que são fornecidos pela CLI do Cloud Foundry para recuperar as configurações de segmento de isolamento. Como um administrador, é possível ver todos os segmentos de isolamento que estão registrados nas implementações atuais do Cloud Foundry. Os usuários podem visualizar apenas os segmentos de isolamento que são designados a suas organizações.

Listar segmentos de isolamento

Execute o comando `$cf isolation-segments` para visualizar os segmentos de isolamento disponíveis para sua função. O comando retorna o resultado a seguir:

```
$ cf isolation-segments
Getting isolation segments as johndoe...
OK

name           orgs
segment_1     org1
```

Visualizando segmentos de isolamento que são ativados para uma organização

Como um administrador, é possível designar múltiplos segmentos de isolamento para uma organização.

Execute o comando `cf org <org_name>` para visualizar segmentos de isolamento que são designados ao `<org_name>`. O comando retorna o resultado a seguir:

```
$ cf org org1
Getting info for org org1 as johndoe@example.com...

name:                org1
domains:             example.com, apps.example.com
quota:               paid
```

```
spaces:               development, production, staging
isolation segments:   segment_1
```

Visualizando um segmento de isolamento que é designado a um espaço

Apenas um segmento de isolamento pode ser designado a um espaço.

Execute o comando `cf space <space_name>` para visualizar segmentos de isolamento que são designados ao `<space_name>`. O comando retorna o resultado a seguir:

```
$cf space staging

name:                staging
org:                 org1
apps:
services:
isolation segment:   segment_1
space quota:
security groups:
```

Excluindo segmentos de isolamento

Somente administradores podem excluir segmentos de isolamento. ***Nota:*** não é possível excluir segmentos de isolamento que contenham aplicativos implementados.

Execute o comando `cf delete-isolation-segment <isolation_segment_name>` para excluir o segmento de isolamento. O comando retorna o resultado a seguir:

```
$ cf delete-isolation-segment segment_1
Deleting isolation segment my_segment as admin...
OK
```

Aumentando o número de células Diego

Depois de implementar aplicativos, pode ser necessário aumentar o número de células Diego em seu ambiente.

Nota: no modo de desenvolvedor, é possível usar apenas uma célula Diego.

Antes de iniciar

Assegure-se de que o IBM® Cloud Private Cloud Foundry esteja implementado.

Etapas

1. Edite o arquivo `your-uiconfig.yml`. Aumente o valor para o parâmetro `diego_cell_instances` e salve as mudanças.
2. Altere o estado da implementação. Execute o comando a seguir para mudar o status para `READY`:

   ```
   ./cm state -s prepare-cf set --status READY
   ./cm state -s deploy-cf set --status READY
   ```

3. Reative a implementação com o comando a seguir:

   ```
   ./launch_deployment.sh -c your-uiconfig.yml
   ```

Nota: a implementação das células adicionais pode levar alguns minutos.

Configurar certificados confiáveis para aplicativos para IBM Cloud Private Cloud Foundry

Configure certificados confiáveis para instâncias do aplicativo no IBM® Cloud Private Cloud Foundry e no Cloud Foundry Enterprise Environment.

É possível, opcionalmente, implementar um conjunto de certificados confiáveis do sistema codificados pelo PEM que você pode disponibilizar para instâncias do aplicativo. Esses certificados são instalados em:

- O armazenamento confiável `/etc/ssl/certs/` para aplicativos baseados em buildpack que usam a pilha `cflinuxfs2`. Esses certificados estão disponíveis automaticamente para programas que respeitam o armazenamento confiável, como `openssl`.
- `Os arquivos /etc/cf-system-certificates` para aplicativos baseados em imagens do Docker e Windows™ que usam a extensão do arquivo `.crt`. O local dos certificados é fornecido na variável de ambiente `CF_SYSTEM_CERT_PATH` no contêiner de instância.

- Para o IBM Cloud Private Cloud Foundry:

  Os certificados devem ser colocados no arquivo `certificates.yml` que está no diretório `./CloudFoundry/certificates.yml` no caminho especificado durante a execução do comando `claunch.sh -c <directory>`. Coloque o conteúdo dos três arquivos de certificado nesta sub-rotina:

  ```
  ```
  trusted_certs:
  certificate:
    data: |
      -----BEGIN CERTIFICATE-----
      (contents of certificate #1)
        -----END CERTIFICATE-----
      -----BEGIN CERTIFICATE-----
      (contents of certificate #2)
        -----END CERTIFICATE-----
    description: "Trusted certificates to be injected into Diego containers"
  ```
  ```

- Para o Cloud Foundry Enterprise Environment:

  O segredo deve estar no seguinte formato:

  ```
  kind: Secret
  ...
  data:
    trusted_certs:
        -----BEGIN CERTIFICATE-----
        (contents of certificate #1)
        -----END CERTIFICATE-----
          -----BEGIN CERTIFICATE-----
          (contents of certificate #2)
  ```

  O objeto Secret deve estar no mesmo `namespace` no qual o gráfico do Helm do Cloud Foundry Enterprise Environment está implementado. Veja aqui um exemplo de como criar um segredo do Kubernetes a partir de um arquivo que contém certificados confiáveis:

  ```
  kubectl create secret generic trust-secret --from-file=trusted_certs=<path-to-file-containing-trusted-certs>
  ```

Usando extensões no IBM Cloud Private Cloud Foundry

No IBM® Cloud Private Cloud Foundry, uma extensão é um pacote que contém um processo de implementação e seus scripts e arquivos necessários. É possível usar a estrutura de extensão para integrar suas implementações com a implementação do Cloud Foundry ou executá-las de forma independente.

É possível dividir o processo de implementação em várias etapas. O gerenciador de configuração executa as etapas, ou estados, sequencialmente, de acordo com a ordem que você configurou no arquivo de manifesto ou que você configurou nas dependências entre os estados. No arquivo manifest, você fornece nomes de etapa, o nome de um script para executar em cada etapa e quaisquer parâmetros necessários e seus valores. Ao executar uma implementação, as etapas são executadas seguindo uma ordem de `classificação de topologia`, até que uma execução de script falhe ou todas as etapas sejam concluídas. Se a execução do script que está associado à etapa falhar, essa etapa será marcada como `FAILED` e todas as etapas bem-sucedidas serão marcadas com `SUCCEEDED`.

Criando uma extensão

Você cria extensões gravando scripts para cada etapa de implementação, criando um arquivo manifest que descreve a ordem de execução da etapa e quaisquer parâmetros e compactando os arquivos manifest e da etapa.

1. Crie os scripts para executar suas etapas de implementação. Deve-se estar apto para executar esses scripts em um computador que use Ubuntu.

2. Crie o arquivo manifest. O arquivo manifest deve estar localizado no diretório-raiz de sua extensão como `extension-manifest.yml` e deve conter uma seção `states` que fornece a sequência de etapas e o nome do script para cada etapa. É possível encontrar um exemplo do `extension-manifest.yml` no `<your_data_directory>/extensions/embedded/cfp-bosh-templates`. A seção `states` é extraída durante o registro da extensão e armazenada como `states-file.yml` no diretório de extensão. Se você precisar fornecer outros parâmetros, será possível criar mais seções para contê-los.

   Nota: Deve-se especificar o atributo `next_states` para cada estado e, em seguida, uma `classificação de topologia` é executada para calcular a sequência de execução. Ou, se você não especificar o atributo `next_states`, a ordem de execução seguirá a sequência do estado que está listado na seção `states` do `extension-manifest.yml`.

   A seção `states` assume o formato a seguir:

   ```
   states:
   - name: task1
     log_path: /tmp/task.log
     status: READY
     start_time:
     end_time:
     reason:
     script: scripts/success.sh task1
     script_timeout: 10
     next_states: [ "task2", "task3" ]
   ...
   ```

   Consulte a tabela de parâmetros de estado .

   O `extension-manifest.yml` contém parâmetros globais.

   - O parâmetro global `states_update_mode` descreve o que deve ser feito quando uma extensão é registrada novamente. Consulte a seção de registro. Ele pode assumir os valores `merge`, `replace` ou `new`. O `merge` é o valor padrão.

   - O parâmetro global `validation_config_url` permite que você chame um serviço da web para validar sua configuração. Ele contém a URL desse serviço da web. A configuração é postada para esse serviço da web. O serviço da web deve retornar uma resposta para cada atributo da configuração. A resposta inclui um atributo `value` que está configurado para o valor atual do atributo e um , que inclui o valor `warning` ou `error`.

     Exemplo de resposta:

     ```
     uiconfig:
     bluemix_apps_domain:
       value: local.mybluemix.net
     ...
     cluster_name:
       message: 'ESX: 10.1.1.10 is not accessible using port 443. dial tcp 10.1.1.10:443:
     i/o timeout'
       message_type: warning
       value: MY_CLUSTER
     ...
     ```

   - O parâmetro global `generate_config_url` permite que você chame um serviço da web para transformar a configuração salva, se necessário.

   - O parâmetro global `persisted_paths` permite listar os diretórios de arquivos que devem ser mantidos entre as diferentes liberações da extensão. Os caminhos estão relacionados ao diretório-raiz da extensão. O `state-files.yml` e os arquivos de configuração são sempre mantidos por meio de liberações.

     Opcionalmente, é possível incluir uma seção `call_state`, que insere uma etapa no arquivo `states` pai de onde a extensão deve ser chamada. Se você incluir um `call_state`, não será possível criar um arquivo `state.yml` a ser usado para incluir a etapa de chamada no arquivo `states` pai.

     A seção `call_states` assume o formato a seguir:

```
call_state:
phase: AtEachRun
previous_states: [ <previous_states> ]
 next_states: [ <next states> ]
```

A seção `call_states` usa os mesmos valores padrão que a seção `states`. Especificar `previous_states` e `next_states` facilita o processo de inserção de extensão. Não é preciso fornecer informações no comando `./cm states insert -i <extension_name>  -n <insertion_state_position_reference> -b` ou ao registrar a extensão usando a interface com o usuário. Sufixos de `./cm states insert -i <extension_name>` para inserir a extensão no local correto no arquivo `states`.

A seção `ui_metadata` assume o formato a seguir:

```
ui_metadata:
 production:
   label: "Production environment"
   groups:
   - name: "network"
     title: "Network"
     properties:
     - name: "console_ip"
       label: "Console IP"
       description: "The IP address of the console"
       type: "text"
       validation_regex: "^(?:(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9 ]) \ \.) {3 }
(?: 25 [ 0-5 ] | 2 [ 0-4 ][0-9] | 1 [ 0-9 ][0-9] | [ 1-9 ]? [0-9 ]) $"
       validation_error_message: "The field must be an IP address"
   mandatory: true
   hidden: false
   sample_value: "E.g. 10.10.1.12"
   default: "10.10.1.12"
   - name: "security"
     title: "Security"
     properties:
     - name: "encryption"
       label: "Encryption"
       properties:
       - name: "encryption_type"
         label: "Encryption type"
         type: "text"
       - name: "method"
         label: "Method"
         type: "text"

     ...
 development:
   label: "Development environment"
   groups:
   - name: "Security"
   ...
```

A seção `ui_metadata` é composta de uma lista de `configurations`. No exemplo anterior, os nomes de configuração são `production` e `Development`. The UI includes a drop-down box to select the configuration that you want to use. O `laber` da configuração é usado para preencher a caixa suspensa. Se você omitir o rótulo, o nome da `configuration` será usado. Cada `configuração` contém uma lista de grupos que são definidos em `grupos`. Cada grupo é representado por uma guia na IU e possui um `name`, um `title` e uma lista de `properties`. Cada propriedade possui vários atributos que incluem o atributo `properties`, que permite definir uma hierarquia de propriedades.

Consulte a Tabela de parâmetros de metadados da IU.

O código a seguir é exemplos de propriedade.

- dropdown:

```
- name: "vmware_disk_type"
  label: "VMware disk type"
  description: "VMware default disk type (thin/preallocated)"
  type: "dropdown"
  mandatory: true
  default: thin
  items:
```

```
    - id: thin
      label: thin
    - id: preallocated
      label: preallocated
```

- Caixa de opção:

```
- name: enabled
  label: "Enable backup"
  description: "If checked the backup will run on the provided schedule"
  type: checkbox
  default: true
  sample_value: 'true'
```

- hierarquia:

```
- name: "bbr_backup"
  label: "BBR backup setup"
  description: "BBR backup setup for director/deployment"
  properties:
  - name: customer_nfs_host
    label: "Customer NFS host"
    type: "text"
    default: ''
    sample_value: 'mynsf.mycompany.com'
  - name: customer_nfs_path
    label: "Customer NFS path"
    type: "text"
    default: ''
    sample_value: '/cfbackup'
```

3. Crie uma pasta para sua extensão e copie os scripts e o arquivo manifest para a pasta. Coloque o arquivo manifest na raiz do diretório. É possível colocar os arquivos de script em qualquer subpasta, mas assegure-se de que os valores de parâmetro de script contenham os caminhos de arquivo relativo corretos. A estrutura de pasta é semelhante ao exemplo a seguir:

```
extension-manifest.yml
scripts/
scripts/success.sh
scripts/README.md
```

4. Compacte a pasta de extensão em um arquivo .zip. O archive deve ser compatível com o formato ZIP64. Consulte Especificação de formato de arquivo .ZIP. O comando `zip` no Ubuntu e iOS é compatível com ZIP64.

5. A estrutura de extensão suporta `i18n` para a seção ui_metadata. Os arquivos de tradução devem estar localizados em um diretório `/i18n` da extensão. As extensões de arquivo suportadas são `.yml`, `.yaml` e `.json`. Cada candidato de atributo para tradução deve ser configurado com uma chave referenciando uma entrada nos arquivos `i18n`. Quando ui_metadata é solicitado do gerenciador de configuração, por meio da API, o gerenciador de configuração recebe o idioma do cliente. Se o `ui_metadata.<lang>.yml` ainda não existir no diretório de tempo de execução de extensão, ele será criado e retornado como uma resposta à solicitação. Como a ferramenta de implementação Cloud Foundry usa as APIs do gerenciador de configuração, o ui_metadata é mostrado no idioma do cliente configurado no navegador.

Os seguintes atributos são traduzidos:

- O rótulo de configuração
- O rótulo do grupo
- A descrição do grupo
- O rótulo da propriedade
- A descrição da propriedade
- A propriedade validation_error_message
- A propriedade items.label
- A propriedade sample_value

Parâmetros de estado

Parâmetro	Descrição	Valor	Valor Padrão	Obrigatório

Parâmetro	Descrição	Valor	Valor Padrão	Obrigatório
`name`	O nome do estado.	Sequência alfanumérica	N/D	True
`label`	O rótulo do estado.	Sequência alfanumérica	N/D	False
`script`	O script a ser executado durante o estado. É possível fornecer o caminho relativo da raiz do arquivo .zip de extensão. Se você fornecer um caminho absoluto, ele será um caminho no contêiner de concepção.	Caminho de arquivo	N/D	True
`phase`	Quando o script é executado. Se você configurar esse valor de parâmetro como			

`AtEachRun`, o script será executado, mesmo se o status do estado for `SUCCEEDED`. Se `phase` estiver em branco ou omitido, os estados serão executados somente se o status for `READY` ou `FAILED`.|

- `AtEachRun`
- Em branco

|Em branco|False | | `log_path` | O caminho do log para esse estado. Quando o script para um estado é executado, o log de execução anterior é submetido a backup.|Caminho de arquivo|/data/logs/extensions/custom/extension_name.log|False| | `time_out`| O tempo máximo de execução do script. Se o script não concluir no tempo alocado, ele parará e o status do estado mudará para `FAILED`.|Tempo em minutos|60|False| | `status` | O status do estado. O status do estado muda após a execução do script.|

- READY
- SKIP
- RUNNING
- SUCCEEDED
- FAILED

|N/D|True| | `start_time` |O registro de data e hora de início do script na Hora Universal Coordenada. O gerenciador de configuração muda esse valor.|N/D|N/D|False | | `end_time` | O registro de data e hora final do script na Hora Universal Coordenada. O gerenciador de configuração muda esse valor.|N/D|N/D|False | | `reason`|O motivo de falha do estado. Esse valor é mudado pelo gerenciador de configuração durante a implementação.|N/D|N/D|True | | `protegido`|Se o estado for protegido, o usuário não poderá removê-lo.|N/D|False|False | | `excluído`|Se esse parâmetro for configurado como true, o estado será removido do `states-file.yml` após a mesclagem.|N/D|false|False | | `prerequisite_states` | Se os estados atuais forem READY/FAILED, os estados listados também serão configurados como READY. Cada estado listado deve ser anterior ao estado atual. | matriz de sequência | nenhum | False| | `rerun_on_run_of_states` | Se um estado na lista for READY/FAILED, o estado atual será configurado como READY.| matriz de sequência | nenhum | False| | `states_to_rerun` | Se o estado atual for READY/FAILED, os estados listados também serão configurados como READY. Cada estado listado deve estar após o estado atual. | matriz de sequência | nenhum | False | | `previous_states`|Este é um campo calculado.|N/D|nenhum|False | | `next_states`|Matriz de nomes de estados que devem ser executados após este estado. Use este parâmetro para definir a ordem na qual você deseja executar os estados. O mecanismo de implementação faz uma classificação topológica com base nessa matriz. Se você configurá-lo para um estado, deverá configurá-lo para todos os estados. Se você não especificar esse parâmetro, o próximo estado no arquivo de estados será considerado como o próximo estado a ser executado.|N/D|nenhum|False |

Parâmetros de metadados da UI

Atributos	Descrição	Valor	Valor Padrão	Obrigatório
`name`	O nome do atributo. This name is used in the configuration file. O nome deve seguir o			

mesmo formato de uma tag YAML (tag YAML) e não deve conter um ponto (.) ou um ponto de exclamação (!). |Sequência alfanumérica|N/D|True| | `label`|O rótulo a ser exibido na IU.|N/D|O valor definido pelo nome|False| | `description`|A descrição a ser exibida na IU.|N/D|N/D|False| | `type`|O tipo pode ser `text`, `textarea`,`number`, `checkbox`, `dropdown` e `array`.|N/D|text|False| | `items`|Usado quando o tipo é igual a `dropdown` e contém uma matriz de objetos, por exemplo, ID e rótulo.|N/D|text|False| | `validation_regex`|A expressão regular para validar a entrada.|N/D|N/D|False| | `validation_error_message`|A mensagem a ser exibida se a entrada não corresponder à expressão regular.|N/D|N/D|False| | `obrigatórios`|Define se o campo é obrigatório.|verdadeiro / falso|true|False| | `hidden`|Se true, o campo não será mostrado na UI. Consultar nota.|verdadeiro / falso|false|False| | `sample_value`|O valor a ser exibido como uma sugestão na IU|N/D|N/D|False| | `Padrão`|O valor padrão para o campo.|N/D|N/D|False|

Notas:

- Os atributos `validation_regex` e `mandatory` não são cumpridos e não impedem que você salve a configuração. O designer de extensão deve implementar validação extra nos scripts que estão associados aos estados da extensão.
- Um valor é double-quoted if quoted ou double-quoted na IU. This is useful when you want to set numbers as a string.
- O tipo `array` é suportado apenas para matrizes de campos de texto e matrizes de objetos, conforme ilustrado nos exemplos a seguir.
  - Matriz de campos de texto

    ```
    - name: "default_security_groups"
    label: "Default security groups"
    description: "Security group used for each AMI to control network traffic"
    type: "array"
    ```

  - Matriz de objetos

    ```
    - name: deployments
      label: "Deployments"
      description: "Deployments to backup"
      type: array
      properties:
      - name: name
        label: "Deployment name"
        description: "The deployment name to backup"
        type: text
        sample_value: "My deployment"
    ```

- O parâmetro `hidden` pode ser configurado com literal true, false ou com uma expressão. As expressões são suportadas nos níveis de grupo e de propriedade e têm o formato de: $, ,. O path_to_attribute é o uiconfig. no nível do grupo e no nível da propriedade. Os verbos disponíveis são: $equal e $in.
  - $equal retorna true se o valor de atributo for igual ao valor fornecido, por exemplo, $equal,developer_mode,true
  - $in retorna true se o valor de atributo estiver na lista fornecida de valores, por exemplo, $in,deployment_mode,update

Executando a extensão

Ao executar uma extensão, as regras a seguir serão aplicadas:

- O primeiro script com status `READY` ou `FAILED` é executado.
- A execução da extensão para quando uma execução de script falha.
- Scripts com o status `SKIP` ou `SUCCEEDED` não são executados.
- Se você configurar o valor de parâmetro `phase` como `AtEachRun`, o script para essa etapa será executado cada vez que você iniciar o mecanismo, mesmo se seu status for `SUCCEEDED`.
- Enquanto um script é executado, seu status é configurado para `RUNNING`.

1. Registre a extensão no contêiner de concepção. O registro de uma extensão faz upload do conteúdo do archive de extensão para a pasta `/data/extensions/custom/<extension_name>` no contêiner de concepção e extrai a seção `states` para criar um arquivo `states-file.yml`. Se a extensão já tiver sido registrada, o `states-file.yml` existe. Dependendo do `states_update_mode` (merge|replace|new), o states-file atual de um registro anterior será mesclado, substituído ou o novo `states-file.yml` se chamará `states-file-new.yml`.

   Nota: não é necessário registrar extensões que sejam fornecidas pela IBM. Eles já estão registrados. Execute o seguinte comando para registrar extensões. `<extension_name>` é o nome da extensão e `<archive_path>` é o caminho e o nome da pasta de archives compactados.

   ```
   ./cm extension -e <extension_name> register -p <archive_path>
   ```

2. Confirme se a extensão está registrada. Execute o comando a seguir para verificar se a extensão está registrada e se o nome da extensão está listado na saída:

   ```
   ./cm extensions
   ```

3. Se seus scripts contiverem variáveis, será possível criar um arquivo de configuração para definir as variáveis e fornecê-las para a extensão.

   1. Crie o arquivo de configuração. O arquivo deve estar no formato YAML e iniciar com uma seção `uiconfig`. Por exemplo, para configurar o valor de um parâmetro `servicebroker_port`, o conteúdo do arquivo de configuração

será semelhante ao texto a seguir:

```
uiconfig:
 servicebroker_port: 8080
```

2. Para fornecer o arquivo de configuração para o contêiner de concepção, execute o seguinte comando. `<extension_name>` é o nome da extensão que foi registrada e `<config_file>` é o nome do arquivo de configuração. O arquivo de configuração é armazenado como o arquivo `uiconfig.yml` na pasta `/data/extensions/custom/<extension_name>` no contêiner de concepção. Os scripts podem usar o arquivo de configuração para recuperar cada parâmetro de que eles precisam no tempo de execução.

   ```
   ./cm extension -e <extension_name> save -c <config_file>
   ```

4. Implemente a extensão. É possível integrar a extensão com a implementação principal do IBM Cloud Private Cloud Foundry, incluir a extensão na implementação de outra extensão ou implementar a extensão por si mesma.

   Importante: a ferramenta do gerenciador de configuração, `./cm`, é instalada automaticamente ao implementar o IBM Cloud Private Cloud Foundry, portanto, deve-se executar os comandos `cm` a partir do diretório de instalação.

   - Para inserir a extensão na implementação do IBM Cloud Private Cloud Foundry, conclua as etapas a seguir:

     1. Visualize os estados IBM Cloud Private Cloud Foundry .

        ```
        ./cm states
        ```

     2. Se o manifesto de extensão não contiver uma seção `call_step`, crie um arquivo `state_cf.yml` para a implementação do IBM Cloud Private Cloud Foundry. Este arquivo é o arquivo de modelo que é inserido na implementação do IBM Cloud Private Cloud Foundry como uma nova etapa a ser executada. O parâmetro `script` é automaticamente configurado para chamar a extensão. Os parâmetros são configurados por padrão como descrito na tabela anterior. O arquivo contém o seguinte texto:

        ```
        name: <extension_name>
        ```

        Inclua a extensão no arquivo de estado IBM Cloud Private Cloud Foundry. `<state_name>` é o estado após o qual a extensão é executada. Se você precisar executar a extensão antes de um estado, substitua a opção `-n` pela opção `-b`.

        ```
        ./cm states insert -s state_cf.yml -n <state_name>
        ```

     3. Se o manifest de extensão contiver uma seção `call_step`, será possível incluir a extensão no arquivo de estado do IBM Cloud Private Cloud Foundry, conforme ilustrado no comando a seguir:

        ```
        ./cm states insert -i <extension_name> -n <state_name>
        ```

        O `<extension_name>` é o nome da extensão que você deseja executar e `<state_name>` é o estado após o qual a extensão é executada. Se você precisar executar a extensão antes de um estado, substitua a opção `-n` pela opção `-b`. O `<state_name>` não será solicitado se `previous_states` e `next_states` estiverem definidos no `call_state` da extensão.

     4. Para verificar a ordem de estado atualizada, execute novamente o comando `./cm states -e extension_name`.

     5. Execute o comando `launch_deployment.sh` para implementar o IBM Cloud Private Cloud Foundry e sua extensão.

   - Para inserir a nova extensão em uma extensão existente, conclua as seguintes etapas. `<extension_name>` é a extensão existente.

     1. Visualize os estados da extensão existente.

        ```
        ./cm states -e extension_name
        ```

     2. Crie um arquivo `state_<extension_name>.yml` para a extensão, em que `<extension_name>` é o nome da extensão existente. Esse arquivo é o arquivo de modelo inserido na implementação de extensão como uma nova etapa a ser executada. O parâmetro `script` é automaticamente configurado para chamar a extensão. Os parâmetros são configurados por padrão como descrito na tabela anterior. O arquivo contém o seguinte texto:

        ```
        name: <extension_name>
        ```

     3. Inclua a nova extensão no arquivo de estado de extensão existente:

```
./cm states -e <extension_name> insert -s state_<extension_name>.yml -n <state_name>
```

`<extension_name>` é a extensão existente e `<state_name>` é o estado após o qual a extensão é executada. Se você precisar executar a extensão antes de um estado, substitua a opção `-n` pela opção `-b`.

4. Para verificar a ordem de estado atualizada, execute o comando `./cm states -e extension_name` novamente.
5. Implemente a extensão existente:

   ```
   ./cm extension -e <extension_name> deploy
   ```

   O parâmetro `<extension_name>` é o nome da extensão existente.

- Para implementar a extensão sozinha, execute o comando a seguir:

  ```
  ./cm extension -e <extension_name> deploy
  ```

  O parâmetro `<extension_name>` é o nome da extensão. Esse comando é executado em segundo plano, mas é possível incluir um sinalizador `-w` no comando para aguardar a implementação concluir.

  Durante a implementação, um arquivo `states-file.yml` é criado no diretório de extensão. O mecanismo do gerenciador de configuração atualiza o arquivo. Cada entrada é atualizada durante a implementação com valores padrão, se necessário, um status e registros de data e hora de início e parada.

  - Os registros de data e hora do script estão na Hora Universal Coordenada. O registro de data e hora é configurado no início e no término da execução do script.
  - Os status são configurados automaticamente com base no `exitCode` dos scripts. Se o script não concluir a execução (se o código de saída não for igual a zero), o status do script será marcado como `FAILED`.
  - As saídas `StdOut` e `StdErr` do script são registradas no arquivo de log.
  - O processo para e é marcado como `FAILED` se o valor de tempo limite é excedido.

5. Monitore a extensão: é possível verificar os arquivos de log de uma extensão executando o comando `./cm extension -e extension_name logs`. É possível incluir o sinalizador `-f` para acompanhar os logs. Os arquivos de log diferentes da implementação são exibidos sucessivamente.
6. Opcional: cancele o registro da extensão. O cancelamento de registro da extensão remove os dados de extensão do mecanismo de concepção, mas não remove a implementação de seu cluster. Depois de implementar a extensão, não será possível removê-la, a menos que você crie uma extensão que contenha as etapas para remover a extensão instalada. Para cancelar o registro da extensão, execute o comando a seguir:

   ```
   ./cm extension -e extension_name unregister
   ```

   `<extension_name>` é o nome da extensão.
7. Opcional: se você incluiu a extensão em outra extensão, mas não deseja implementar a nova extensão na próxima vez que você implementar a extensão existente, será possível removê-la. Execute o comando a seguir:

   ```
   ./cm states -e extension_name delete -n state_name
   ```

   `<extension_name>` é a extensão existente e `<state_name>` é o estado após o qual a extensão é executada.

Configuration Manager (CM) - guia de referência rápida

Introdução

O Configuration Manager é uma ferramenta que gerencia a implementação do Cloud Foundry. Ele é composto de um servidor e um cliente. O servidor está em execução no contêiner de concepção. O cliente é instalado em seu diretório de instalação e interage com o servidor. Este é somente um guia de consulta rápida.

Formato geral da linha de comandos

```
./cm [-f <format>] [-u <s_url>] [-t <timeout_sec>] [--cacert <certificate_path>] [--insecure] [--
token <token>] <cmd> [<options>...] [<sub-command>] [<options>...]
```

Insira os comandos a seguir para obter detalhes:

- `./cm -h`
- `./cm ... <cmd> -h`
- `./cm ... <cmd>... <sub-commad> -h`

Os valores padrão para as opções `-f`, `-u`, `-t` e `--cacert` são configurados durante a instalação. Execute `./cm api` para obter mais detalhes.

API

- Uso: gerenciar o acesso à API que está em execução no servidor por meio do cliente
- Mostrar: `./cm api`
- Configurar: `./cm [-f yaml|text|json] [-u <api_url>] [-t <timeout_sec>] [--insecure] [--token <token>] [--cacert <certificate_path>] [-k] [-token <token>] api save`
- Remover: `./cm api remove`

CFP

- Uso: gerenciar versões de componente
- Mostrar: `./cm cfp`

Configuration

- Uso: gerenciar a configuração de implementação
- Mostrar: `./cm bmxconfig`
- Configurar: `./cm bmxconfig save -c <configuraiton_file_path>`
- Incluir certificados para o domínio de gerenciamento no arquivo de configuração: `./cm bmxconfig add-certificates -c <configuraiton_file_path> --mgt --key <key_file_path> --cert <cert_file_path> --rootca <rootcA_file_path>`
- Incluir certificados para o domínio dos aplicativos no arquivo de configuração: `./cm bmxconfig add-certificates -c <configuraiton_file_path> --apps --key <key_file_path> --cert <cert_file_path> --rootca <rootcA_file_path>`
- Validar: `./cm bmxconfig validate`

Mecanismo

- Uso: gerenciar o mecanismo de implementação
- Mostrar status do mecanismo: `./cm engine`
- Reconfigurar: `./cm engine reset`

Extensão

- Uso: gerenciar as extensões da IBM ou do cliente
- Mostrar: `./cm extension -e <extension_name>`
- Registrar: `./cm extension -e <extenstion_name> -p <extension_zip_path> [--force]`
- Pós-configuração: `./cm extension -e <extension_name> save -c <config_file>`
- Implementar: `./cm extension -e <extension_name> deploy [--from-state <state_name>] [to-state <state_name>] --wait`
- Cancelar registro: `./cm extension -e <extenstion_name> unregister`
- Reconfigurar: `./cm extension -e <extenstion_name> reset`
- Logs: `./cm extension -e <extenstion_name> [--state <state_name>] [--follow]`

Extensões

- Uso: exibir as extensões disponíveis
- Mostrar: `./cm extenstions`

Logs

- Uso: exibir logs de implementação ou logs de extensão
- Mostrar: `./cm logs [-e <extension_name>] [--state <state_name>] [--follow]`

Estado

- Uso: gerenciar o estado de implementação
- Mostrar: `./cm state [-e <extension_name>] -s <state_name>`
- Configurar novo status para um estado: `./cm state [-e <extension_name>] -s <state_name> set --status <STATUS>`
- Configurar novo tempo limite para um estado: `./cm state [-e <extension_name>] -s <state_name> set --timeout <timeout_min>`

Estados

- Uso: gerenciar os estados de uma implementação
- Mostrar: `./cm states [-e <extension_name>]`
- Localizar estados com um determinado status: `./cm states [-e <extension_name>] --status <STATUS>`
- Inserir um estado em uma implementação: `./cm states [-e <extension_name>] insert --state-path <state_file_path> --insert-extension-name <extension_name> [--state-name <state_name_reference> | --position <position_to_insert>] [--before]`
- Excluir um estado de uma implementação: `./cm states [-e <extension_name>] delete [--state-name <state_name> | --position <position_to_delete]`
- Configurar status por intervalo: `./cm states [-e <extension_name>] set-status-by-range --status <new_status> --from-state <from_state_name> [--from-included] --to-state <to_state_name> [--to-included]`

Status

- Uso: exibir o status do servidor
- Mostrar: `./cm status`

Token

- Uso: criar um novo token no servidor para comunicação de API
- Criar: `./cm token create -c <token_path>`

Criando log e monitorando

Revise e configure os logs do aplicativo e do sistema.

- Configurando o encaminhamento de logs do sistema de plataforma
- Configurando o encaminhamento de log do aplicativo
- Gerenciando a criação de log de eventos de segurança para o IBM Cloud Private Cloud Foundry
- Integrando syslogs do IBM Cloud Private Cloud Foundry com Splunk
- Configurar o Splunk Firehose Nozzle Release como um aplicativo Cloud Foundry
- Conectando o IBM Cloud Private Cloud Foundry ao Prometheus
- Conectando-se ao Elastic Stack no IBM Cloud Private

Configurando o encaminhamento de log do sistema de plataforma

Uma extensão integrada incluída com o IBM® Cloud Private Cloud Foundry permite configurar o encaminhamento de eventos syslog locais no formato RFC5424 de sua plataforma IBM Cloud Private Cloud Foundry para um terminal syslog remoto.

A extensão `cfp-ext-syslog-forwarder` ativa essa funcionalidade e suporta uma série de diferentes opções de configuração, incluindo comunicações seguras usando a autenticação baseada em TLS e no certificado mútuo. Se você deseja encaminhar eventos syslog para o ElasticStack integrado no IBM Cloud Private, o gráfico Helm `ibm-cflogging` poderá configurar automaticamente essa extensão durante a instalação do gráfico. Consulte Conectando-se ao Elasticstack no IBM Cloud Private para obter mais informações.

Para enviar eventos do syslog para um terminal syslog remoto de sua opção, ative a extensão `cfp-ext-syslog-forwarder`. É possível ativar extensões usando uma CLI ou uma interface com o usuário. Para usar a CLI, prepare seu arquivo de configuração de acordo com os Valores de configuração. Em seguida, siga as instruções para Executando a extensão, ignorando a etapa de registro, pois esta é uma extensão integrada incluída com o produto.

Se você preferir usar a interface com o usuário, o `cfp-ext-syslog-forwarder` oferece a edição orientada dos valores de configuração. Selecione um tipo de configuração de `Insecure`, `Server TLS` ou `Mutual TLS`. A interface com o usuário exibe os valores de configuração necessários e opcionais para o cenário selecionado. A interface com o usuário fornece descrições, valores de amostra e validação dos valores de configuração. Para obter informações sobre extensões, consulte Gerenciando extensões e Configurações.

Valores de Configuração

Forneça os valores de configuração necessários a seguir para configurar o encaminhamento de syslog.

- **syslog_address**
  Endereço IP ou domínio do servidor para receber syslogs
- **syslog_port**
  A porta na qual o servidor syslog está atendendo

Esses valores de configuração opcionais também estão disponíveis.

- **configuration_name** Nome da configuração para a entrada assistida pela IU dos valores de configuração. Os valores válidos são `insecure`, `server_tls` ou `mutual_tls`.
- **syslog_transport**
  - Padrão *: `tcp`
    Transporte para encaminhamento de syslog. Os valores válidos são `tcp`, `udp` ou `relp`.
- **syslog_fallback_servers:**
  Uma lista de servidores de fallback a serem usados em que o servidor syslog principal deve ficar indisponível. Cada item de lista possui três chaves (**address**, **port** e **transport**), que definem o servidor de fallback. Isso é suportado apenas quando o transporte para servidores principal e de fallback é especificado como `tcp` ou `relp`.
- **syslog_custom_rule**
  As regras customizadas para rsyslog são gravadas em RainerScript. Por exemplo:
  `if ($msg contains "DEBUG") then stop`
- **syslog_tls_enabled**
  - Padrão *: `false`*
    *Encaminha syslogs por meio de uma conexão segura (*syslog_transport* deve ser `tcp` quando o TLS está ativado).*
- **syslog_permitted_peer**
  O nome do host do servidor syslog a ser verificado ao usar TLS (curinga `*` permitido).
- **syslog_ca_cert**
  Autoridade de Certificação a ser confiada quando o TLS está ativado, se o certificado do servidor é autoassinado ou assinado por uma CA que não esteja disponível no armazenamento de certificados padrão.
- **syslog_cert**
  Certificado do cliente para rsyslog; quando o certificado de cliente e a chave do cliente são fornecidos, o TLS mútuo está ativado.
- **syslog_key**
  A chave do cliente (sem passphrase) para o rsyslog; quando o certificado cliente e a chave do cliente são fornecidos, o TLS mútuo está ativado.

Configuração de Exemplo

Os valores de configuração devem ser especificados como filhos de uma chave **uiconfig** no exemplo a seguir.

```
uiconfig:
  configuration_name: mutual_tls
  syslog_address: log1.logstash.example.com
  syslog_port: 5000
  syslog_transport: tcp
  syslog_fallback_servers:
    - address: log2.logstash.example.com
      port: 5001
      transport: tcp
    - address: log3.logstash.example.com
      port: 5001
      transport: tcp
  syslog_tls_enabled: true
  syslog_permitted_peer: *.logstash.example.com
  syslog_ca_cert: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  syslog_cert: |
```

```
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  syslog_key: |
    -----BEGIN RSA PRIVATE KEY-----
    -----END RSA PRIVATE KEY-----
```

Configurando o encaminhamento de log do aplicativo

É possível configurar o encaminhamento de logs de aplicativo do Cloud Foundry usando uma extensão integrada que está incluída com o IBM® Cloud Private Cloud Foundry.

A extensão usa o protocolo Lumberjack v2 para encaminhar logs de aplicativo de sua plataforma do IBM Cloud Private Cloud Foundry para um terminal remoto, como o plug-in de entrada do Logstash Beats.

A extensão `cfp-ext-applog-forwarder` ativa essa função e suporta uma série de opções de configuração diferentes, incluindo comunicações seguras que usam autenticação baseada em TLS e no certificado mútuo. Se você deseja encaminhar logs de aplicativo para o ElasticStack integrado no IBM Cloud Private, o gráfico do Helm `ibm-cflogging` poderá configurar automaticamente essa extensão durante a instalação do gráfico. Para obter mais informações, consulte Conectando-se ao Elasticstack no IBM Cloud Private.

Para enviar logs de aplicativo para um terminal remoto de sua escolha, ative a extensão `cfp-ext-applog-forwarder`. É possível ativar extensões usando uma CLI ou uma interface com o usuário. Para usar a CLI, prepare seu arquivo de configuração de acordo com os Valores de configuração. Em seguida, siga as instruções para Executando a extensão. Ignore a etapa de registro, já que essa extensão integrada é incluída com o produto.

Se você preferir usar a interface com o usuário, o `cfp-ext-applog-forwarder` oferece a edição orientada dos valores de configuração. Selecione um tipo de configuração de `Insecure`, `Server TLS` ou `Mutual TLS`. A interface com o usuário exibe os valores de configuração necessários e opcionais para o cenário selecionado. A interface com o usuário fornece descrições, valores de amostra e validação dos valores de configuração. Para obter informações sobre extensões, consulte Gerenciando extensões e Configurações.

Valores de Configuração

Forneça os valores a seguir para configurar o encaminhamento de log do aplicativo.

- **lumberjack_logs_enabled**
  - Padrão *: `false`
    Configure esse valor como `true` para ativar o envio de logs de aplicativo para o terminal do Lumberjack
- **lumberjack_logs_url**
  Endereço de IP ou nome de domínio e porta do servidor para receber logs do aplicativo
- **lumberjack_logs_use_tls**
  - Padrão *: `false`
    Configure esse valor como `true` para ativar uma conexão segura com o terminal
- **lumberjack_logs_mutual_tls**
  - Padrão *: `false`
    Ao usar TLS, configure esse valor como `true` para ativar a autenticação do cliente usando certificados
- **lumberjack_logs_server_name**
  Nome do host do terminal a ser verificado ao usar o TLS do Mutual
- **lumberjack_logs_ca_cert**
  Certificado ou cadeia de Autoridade de Certificação para confiança quando o TLS está ativado, quando o certificado do servidor é autoassinado ou assinado por uma CA que não está disponível no armazenamento de certificados padrão
- **lumberjack_logs_client_cert**
  Certificado do cliente para TLS mútuo
- **lumberjack_logs_client_key**
  Chave do cliente (sem passphrase) para TLS mútuo

Um terminal do Lumberjack v2 duplicado é suportado. Se ativado, a extensão sempre tenta encaminhar logs de aplicativo para os terminais primário e duplicado. Cada um desses valores de configuração é repetido com `lumberjack_logs` substituído por `lumberjack_logs_dup` na chave. Por exemplo, para ativar o terminal duplicado, deve-se configurar, no mínimo, **lumberjack_logs_dup_enabled** como `true` e fornecer o endereço do terminal e a porta como o valor para **lumberjack_logs_dup_url**.

Configuração de Exemplo

Os valores de configuração devem ser especificados como filhos de uma chave **uiconfig** no exemplo a seguir.

```
uiconfig:
  configuration_name: mutual_tls
  lumberjack_logs_enabled: true
  lumberjack_logs_url: log1.logstash.example.com:5000
  lumberjack_logs_use_tls: true
  lumberjack_logs_mutual_tls: true
  lumberjack_logs_server_name: log1.logstash.example.com
  lumberjack_logs_ca_cert: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  lumberjack_logs_client_cert: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  lumberjack_logs_client_key: |
    -----BEGIN RSA PRIVATE KEY-----
    -----END RSA PRIVATE KEY-----
  lumberjack_logs_dup_enabled: true
  lumberjack_logs_dup_url: log2.logstash.example.com:5000
  lumberjack_logs_dup_use_tls: true
  lumberjack_logs_dup_mutual_tls: true
  lumberjack_logs_dup_server_name: log2.logstash.example.com
  lumberjack_logs_dup_ca_cert: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  lumberjack_logs_dup_client_cert: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  lumberjack_logs_dup_client_key: |
    -----BEGIN RSA PRIVATE KEY-----
    -----END RSA PRIVATE KEY-----
```

Integrando syslogs do IBM® Cloud Private Cloud Foundry com Splunk

Se você usar Splunk Enterprise, será possível integrar seus logs do aplicativo IBM Cloud Private Cloud Foundry com o Splunk.

O Cloud Foundry usa o protocolo syslog RFC5425, portanto, também é necessário instalar o complemento Splunk que suporta esse protocolo.

Instalando o Splunk Enterprise usando Docker

1. Instale o Docker. Para obter mais informações, consulte Instalar Docker .
2. Efetue pull das imagens do Splunk por meio do Docker Hub:

   ```
   docker pull splunk/splunk
   ```

3. Inicie o Splunk:

   ```
   docker run -d -e "SPLUNK_START_ARGS=--accept-license" -e "SPLUNK_USER=root" -p "8000:8000" -p
   "12345:12345" splunk/splunk
   ```

   Nesse comando, `8000` é a porta padrão na qual a interface com o usuário do Splunk é executada, e `12345` é a porta TCP que recebe os logs do sistema do Cloud Foundry. É possível usar qualquer porta disponível para a porta TCP.

4. Para verificar se o Splunk está em execução, abra `http://localhost:8000` em um navegador. As credenciais de login padrão são `admin: changeme`.

Configurando o Splunk para receber os syslogs do Cloud Foundry

1. Faça download do complemento do syslog RFC5424 para Splunk. Deve-se ter uma conta do Splunk para fazer download do complemento. Para obter mais informações, consulte Syslog RFC5424 .
2. Extraia o arquivo `rfc5424-syslog_11.tar`.

   ```
   tar -xvzf rfc5424-syslog_11.tgz
   ```

3. Abra o arquivo `./rfc5424/default/transforms.conf` e substitua seu conteúdo pelo texto a seguir:

```
[rfc5424_host]
DEST_KEY = MetaData:Host
REGEX = <\d+>\d{1}\s{1}\S+\s{1}(\S+)
FORMAT = host::$1

[rfc5424_header]
REGEX = <(\d+)>\d{1}\s{1}\S+\s{1}\S+\s{1}(\S+)\s{1}(\S+)\s{1}(\S+)
FORMAT = prival::$1 appname::$2 procid::$3 msgid::$4
MV_ADD = true
```

4. Recupere o `containerId` do Docker para o contêiner Splunk.

```
docker ps -aqf "ancestor=splunk/splunk"
```

5. Copie a pasta `rfc5424` no contêiner do Docker para Splunk.

```
docker cp rfc5424 <containerId>:/opt/splunk/etc/apps
```

6. Reinicie o contêiner do Docker.

```
docker restart <containerId>
```

7. Abra `http://localhost:8000` em um navegador e efetue login no Splunk.
8. Clique em **Incluir dados > Monitor > TCP/UDP**.
9. Selecione a **Porta TCP** que você especificou quando iniciou o Splunk, como `12345` e, em seguida, clique em **Avançar**.
10. Para o **Tipo de origem**, selecione `rfc545_syslog` e, em seguida, clique em **Revisar e enviar**.
11. Para enviar dados de IBM Cloud Private Cloud Foundry para o Splunk, clique em **Iniciar procura**.

Envie syslogs do IBM Cloud Private Cloud Foundry para o Splunk

1. Efetue login no Cloud Foundry por meio da interface da linha de comandos (CLI).
2. Crie um serviço fornecido pelo usuário para o Splunk executando o comando a seguir:

```
cf create-user-provided-service <SERVICE-NAME> -l syslog://<splunkipaddress>:<port>
```

   Nota: No exemplo anterior, `<SERVICE-NAME>` é o nome do serviço Splunk, `<splunkipaddress>` é o endereço IP usado pelo Splunk e `<port>` é a porta TCP usada pelo Cloud Foundry para enviar syslogs para o Splunk.

3. Ligue este serviço a um aplicativo Cloud Foundry existente executando o comando a seguir:

```
cf bind-service <CF-APP-NAME> <SERVICE-NAME>
```

   Nota: No exemplo anterior, `<SERVICE-NAME>` é o nome para o serviço Splunk e `<CF-APP-NAME>` é o nome do aplicativo Cloud Foundry.

4. Remonte o aplicativo Cloud Foundry:

```
cf restage <CF-APP-NAME>
```

5. Confirme se você pode acessar os logs do Cloud Foundry no Splunk procurando por `sourcetype=rfc5424_syslog`.

Configure a liberação do Splunk Firehose Nozzle como um aplicativo do Cloud Foundry

É possível configurar o aplicativo da comunidade `splunk-firehose-nozzle` para enviar suas métricas de componente, logs do aplicativo e métricas do aplicativo do IBM® Cloud Private Cloud Foundry para o Splunk.

Antes de instalar o aplicativo Splunk Firehose Nozzle, deve-se instalar o Splunk em um contêiner. Se você não tiver instalado o Splunk, será possível instalá-lo usando o Docker. Se você já usa o Splunk, será possível configurar o complemento do Splunk para Cloud Foundry por meio do Marketplace do Splunk.

Instalando o Splunk

1. Instale o Docker. Para obter mais informações, consulte Instalar o Docker.
2. Obtenha as imagens de Splunk do Docker Hub.

   ```
   docker pull splunk/splunk
   ```

3. Inicie o Splunk:

   ```
   docker run -d -e "SPLUNK_START_ARGS=--accept-license" -e "SPLUNK_USER=root" -p "8000:8000" -p "8088:8088" splunk/splunk
   ```

   Neste exemplo, 8000 representa a porta padrão na qual o painel do Splunk está em execução e 8088 é a porta do Coletor de Eventos HTTP que recebe os logs do nozzle.

4. Para verificar se o Splunk está em execução, abra `http://localhost:8000` em seu navegador e efetue login. O nome do usuário padrão é `admin` e a senha padrão é `changeme`.

Instalando o Complemento do Splunk para Cloud Foundry

1. Clone o complemento do Splunk mais recente para o Cloud Foundry executando o comando a seguir:

   ```
   git clone git@github.com:splunk/splunk-addon-for-cloud-foundry.git
   ```

2. Recupere o containerId do Docker para Splunk executando o comando a seguir:

   ```
   docker ps -aqf "ancestor=splunk/splunk"
   ```

3. Copie a pasta `Splunk\_SA\_CloudFoundry` que você clonou para o contêiner do Docker executando o comando a seguir:

   ```
   docker cp splunk-addon-for-cloud-foundry/Splunk_SA_CloudFoundry <containerId>:/opt/splunk/etc/apps
   ```

Nota: use o valor `<containerId>` para o Splunk obtido na etapa 2.

1. Reinicie o contêiner do Docker.

   ```
   docker restart <containerId>
   ```

2. Abra o painel do Splunk em `http://localhost: 8000`. Se o complemento foi instalado com êxito, você verá um novo ícone com o título **Complemento Splunk para Cloud Foundry**.

Configurando o Coletor de Eventos HTTP Splunk

Use o Coletor de Eventos HTTP Splunk para enviar dados de HTTP para Splunk. In this case, you can send Cloud Foundry system logs.

1. No painel do Splunk, clique em **Configurações** > **Entradas de dados**
2. Na seção Coletor de Eventos HTTP, clique em **Incluir novo**.
3. Insira um nome para a entrada de dados e clique em **Avançar**.
4. Na lista **Tipo de origem**, selecione **Automático**.
5. Para o **Índice**, selecione um índice existente, como **Padrão**, ou crie um novo.
6. Clique em **Revisar** e, em seguida, em **Enviar** .
7. Salve o **Valor de Token Gerado** .
8. Ative o Coletor de Eventos HTTP:
   1. Clique em **Configurações** > **Entradas de Dados** .
   2. Clique em **Coletor de Eventos HTTP**.
   3. Clique em **Configurações Globais** .
   4. Na seção Todos os tokens, clique em **Ativado** e, em seguida, em **Salvar**
9. Splunk is now ready receive logs through HTTP by using port 8088. **Nota**: Se desejar usar uma porta diferente, é possível especificar seu valor nas Configurações globais.

Configurando e implementando o aplicativo de liberação Splunk Firehose Nozzle

1. Clone o aplicativo splunk-firehose-nozzle do GitHub.

   ```
   git clone https://github.com/cloudfoundry-community/splunk-firehose-nozzle.git
   cd splunk-firehose-nozzle
   ```

2. Efetue login no Cloud Foundry:

   ```
   cf login -a https://api.<your cf system domain> -u <your Cloud Foundry id>
   ```

   Se você não instalou a interface da linha de comandos do Cloud Foundry, consulte Interfaces da linha de comandos para IBM® Cloud Private Cloud Foundry.

3. Modifique o arquivo `nozzle_manifest.`

   ```
   vi ci/nozzle_manifest.yml
   ```

   - **API_ENDPOINT**: Endereço de terminal da API do Cloud Foundry.
   - **API_USER** : Cloud Foundry nome do usuário.
   - **API_PASSWORD** : Cloud Foundry senha do usuário.
   - **SPLUNK_TOKEN**: Para obter informações adicionais sobre este parâmetro, consulte Configurar e usar o Coletor de Eventos HTTP.
   - **SPLUNK_HOST**: Host do coletor de eventos HTTP Splunk, como `https://example.cloud.splunk.com:8088`.
   - **SPLUNK_INDEX**: O índice Splunk para o qual os eventos são enviados.
   - **FIREHOSE_SUBSCRIPTION_ID**: Marca eventos nozzle com um ID da assinatura de Firehose. Para obter informações adicionais, consulte Loggregator Guide for Cloud Foundry Operators. O Nozzle requer um usuário com o escopo `doppler.firehose`, e se o valor `ADD_APP_INFO` for `true`, o escopo `cloud_controller.admin_read_only`. Se o escopo `cloud_controller.admin_read_only` não estiver disponível no sistema, use o valor `cloud_controller.admin`.

4. Envie o nozzle por push executando o seguinte comando:

   ```
   make deploy-nozzle
   ```

Conectando o IBM Cloud Private Cloud Foundry ao Prometheus

IBM® Cloud Private Cloud Foundry: Exceto para a seção de painéis do Grafana, essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry. A seção de painéis do Grafana também se aplica ao Cloud Foundry Enterprise Environment.

Os exportadores do IBM® Cloud Private Cloud Foundry Prometheus são usados para agregar dados de monitoramento do IBM Cloud Private Cloud Foundry. Os exportadores desenham com base nos conjuntos de monitoramento BOSH, IBM Cloud Private Cloud Foundry e Firehose para fornecer informações sobre o funcionamento da implementação do IBM Cloud Private Cloud Foundry. Instale o gráfico do Helm `chart-cf-exporters-prometheus` para instalar esses exportadores no ambiente IBM Cloud Private Cloud Foundry.

Pré-requisitos

- O Kubernetes 1.4 ou mais recente com as APIs beta ativadas. Conclua estas etapas:
  - Efetue login no Kubernetes.
  - Clique no ícone do usuário e escolha **Configurar cliente**.
- Instale a interface da linha de comandos (CLI) `kubectl`, caso ela ainda não esteja instalada.
  - Copie e cole os comandos para configurar a CLI.

    Nota: atualize o valor de `--namespace` com o namespace que você deseja usar para hospedar os exportadores do IBM Cloud Private Cloud Foundry.
- Instale o `cloudctl` IBM Cloud Private CLI. Para obter mais informações, consulte Instalando a CLI do IBM® Cloud Private.
- Instale a CLI do Helm. Para obter mais informações, consulte Instalando a CLI do Helm (helm)
- Implemente IBM Cloud Private Cloud Foundry.

Instalando o gráfico

Antes de iniciar

- Acesse o `diretório de configuração de instalação` e, em seguida, o `IBMCloudPrivate/chart-cf-exporters-prometheus.tgz`.
  - O `values.yaml` e os gráficos do Helm `chart-cf-exporters-prometheus` estão localizados aqui.

Configuration

- A tabela a seguir lista os parâmetros configuráveis do gráfico `chart-cf-exporters-prometheus` e seus valores padrão.
- Esses parâmetros **devem** ser configurados **antes** de instalar o gráfico Helm. Esses parâmetros são mapeados para as variáveis de ambiente que estão definidas em `values.template_yaml`.
- Todos os arquivos yaml estão no `installation configuration directory` após a instalação.

Parâmetro	Descrição	Padrão
`cf_api_url`	URL da API do Cloud Foundry `https://api.DOMAIN`	Para `DOMAIN`, consulte **bmxconfig.yml:**

`bmxconfig.bluemix_env_domain` | | `cf_client_secret` | Cloud Foundry `cf_exporter` ID do segredo do cliente | Veja **credentials.yml:** `credentials.cloudfoundry.uaa_cf_exporter_secret` | | `uaa_url` | URL de UAA do Cloud Foundry `https://uaa.DOMAIN` | Para `DOMAIN`, consulte **bmxconfig.yml:** `bmxconfig.bluemix_env_domain` | | `firehose_client_secret` | O segredo para o cliente, `firehose_client` | Veja **credentials.yml:** `credentials.cloudfoundry.uaa_firehose_exporter_secret.secret` | | `firehose_doppler_url` | URL de Doppler do Cloud Foundry `wss://doppler.DOMAIN:443 URL` | Para `DOMAIN`, consulte **bmxconfig.yml:** `bmxconfig.bluemix_env_domain` | | `bosh_url` | URL do diretor BOSH `https://IP_ADDRESS:25555 URL` | Para `IP_ADDRESS`, consulte **bmxconfig.yml:** `bmxconfig.director_ip` | | `bosh_pwd` | Senha do usuário do BOSH `admin` | Veja **credentials.yml:** `credentials.boshdirector.bosh.password` | | `director_cert` | Certificado de CA do diretor BOSH | Veja **certificates.yml:** `certificates.rootca.certificates.data` | | `environment` | O nome do ambiente para exibir no Prometheus e Grafana Deve ser exclusivo por IBM Cloud Private Cloud Foundry implementação | Cloud Foundry |

1. Acesse o arquivo `IBMCloudPrivate/chart-cf-exporters-prometheus.tgz` no `diretório de configuração de instalação`.
2. Inspecione o `values.yaml` ou crie um `values.yaml` a partir do `values.template_yaml`, editando parâmetros de acordo com a tabela anterior.
3. Execute o comando a seguir:

   ```
   Helm install -- name = cf-exportadores. -- tls
   ```

O comando de instalação do Helm falha com a mensagem a seguir:

```
Error: release cf-exporters failed: Internal error occurred: admission webhook
"trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the request:
Deny "docker.io/boshprometheus/firehose-exporter", no matching repositories in ClusterImagePolicy
and no ImagePolicies in the "default" namespace
```

Siga as etapas para corrigir esse problema:

1. Navegue para **Gerenciar** > **Segurança do recurso**

   Políticas de Imagem.
2. Clique em **Criar política de imagem**.
3. Forneça um nome.
4. Escolha **Namespace**. Selecione o namespace no qual você instalará o exportador.
5. Clique em **Incluir registro** e preencha a URL do Registro = docker.io/boshprometheus/*.
6. Deixe a Varredura do VA desligada e clique em **Incluir**.
7. Clique em **Incluir** novamente para ativar a política.

Resultados

- O comando implementa os exportadores do IBM Cloud Private Cloud Foundry em um cluster do Kubernetes que se comunica com a implementação do IBM Cloud Private Cloud Foundry.

Painéis do Grafana

Com sua instalação do Cloud Foundry Enterprise Environment, cinco painéis Grafana são carregados automaticamente.

Quando você abrir o Grafana, o painel pode não estar visível na organização (namespace) existente. Os painéis são implementados no mesmo namespace no qual o instalador do Cloud Foundry Enterprise Environment foi implementado.

Para alternar para a organização apropriada, passe o mouse sobre o ícone do perfil do usuário na barra de navegação, selecione **Org. Atual: Alternar** e escolha o nome que corresponde ao namespace no qual a ferramenta de implementação foi fornecida. Haverá cinco painéis disponíveis após você mudar a organização.

Se necessário, é possível incluir painéis em sua implementação. Painéis adicionais estão disponíveis nestes locais:

- Painéis do IBM Cloud Private Cloud Foundry

- Painéis do Bosh

- Usando o console do IBM Cloud Private , vá para `/grafana`.

- Clique em **Início** > **Importar Painel** .
- Escolha o arquivo `.json` no repositório GitHub que foi listado anteriormente para o painel que você deseja instalar. Copie o conteúdo e cole-o em `Importar painel` no IBM Cloud Private. Em seguida, clique em **Carregar**.
- Quando for solicitado, mude o `Nome`, se necessário e, em seguida, selecione `prometheus` no campo **Prometheus**.
- Clique em **Importar**.
- Você é levado ao painel, que agora é acessível por meio do menu **Página inicial**.

Desinstalando o Gráfico

Execute o comando a seguir para desinstalar ou excluir a implementação `my-release`:

```
Helm delete cf-exportadores -- tls
```

Esse comando remove todos os componentes do Kubernetes que estão associados com o gráfico e exclui a liberação.

Conectando ao Elastic Stack no IBM Cloud Private

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

O IBM® Cloud Private Cloud Foundry fornece um gráfico do Helm que é chamado de `ibm-cflogging`. Esse gráfico do Helm pode ser instalado no IBM Cloud Private para permitir que os logs de ambas as plataformas sejam coletados e analisados em uma única instalação do Elastic Stack no IBM Cloud Private. Essa integração depende das extensões `cfp-ext-syslog-syslog-forwarder` e `cfp-ext-applog-forwarder` para o IBM Cloud Private Cloud Foundry, que podem ser configuradas automaticamente durante a instalação do gráfico do Helm. Se você optar por não aplicar a configuração durante a instalação, ela poderá ser recuperada posteriormente e aplicada às extensões manualmente. Também é possível usar as extensões para encaminhar logs do sistema e logs de aplicativo da plataforma Cloud Foundry para um terminal de sua escolha. Para obter mais informações, consulte Configurando o encaminhamento de logs do sistema de plataforma e Configurando o encaminhamento de log do aplicativo.

- Pré-requisitos
- Carregando o archive de gráfico
- Instalando o gráfico
- Ativando o encaminhamento de logs
- Processamento de log

Pré-requisito

Deve-se ter um ambiente do IBM Cloud Private 3.1 ou mais recente com o gráfico `ibm-icplogging` 2.0.0 ou mais recente instalado. (Esse gráfico é instalado por padrão durante a instalação do IBM® Cloud Private como `logging` de liberação do Helm). Deve-se também ter o IBM Cloud Private Cloud Foundry instalado, já que o gráfico do Helm `ibm-cflogging` é exportado durante a instalação.

- Instale a CLI do `cloudctl` IBM Cloud Private e a CLI do `kubectl`. Efetue login em seu cluster. Certifique-se de que você tenha como destino o namespace no qual o gráfico `ibm-icplogging` está instalado, que é `kube-system` por padrão. A CLI `kubectl` é configurada automaticamente ao efetuar login usando o `cloudctl`. Consulte Instalando a CLI do IBM® Cloud Private.
- Instale e configure a CLI do Docker para usar com o cluster. Consulte Configurando a autenticação para a CLI do Docker

Carregando o archive de gráfico

Após a instalação do IBM Cloud Private Cloud Foundry, é possível localizar o archive de gráfico no diretório `<data_directory>/IBMCloudPrivate` no sistema no qual você executou o instalador do Cloud Foundry. (`<data_directory>` é o diretório fornecido para o script `launch.sh` usando a opção `-c`.)

O archive de gráfico é denominado `ibm-cflogging-1.1.0-archive.tgz` e contém o gráfico do Helm e uma imagem necessária. Se você instalou e configurou as CLIs conforme descrito em Pré-requisitos em um sistema diferente, copie o archive para esse sistema. Certifique-se de ter efetuado login em seu cluster do IBM® Cloud Private e de direcioná-lo para o mesmo namespace que a liberação de `ibm-icplogging` que você deseja usar. Certifique-se de que a CLI do Docker esteja com login efetuado no registro de imagem privado para seu cluster. Em seguida, execute o comando a seguir:

```
cloudctl catalog load-archive -- archive ibm-cflogging-1.1.0-archive.tgz
```

Por padrão, esse comando carrega o gráfico no repositório do Helm `local-charts`. O comando também carrega a imagem no registro de imagem privado em que ele é acessível apenas por gráficos que estão instalados no namespace de destino atual. Para obter mais informações, consulte o comando `cloudctl catalog load-archive e Gerenciando imagens.

Clusters em um ambiente de airgap

O archive de gráfico inclui uma imagem necessária, mas também depende de outras imagens que estão publicamente disponíveis na Internet. Se seu cluster não tiver conectividade com a Internet, serão necessárias etapas adicionais após o carregamento do archive de gráfico.

1. Localize e clique no gráfico `ibm-cflogging` no catálogo. Faça download do arquivo `.tgz` do gráfico a partir da seção **Origem & Arquivos TAR**.
2. Em um computador com conectividade de Internet e Docker, execute o comando a seguir.

   ```
   cloudctl catalog create-archive --chart ibm-cflogging-1.1.0.tgz --archive ibm-cflogging-1.1.0-archive-offline.tgz
   ```

3. Carregue esse novo archive dentro do catálogo, substituindo o gráfico original.

   ```
   cloudctl catalog load-archive -- archive ibm-cflogging-1.1.0-archive-offline.tgz
   ```

As imagens adicionais agora são carregadas no repositório de imagem privado e os valores padrão do gráfico são atualizados para se referir a essas imagens.

Instalando o gráfico

O gráfico do Helm não requer configuração para instalação, mas para evitar uma configuração manual, ative a configuração automática de extensões do Cloud Foundry. Para isso, siga as instruções em Ativando a configuração automática antes de instalar o gráfico.

A instalação pode ser concluída usando o catálogo do IBM Cloud Private ou a CLI do Helm.

Ativando a Configuração Automática

Para ativar a configuração automática, é necessária a URL do gerenciador de configuração e o Token do gerenciador de configuração. Esses são os mesmos valores que são inseridos ao configurar a ferramenta de implementação do IBM Cloud Private Cloud Foundry. Consulte Instalando o IBM Cloud Private Cloud Foundry com o Ferramenta de implementação do Cloud Foundry.

Deve-se criar um segredo do Kubernetes que contenha o token do gerenciador de configuração no campo `configManagerToken`. Ele deve ser criado no namespace de destino para o gráfico do Helm. Por exemplo, crie o arquivo `cf-config-manager-secret.yaml` com o conteúdo a seguir, substituindo `{{token}}` pelo valor real do token.

```
apiVersion: v1
kind: Secret
metadata:
  name: cf-config-manager-secret
type: Opaque
stringData:
  configManagerToken: {{token}}
```

Em seguida, crie o segredo:

```
kubectl create -f cf-config-manager-secret.yaml
```

É necessário fornecer o nome do segredo ao seguir as instruções para instalação usando o catálogo do IBM Cloud Private ou a CLI do Helm do .

Catálogo do IBM Cloud Private

Localize e clique no gráfico `ibm-cflogging` no catálogo. A visão geral contém informações detalhadas sobre todos os parâmetros de configuração do gráfico. Alterne para a guia **Configuração** ou clique em **Configurar**.

- Insira um nome exclusivo para o **Nome da liberação do Helm**.
- Selecione o **Namespace de destino** (deve corresponder ao namespace da liberação `ibm-icplogging`, normalmente `kube-system`).
- Aceite a **Licença** .

- Por padrão, o Logstash é configurado para receber os logs do sistema e do aplicativo. Na seção **Todos os parâmetros**, é possível mudar esse parâmetro desmarcando a caixa de seleção **Ativar** sob a extensão **cfp-ext-syslog-forwarder extension** ou **cfp-ext-applog-forwarder extension**. Deve-se ter pelo menos uma dessas extensões ativadas.
- Se quiser ativar a configuração automática de extensões do Cloud Foundry:
  - Na seção **Todos os parâmetros**, em **Configuração do IBM Cloud Private**, marque a caixa de seleção **Ativar configuração automática** e preencha os campos **URL do gerenciador de configuração** e **Segredo do token do gerenciador de configuração**.
- Clique em **Instalar** .

CLI do Helm

Se você preferir instalar o gráfico usando a CLI `helm`, primeiro consulte Instalando a CLI do Helm (helm) para obter instruções sobre a instalação da CLI `helm`. O Helm é configurado automaticamente quando você efetua login usando `cloudctl`.

Se você executou a configuração que está descrita em Incluindo o repositório interno do Helm na CLI do Helm, é possível executar o seguinte comando.

```
helm install local-charts/ibm-cflogging --version 1.1.0 --name <release_name> --namespace kube-system --tls
```

Como alternativa, é possível fazer download do gráfico e instalá-lo a partir do arquivo TAR. Visualize a visão geral do gráfico no catálogo do IBM® Cloud Private para obter detalhes sobre os parâmetros de configuração e faça download do arquivo `.tgz` do gráfico a partir da seção **Origem & Arquivos TAR**. Minimamente, tudo o que é necessário é o nome para a liberação.

```
helm install ibm-cflogging-1.1.0.tgz --name <release_name> --namespace kube-system --tls
```

Se quiser ativar a configuração automática de extensões do Cloud Foundry, configure `cloudFoundry.configureExtensions` para `true` e forneça valores para `cloudFoundry.configManagerUrl` e `cloudFoundry.configManagerTokenSecret`.

Para desativar o encaminhamento de logs do sistema, configure `syslogForwarder.enabled` como `false`. Para desativar o encaminhamento de logs de aplicativo, configure `applogForwarder.enabled` como `false`.

Se desejar configurar esses ou outros parâmetros de configuração, crie um arquivo YAML que contenha os valores e forneça-o usando a opção `--values`.

Ativando o encaminhamento de logs

Depois de instalar o gráfico, deve-se ativar uma implementação do Cloud Foundry para atualizar as instâncias do componente Cloud Foundry para que elas comecem a encaminhar logs.

Se você ativou a configuração automática, a extensão `cfp-ext-syslog-forwarder`, a extensão `cfp-ext-applog-forwarder`, ou ambas, elas serão inseridas nos estados de implementação e configuradas. Clique no botão `Iniciar implementação` na ferramenta de implementação Cloud Foundry ou execute o script `launch_deployment.sh`. Para obter informações adicionais, consulte Instalando o IBM Cloud Private Cloud Foundry com o Ferramenta de implementação do Cloud Foundry ou Instalando o IBM Cloud Private Cloud Foundry.

Se você não ativou a configuração automática, as notas para a liberação do Helm que são criadas por sua instalação especificam o nome do segredo do Kubernetes no qual é possível localizar a configuração para `cfp-ext-syslog-forwarder`, `cfp-ext-applog-forwarder`, ou ambos. Siga as instruções para instalar e configurar essas extensões e, em seguida, atualize a implementação do Cloud Foundry.

- Configurando o encaminhamento de logs do sistema de plataforma
- Configurando o encaminhamento de log do aplicativo

Processamento de log

Todos os logs do sistema que são recebidos da extensão `cfp-ext-syslog-forwarder` são gerados pelo Logstash para um índice datado no Elasticsearch. O índice é nomeado usando o valor fornecido para `syslogForwarder.logstash.index`, que é `cloudfoundry` por padrão. Por exemplo, usando o valor padrão, os nomes de índice têm o formato `cloudfundry-YYYY.Y.MM.DD`. Crie um padrão de índice no primeiro uso do Kibana com o padrão `cloudfoundry-*`.

O documento resultante para Elasticsearch inclui os campos a seguir.

Campo	Valor
`message`	O conteúdo do log completo não analisado

Campo	Valor
type	O valor do parâmetro de configuração syslogForwarder.logstash.type

(padronizado para platform) syslog5424_pri | Valor de prioridade bruto syslog_facility | Nome da instalação determinado a partir de syslog5424_pri syslog_facility_code | Código de recurso determinado a partir do syslog5424_pri syslog_severity | Nome da Severidade determinado a partir de syslog5424_pri syslog_severity_code | Código de severidade determinado a partir de syslog5424_pri syslog5424_ver | Versão do protocolo syslog (o RFC 5424 define a versão 1) syslog5424_ts | Registro de data e hora no formato ISO8601 syslog5424_host | Nome do host ou endereço IP da MV BOSH syslog5424_app | Nome do aplicativo (nome da tarefa BOSH) syslog5424_proc | ID do Processo (significado varia por componente) syslog5424_sd | Dados Estruturados. Os logs do Cloud Foundry usam o ID de Dados Estruturados instance@47450 director | Nome do diretor BOSH. Parsed from syslog5424_sd. (Exemplo: IBMCloudPrivate) deployment | Nome da implementação do BOSH. Parsed from syslog5424_sd. (Exemplo: Bluemix) group | Nome do grupo de instâncias BOSH. Parsed from syslog5424_sd. (Exemplo: uaa) az | Zona de disponibilidade. Analisado a partir de syslog5424_sd id | ID da VM BOSH. Analisado a partir de syslog5424_sd sslsubject | Assunto SSL do certificado de cliente para o encaminhador de syslog. (Exemplo: /OU=IBM Cloud Private / CN=syslog_forwarder) syslog5424_msg | A saída da mensagem de log pelo componente Cloud Foundry. Se essa mensagem contiver dados JSON, esses dados serão analisados ainda mais no syslog5424_msg_json e nos subcampos.

Para obter mais informações, consulte RFC 5424 - O Protocolo Syslog.

Todos os logs de aplicativo que são recebidos da extensão cfp-ext-applog-forwarder são gerados por Logstash para um índice datado no Elasticsearch. O índice é nomeado usando o valor fornecido para applogForwarder.logstash.index, que é cloudfoundry-apps por padrão. Por exemplo, usando o valor padrão, os nomes de índice têm o formato cloudFundry-apps-YYYY.MM.DD. Crie um padrão de índice no primeiro uso do Kibana com o padrão cloudfundry-apps-*.

O documento resultante para Elasticsearch inclui os campos a seguir.

Campo	Valor
message	A mensagem de log
type	O valor do parâmetro de configuração applogForwarder.logstash.type

(padronizado para application) app_id | ID do aplicativo no CCloud Foundryloud Foundry app_name | Nome do aplicativo no Cloud Foundry space_id | ID do espaço no Cloud Foundry space_name | Nome do espaço no Cloud Foundry org_id | ID da organização no Cloud Foundry org_name | Nome da organização no Cloud Foundry message_type | OUT (fluxo stdout) ou ERR (fluxo stderr) source_id | Onde no Cloud Foundry a mensagem é originada - App, RTR (roteador) instance_id | Instância de origem que registrou a mensagem origin | Qual componente no Cloud Foundry encaminhou a mensagem

Retenção do Log

A retenção de log depende do índice Logstash de destino que você fornece para o encaminhamento de syslog e applog através dos valores syslogForwarder.logstash.index e applogForwarder.logstash.index. Com os valores padrão de cloudfoundry e cloudfoundry-apps, todos os logs são retidos por tempo indeterminado, e isso pode esgotar rapidamente o espaço de armazenamento alocado para o Elastic Stack.

Por padrão, o gráfico ibm-icplogging cura os logs para os índices logstash-YYYY.MM.DD usando uma tarefa diária que exclui todos os dias de logs, exceto o último. Para uma cura simples de logs do Cloud Foundry, é possível configurar syslogForwarder.logstash.index e applogForwarder.logstash.index para logstash.

Como alternativa, para usar os índices de destino cloudfoundry e cloudfoundry-apps e manter os syslogs e applog do Cloud Foundry separados dos outros logs do IBM Cloud Private, é possível incluir na lista esses índices customizados que são limpos pelo curador com as seguintes etapas. Entretanto, as modificações na configuração do curador serão perdidas se a liberação de criação de log for atualizada ou reinstalada.

1. Salve o mapa de configuração do curador atual (chamado logging-elk-elasticsearch-curator-config ou <release-name>-ibm-icplogging-elasticsearch-curator-config se você implementou uma pilha de criação de log customizada) em um arquivo YAML e crie uma cópia de backup desse arquivo.

   ```
   kubectl -n kube-system get cm logging-elk-elasticsearch-curator-config -o yaml > logging-elk-elasticsearch-curator-config.yaml
   ```

2. Modifique o arquivo incluindo um sexto prefixo para que o curador seja limpo sob a chave actions da seção action.yml. Se você estiver usando os índices de destino cloudfoundry e cloudfoundry-apps, um prefixo cobrirá ambos os casos. A amostra a seguir mostra um período de retenção padrão de um dia, mas é possível customizar essa amostra, dependendo

do volume dos logs gerados pela implementação do Cloud Foundry. Certifique-se de corresponder à indentação das outras ações para obter um YAML formatado corretamente.

```
   6:
     action: delete_indices
     description: "Delete Cloud Foundry log indices that are older than 1 days. Cron schedule:
30 23 * * *"
     options:
       timeout_override:
       continue_if_exception: True
       ignore_empty_list: True
       disable_action: False
     filters:
     - filtertype: pattern
       kind: prefix
       value: cloudfoundry-
     - filtertype: age
       source: name
       direction: older
       timestring: '%Y.%m.%d'
       unit: days
       unit_count: 1
```

3. Substitua o mapa de configuração pelas suas modificações:

```
kubectl -n kube-system replace -f logging-elk-elasticsearch-curator-config.yaml
```

Trabalhando com serviços

É possível configurar o acesso aos serviços do IBM Cloud e aos serviços de banco de dados para suas organizações e espaços.

- Usando os serviços de banco de dados do IBM Cloud Private no IBM Cloud Private Cloud Foundry
- Usando os serviços do IBM Cloud no IBM Cloud Private Cloud Foundry

Usando serviços de banco de dados do IBM Cloud Private no IBM Cloud Private Cloud Foundry

Ao registrar o broker de serviço no IBM® Cloud Private Cloud Foundry, o catálogo de serviços se torna disponível para consumo.

Depois que o broker de serviço para o banco de dados é implementado no IBM® Cloud Private, é possível registrar o broker de serviço no IBM Cloud Private Cloud Foundry para importar o catálogo do broker de serviço no IBM Cloud Private Cloud Foundry e provisionar e ligar instâncias de serviço do IBM Cloud Private Cloud Foundry.

Antes de iniciar Deve-se implementar o banco de dados do Open Service Broker (OSB) no IBM Cloud Private. Para obter informações sobre como implementar o OSB, consulte Implementando o banco de dados do Open Service Broker (OSB) no IBM Cloud Private.

Conclua as etapas a seguir para configurar os serviços de banco de dados do IBM Cloud Private no IBM Cloud Private Cloud Foundry:

1. Use a CLI do IBM Cloud Private Cloud Foundry para efetuar login no IBM Cloud Private Cloud Foundry. Execute o seguinte comando para registrar o broker de serviço. `<external-nodeport>` é o número da porta externa obtido ao expor o broker de serviço para acesso externo do IBM Cloud Private.

   ```
   $ cf create-service-broker <your-broker-name> admin password https://<ICPExternalClusterIP>:
   <external-nodeport>
   ```

2. Listar serviços. Isso lista os serviços disponíveis e suas permissões de acesso.

   ```
   $cf service-access
   ```

3. Conceda acesso aos serviços usando este comando.

   ```
   $ cf enable-service-access <service-name>
   ```

4. Execute o comando a seguir para visualizar uma lista de serviços ativados.

```
$cf marketplace
```

Usando serviços do IBM Cloud no IBM Cloud Private Cloud Foundry

Com o acesso do IBM® Cloud Private Cloud Foundry aos serviços do IBM Cloud, seu administrador do IBM Cloud Private Cloud Foundry pode decidir quais serviços tornar acessíveis ao mercado de trabalho do IBM Cloud Private Cloud Foundry para várias organizações e espaços.

Antes de iniciar

Você precisa dos recursos a seguir para implementar serviços do IBM Cloud no IBM Cloud Private Cloud Foundry.

1. IBM Cloud Private Cloud Foundry deve ser instalado. Para obter instruções de instalação, consulte Instalando o IBM Cloud Private Cloud Foundry .
2. Deve-se ter uma conta do serviço do IBM Cloud ativa para configurar a integração. Se você não tiver uma conta, será possível criar uma no IBM Cloud.
3. Seu ambiente do IBM Cloud Private Cloud Foundry deve ter acesso ao terminal da API do IBM Cloud, `https://api.ng.bluemix.net`. Ajuste os firewalls para cada caso de uso de serviço do IBM Cloud. Atualmente, somente o terminal de API `api.ng.bluemix.net` é suportado na região Norte Americana. Quando outros terminais se tornarem disponíveis, será possível mudar a região configurando o atributo `public_api_target` no `uiconfig` para a URL de API apropriada para a região.

Conclua as etapas a seguir para integrar os serviços do IBM Cloud ao IBM Cloud Private Cloud Foundry.

- Criar uma Chave de API
- Criar o broker de serviço
- Verificar conexão com o domínio do IBM Cloud
- Excluir chave de API
- Gerenciar a acessibilidade de serviço do IBM Cloud
- Interagindo com Serviços

Criar uma Chave de API

Para criar o broker de serviço, o administrador deve fornecer uma chave de API. Você precisa da chave API somente ao configurar o broker de serviço. É possível descartar a chave após a configuração ser concluída.
É possível criar a chave de API a partir de qualquer local em que você instalou a CLI do IBM Cloud.
Nas etapas a seguir, a CLI do IBM Cloud é instalada em um contêiner de concepção.

1. No diretório de instalação, conecte-se ao contêiner de concepção:

   ```
   ./connect -n IBMCloudPrivate
   ```

2. Efetue login no IBM Cloud:

   ```
   logins ibmcloud -a https://api.ng.bluemix.net
   ```

   Use a opção a seguir se você tiver uma conta federada:

   ```
   logins ibmcloud -sso https://api.ng.bluemix.net
   ```

3. Criar uma chave API. Você precisa da chave em uma etapa posterior.

   ```
   ibmcloud iam api-key-create NAME [-d DESCRIPTION]
   ```

4. Efetue logout do IBM Cloud.

   ```
   logout ibmcloud
   ```

5. Digite `exit` para sair do contêiner de concepção.

Criar o broker de serviço

O gerenciador de configuração inclui um comando para criar o broker de serviço do IBM Cloud.
Em seu diretório de instalação, execute o comando a seguir para criar o broker de serviço:

```
./cm public-service-broker create --apikey <Your_API_KEY> [--resource-group <Resource_group>]
```

Se você omitir o parâmetro `resource-group`, o grupo de recursos padrão será usado.
Nesse contêiner de concepção, use o comando a seguir para listar todos os grupos de recursos disponíveis.

```
grupos de recursos ibmcloud
```

O grupo de recursos padrão é listado durante o `ibmcloud login`. Também é possível usar o comando a seguir para listar o grupo padrão:

```
ibmcloud grupos de recursos --padrão
```

Após a conclusão do comando `cm`, o IBM Cloud Private Cloud Foundry é vinculado ao IBM Cloud.

Verificar conexão com o domínio do IBM Cloud

1. No diretório de instalação, use o comando a seguir para conectar-se ao contêiner de concepção:

   ```
   ./connect -n IBMCloudPrivate
   ```

2. Use as credenciais do usuário administrador {site.data.keyword.icpcf-notm}} para efetuar login no IBM Cloud Private Cloud Foundry.u

   ```
   cf login
   ```

3. Use o seguinte comando para visualizar o broker de serviço `IBMCloud.<extension>.<extension>` é o domínio do IBM Cloud. Por exemplo, IBMCloudPublic.ng.bluemix.net.

   ```
   cf service-brokers
   ```

4. Efetue logout de {site.data.keyword.icpcf-notm}}.

   ```
   cf logout
   ```

5. Digite `exit` para sair do contêiner de concepção.

Excluir chave de API

Conclua as etapas a seguir para excluir a chave API.

1. No diretório de instalação, conecte-se ao contêiner de concepção.

   ```
   ./connect -n IBMCloudPrivate
   ```

2. Efetue login no IBM Cloud.

   ```
   logins ibmcloud -a https://api.ng.bluemix.net
   ```

   Use a opção a seguir se você tiver uma conta federada:

   ```
   logins ibmcloud -sso https://api.ng.bluemix.net
   ```

3. Exclua uma chave de API. `<NAME>` é o nome usado ao criar a chave de API.

   ```
   ibmcloud iam api-key-delete <NAME>
   ```

4. Efetue logout do IBM Cloud.

   ```
   logout ibmcloud
   ```

5. Digite `exit` para sair do contêiner de concepção.

Gerenciar a acessibilidade de serviço do IBM Cloud

- Para listar os planos de serviço e o acesso atual que estão disponíveis como um resultado do registro, use o comando a seguir:

  ```
  cf service-access
  ```

  Para obter mais informações sobre o acesso de serviço, consulte Exibir acesso aos planos de serviço.

- Para ativar os planos de serviço e gerenciar o acesso de nível de organização, use o comando a seguir:

```
cf enable-service-access
```

Para obter mais informações sobre como ativar o acesso de serviço, consulte Ativar acesso aos planos de serviço.

- Para remover as permissões do plano de serviço, use o comando a seguir:

  ```
  cf disable-service-access
  ```

  Para obter mais informações sobre a desativação do acesso de serviço, consulte Desativar acesso aos planos de serviço.

Interagindo com Serviços

As instâncias de serviço podem ser criadas para serviços disponíveis no mercado de trabalho. Para obter mais informações sobre como interagir com serviços, consulte Gerenciando instâncias de serviço com a CLI cf

IBM Cloud Private Cloud Foundry Guia de ferramentas da CLI

O IBM® Cloud Private Cloud Foundry inclui uma opção de interface da linha de comandos

- Interfaces da linha de comandos para o IBM Cloud Private Cloud Foundry

Interfaces da linha de comandos para IBM Cloud Private Cloud Foundry

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

O contêiner de concepção, que pode ser acessado usando connect.sh, já tem a CLI do BOSH e a CLI do Cloud Foundry instaladas. No entanto, pode haver situações em que você deseja executar comandos da CLI do BOSH ou da CLI do Cloud Foundry a partir de outra máquina. Neste caso, pode ser necessário instalar algumas interfaces da linha de comandos (CLIs).

- BOSH: use essa CLI para gerenciar o BOSH Director, que gerencia todas as máquinas virtuais implementadas.
- Cloud Foundry (CLI Cloud Foundry): use essa CLI para executar comandos principais do Cloud Foundry.

BOSH

Para instalar a CLI do BOSH v2, consulte Interface da linha de comandos do BOSH

Para obter mais informações sobre a CLI do BOSH v2, consulte Documentação da comunidade da CLI do BOSH

Comandos úteis do BOSH

- Destine um BOSH Director. Deve-se estar apto a atingir o endereço IP `director_ip` por meio das redes conectadas.

  ```
  #If you migrated from 3.1.0
  bosh alias-env IBMCloudPrivate -e https://<director_ip>:25555 --ca-cert <(bosh int
<installation configuration directory>/data/CloudFoundry/certificates.yml --path
/certificates/rootca/certificate/data)

  #If this is a fresh deployment
  bosh alias-env IBMCloudPrivate -e https://<director_ip>:25555 --ca-cert <(bosh int
<installation configuration directory>/data/CloudFoundry/director-store.yml --path
/director_ssl/ca)
  ```

- Autentique a CLI do BOSH com um BOSH Director. Nos comandos a seguir, o nome do usuário é `admin`. A senha é o valor de parâmetro `credentials.boshdirector.bosh.password` no arquivo `credentials.yml` na pasta `./CloudFoundry` do diretório de configuração de instalação.

  ```
  export BOSH_ENVIRONMENT=IBMCloudPrivate bosh login

  logout de bosh
  ```

  Para configurar os valores de login do cliente BOSH como variáveis de ambiente:

  ```
  export BOSH_CLIENT=admin
  export BOSH_CLIENT_SECRET=`bosh int /travis/data/CloudFoundry/credentials.yml --path
  <installation configuration di
  ```

- Liste implementações BOSH que são atualmente gerenciadas pelo Director conectado:

  ```
  implementações bosh
  ```

- Liste as máquinas virtuais que estão associadas a uma implementação:

  ```
  bosh -d DEPLOYMENT_NAME vms [--details | --vitals]
  ```

- Direcione uma implementação BOSH com um comando **único**:

  ```
  bosh -d DEPLOYMENT BOSH_COMMAND
  ```

- Destine um arquivo manifest de implementação BOSH de modo persistente:

  ```
  export BOSH_DEPLOYMENT=DEPLOYMENT_NAME
  ```

- Exporte o manifest de implementação:

  ```
  bosh -d DEPLOYMENT_NAME download manifest MANIFEST_FILE.yml
  ```

- Liste tarefas BOSH recentes:

  ```
  tarefas de bosh recentes
  ```

  Uma lista de números de ID da tarefa BOSH é exibida.

- Visualize detalhes da tarefa BOSH:

  ```
  bosh task ID_NUMBER
  ```

  Nota: é possível obter uma lista de números de tarefas exibindo as tarefas BOSH recentes.

Cloud Foundry

Para instalar a CLI do Cloud Foundry, consulte Instalando a CLI do cf.

Para obter mais informações sobre a CLI do Cloud Foundry, consulte Cloud FoundryInterface da linha de comandos (cf CLI).

Comandos úteis do Cloud Foundry

- Destine uma plataforma Cloud Foundry:

  ```
  cf api https://api.<YOUR_DOMAIN>
  ```

- Autentique a CLI do Cloud Foundry com uma plataforma Cloud Foundry:

  ```
  cf login
  ```

  ```
  cf logout
  ```

- Exiba uma lista abreviada de comandos:

  ```
  cf -h
  ```

- Exiba uma lista de todos os comandos:

  ```
  cf -h -a
  ```

IBM Cloud Private Cloud Foundry guia do desenvolvedor

Este guia contém os detalhes de uso do IBM® Cloud Private Cloud Foundry, como o uso de buildpacks e o trabalho com serviços.

- Configurar integrações com o IBM Cloud Private Cloud Foundry
- Usando buildpacks no IBM Cloud Private Cloud Foundry
- Criando uma instância de serviço

Configurando Integrações com o IBM® Cloud Private Cloud Foundry

As integrações a seguir estão disponíveis na IBM. Também é possível usar a estrutura de extensão para integrar suas próprias implementações com a implementação do IBM Cloud Private Cloud Foundry. Para obter mais informações sobre extensões, consulte Usando extensões no IBM® Cloud Private Cloud Foundry.

- Usando o Cloud Foundry App Autoscaler com o IBM® Cloud Private Cloud Foundry
- Aplicativos de Autoscaling no Cloud Foundry Enterprise Environment
- Integrando syslogs do IBM® Cloud Private Cloud Foundry com Splunk
- Configurar o Splunk Firehose Nozzle Release como um aplicativo do Cloud Foundry
- Conectando o IBM® Cloud Private Cloud Foundry ao Prometheus
- Configurando segmentos de isolamento no IBM® Cloud Private Cloud Foundry
- Usando os serviços do IBM Cloud no IBM Cloud Private Cloud Foundry
- Usando os serviços de banco de dados do IBM Cloud Private no IBM® Cloud Private Cloud Foundry
- Conectando-se ao Elastic Stack no IBM Cloud Private

Usando o CloudFoundry App Autoscaler com IBM® Cloud Private Cloud Foundry

IBM® Cloud Private Cloud Foundry: Essas informações são aplicáveis somente ao IBM Cloud Private Cloud Foundry.

É possível integrar o aplicativo IBM® Cloud Private Cloud Foundry com o App Autoscaler.

O App Autoscaler é um serviço de mercado de trabalho que pode ser usado para controlar o custo da execução de apps enquanto mantém o desempenho do app.

Para balancear o desempenho e o custo do app, os desenvolvedores de espaço podem usar o App Autoscaler para executar as tarefas a seguir:

- Configurar regras que ajustam contagens de instância com base em limites de métricas, como o uso da CPU.
- Modificar o número máximo e mínimo de instâncias para um aplicativo, manualmente ou seguindo um planejamento.

O App Autoscaler usa cinco liberações:

- `cfp-cf-release`
- `desvinculado`
- `postgres`
- `cf-routing-release`
- `app-autoscaler-release`

As liberações `cfp-cf-release` e `unbound` são fornecidas com o IBM Cloud Private Cloud Foundry. Deve-se fazer o download e mover as liberações `postgres` e `cf-routing-release` para os diretórios no contêiner `inception`. A liberação do App Autoscaler é incluída no diretório `app-autoscaler-release/src`.

Nota: essas instruções assumem que você ainda não tem um banco de dados configurado para ajuste automático de escala. Se você já tiver um banco de dados configurado, certifique-se de configurar o arquivo `manifests/db-stub-override.yml` para mapear os endereços e as tabelas para o banco de dados correto. Além disso, certifique-se de que o valor de `default_db` no arquivo `manifests/property-overrides.yml` esteja vinculado ao banco de dados correto.

Pré-requisitos para instalar o App Autoscaler

1. Instale as interfaces da linha de comandos do Bosh e do IBM Cloud Private Cloud Foundry. Para obter mais informações sobre essas interfaces da linha de comandos, consulte Interfaces da linha de comandos para o IBM® Cloud Private Cloud Foundry.
2. Assegure-se de que você tenha acesso à Internet quando fizer a extensão para que seja possível fazer o download das liberações `postgres` e `cf-routing-release`.
   - Faça download do `postgres` do Postgres v17.
   - Faça download do `cf-routing-release` do cf-routing-release v162.
3. Configure o endereço IP estático para seu banco de dados (a customização é necessária para suportar múltiplos bancos de dados).
   1. Determine os endereços IP estáticos a serem usados.
   2. Abra o `cloud-config.yml` e inclua os endereços IP estáticos nele.
   3. Ajuste a matriz `networks.subnets.reserved` para incluir o novo endereço IP estático, bem como **sete** endereços adicionais para as outras tarefas de escala de automático que são necessárias. No total, **oito** endereços IP são

necessários.
4. Reimplemente o arquivo `cloud-config.yml` para incluir os endereços IP atualizados executando o comando a seguir:

   ```
   bosh -e ENVIRONMENT_NAME update-cloud-config /data/CloudFoundry/cloud-config.yml
   ```

Instalando o App Autoscaler

Implemente o App Autoscaler por meio do Platform Configuration Manager como uma extensão. Para obter mais informações sobre extensões, consulte Usando extensões no IBM® Cloud Private Cloud Foundry

1. Clone o repositório Git que contém o código-fonte do App Autoscaler no host do contêiner de concepção. Execute o comando a seguir para clonar o repositório Git:

   ```
   git clone https://github.com/jnpacker/app-autoscaler-release.git
   ```

2. Em um terminal, mude para a pasta `app-autoscaler-release` nos arquivos clonados:

   ```
   cd app-autoscaler-release
   ```

3. Revise o arquivo `manifest/property-override.yml` e customize todos os valores de propriedade necessários. Confirme se as portas, os nomes de usuário e as senhas estão corretos. Revise e atualize as propriedades a seguir:

   - `cf_properties.api`
   - `default_db.address`
   - `service_broker_properties.uri`
   - `service_broker_properties.port`
   - `service_broker_properties.username`
   - `service_broker_properties.password`
   - `default_db.password`

4. Crie a extensão executando o comando a seguir em seu terminal:

   ```
   make extension
   ```

5. Copie o arquivo `app-autoscaler-extension.zip` no servidor que está executando o contêiner de instalação para IBM Cloud Private Cloud Foundry.

6. Abra um shell `bash` interativo no servidor.

7. No shell `bash`, registre a extensão. Deve-se estar no mesmo diretório que `install.sh`. Registre a extensão executando o comando a seguir:

   ```
   ./cm extension -e app-autoscaler register -p ./app-autoscaler-extension.zip
   ```

8. No shell `bash`, implemente a extensão executando o comando a seguir:

   ```
   ./cm extension -e app-autoscaler deploy
   ```

9. Confirme se todas as etapas têm o status `SUCCEEDED`. Execute o comando a seguir e revise a saída:

   ```
   ./cm states -e app-autoscaler
   ```

Registrando o App Autoscaler

1. Registre o serviço do App Autoscaler. Efetue login no {{site.data.keyword.cf_notm}} como o usuário administrativo e execute o comando a seguir:

   ```
   cf create-service-broker autoscaler <brokerUserName> <brokerPassword> <brokerURL>
   ```

   Nesse comando:

   - O `<brokerUserName>` é o nome de usuário para autenticação com o broker de serviço.
   - O `<brokerPassword>` é a senha para autenticação com o broker de serviço.
   - O `<brokerURL>` é a URL do broker de serviço.

   O valor padrão de `<brokerUserName>` é `username`, o valor padrão de `<brokerPassword>` é `password` e o valor padrão de `<brokerURL>` é `http://autoscalerservicebroker.YOUR_DOMAIN`.

Se você usou os valores padrão, execute o comando a seguir:

```
cf create-service-broker autoscaler username password
http://autoscalerservicebroker.YOUR_DOMAIN
```

2. Para usar o serviço para ajuste de escala automático de seus aplicativos, efetue login no {{site.data.keyword.cf_notm}} como o usuário administrativo e permita o acesso ao serviço para suas organizações. Execute o comando a seguir para ativar o acesso de serviço para todas as organizações ou para organizações específicas.

   ```
   cf enable-service-access autoscaler -o <org>
   ```

   Em que `<org>` é a organização ou organizações cujos membros podem usar o serviço.

3. Como um usuário com a função de desenvolvedor de espaço, crie a instância de serviço.

   ```
   cf create-service autoscaler  autoscaler-free-plan  <service_instance_name>
   ```

   Em que `<service_instance_name>` é o nome designado à instância de serviço.

Criando a política de ajuste automático de escala.

1. Crie o arquivo de políticas de ajuste automático de escala. O arquivo de política de ajuste automático de escala é um arquivo formatado em JSON que contém os parâmetros que definem as regras de ajuste automático de escala. O arquivo JSON é semelhante ao código a seguir:

```
{
 "instance_min_count": 1,
 "instance_max_count": 5,
 "scaling_rules": [{
   "metric_type": "memoryutil",
   "stat_window_secs": 300,
   "breach_duration_secs": 600,
   "threshold": 30,
   "operator": "<",
   "cool_down_secs": 300,
   "adjustment": "-1"
 }, {
   "metric_type": "memoryutil",
   "stat_window_secs": 300,
   "breach_duration_secs": 600,
   "threshold": 55,
   "operator": ">=",
   "cool_down_secs": 300,
   "adjustment": "+1"
 }],
 "schedules": {
   "timezone": "America/Los_Angeles",
   "recurring_schedule": [{
     "start_time": "01:00",
     "end_time": "23:00",
     "days_of_week": [
       1,
       2,
       3,
       4,
       5,
       6,
       7
     ],
     "instance_min_count": 1,
     "instance_max_count": 5,
     "initial_min_instance_count": 5
   }],
   "specific_date": [{
     "start_date_time": "2016-06-02T10:00",
     "end_date_time": "2018-06-15T13:59",
     "instance_min_count": 1,
     "instance_max_count": 4,
     "initial_min_instance_count": 1
   }]
 }
}
```

Esse arquivo contém várias seções. Nele, defina o valor `instance_min_count`, que é o número mínimo de instâncias e o `instance_max_count`, que é o número máximo de instâncias. Na seção `scaling_rules`, é possível definir múltiplos critérios de ajuste de escala. Na seção `schedules`, é possível definir planejamentos recorrentes com base no dia da semana ou planejamentos únicos para um dia ou intervalo de tempo específico. Embora seja possível fornecer múltiplas regras ou planejamentos de ajuste de escala, cada uma dessas definições deve conter todos os parâmetros especificados. A tabela a seguir fornece mais informações sobre estes parâmetros:

Parâmetro	Descrição	Valor
`instance_min_count`	O número mínimo de instâncias do aplicativo.	Número inteiro positivo
`instance_max_count`	O número máximo de instâncias do aplicativo.	Número inteiro positivo
`scaling_rules`	O conjunto de regras que definem quando novos aplicativos são criados ou removidos.	Matriz
`metric_type`		• `memoryused` • `memoryutil` • `responsetime` • `Rendimento do processamento`
`stat_window_secs`		Número inteiro positivo (em segundos)
`breach_duration_secs`		Número inteiro positivo (em segundos)
`threshold`	O limite para quando mudar o número de instâncias do aplicativo.	Número inteiro positivo
`operator`		• < • >
`cool_down_secs`		Número inteiro positivo (em segundos)
`adjustment`	A quantidade de mudança no número de instâncias. É possível aumentar e diminuir	

por um número de instâncias ou por um percentual de instâncias.|

- Números inteiros de -99 a -1 e 1 a 99
- Porcentagem positiva ou negativa

| | `schedules`|O conjunto de regras que definem como novos aplicativos são criados ou removidos em momentos diferentes.|Matriz| | `timezone`|O fuso horário do servidor.|| | `recurring_schedule`||| | `start_time`|O horário do dia em que as regras começam a ser aplicadas.|O tempo de 24 horas, tal como `01:00`| | `end_time`|O horário do dia em que as regras não são mais aplicadas.|O tempo de 24 horas, tal como `023:00`| | `days_of_week`|Os dias da semana aos quais o planejamento se aplica. Aqui, o domingo é representado como `1`.|`1-7`| | `initial_min_instance_count`|O número mínimo de instâncias no início do período de tempo.|Número inteiro positivo| | `specific_date`|A política de ajuste automático de escala a ser aplicada para um intervalo de tempo específico.|Matriz| | `start_date_time`|O horário para começar a aplicar essa política.|Data e hora no seguinte formato: `<four_digit_year>-<two_digit_month>-<two_digit_day>T<hour>:<minute>`, por exemplo, `2018-06-15T13:59`| | `end_date_time`|O tempo para terminar esta política.|Data e hora no seguinte formato: `<four_digit_year>-<two_digit_month>-<two_digit_day>T<hour>:<minute>`, por exemplo, `2018-06-15T13:59`|

1. Como um usuário com a função de desenvolvedor de espaço, ligue um aplicativo existente à instância de serviço.

   ```
   cf bind-service <app_name> <service_instance_name> -c <policy>
   ```

   Nesse comando:

   - O `<app_name>` é o nome do aplicativo para escalar automaticamente.
   - O `<service_instance_name>` é o nome da instância de serviço para o App Autoscaler.
   - O `<policy>` é o caminho para o arquivo de políticas a ser aplicado, como `/data/extension/custom/app-autoscaler/example/example-policy.json`.

   Agora, seu serviço é ligado ao seu aplicativo e é escalado automaticamente para os requisitos que são definidos no arquivo `policy.json`.

Removendo o serviço

Conclua as etapas a seguir para remover o serviço.

1. Efetuar login no {{site.data.keyword.cf_notm}} como o usuário administrativo.
2. Execute os comandos a seguir:

```
cf purge-service-offering autoscaler ; \
cf delete-service-broker autoscaler
```

Resolução de problemas do App Autoscaler

Falha ao criar uma liberação

Sintomas

Ao implementar o App Autoscaler, a mensagem de erro a seguir será exibida:

```
Building a release from directory '/data/extensions/custom/app-autoscaler':
  Compressing staging directory:
    Shelling out to tar:
      Running command: 'tar czf /data/home/.bosh/tmp/bosh-platform-disk-TarballCompressor-
CompressSpecificFilesInDir114079115 -C /data/home/.bosh/tmp/bosh-resource-archive204470236 .',
stdout: '', stderr: '
gzip: stdout: No space left on device
':
        signal: broken pipe

Exit code 1
state:Create Release
```

Causas

O diretório `/data/home/.bosh/tmp` está cheio.

Resolvendo o problema

Limpe o diretório temporário BOSH executando o comando a seguir:

```
rm -rf /data/home/.bosh/tmp
```

Falha ao acessar uma URL durante a criação de liberação

Sintomas

Quando você implementa o App Autoscaler, a mensagem de erro a seguir é exibida:
`org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute INFO: Retrying request to {s}->https://repo.spring.io:443`

Causas

O contêiner de concepção tem problemas para se conectar a alguns repositórios seguros.

Resolvendo o problema

1. Abra o arquivo `pom.xml` na pasta `/data/extensions/custom/app-autoscaler/src/app-autoscaler/scheduler/`.
2. Mude todas as referências de `https` para `http`.
3. Execute o comando de implementação novamente:

```
/opt/ibm/cloud/bin/cm extension -e app-autoscaler deploy
```

Usando buildpacks no IBM Cloud Private Cloud Foundry

Por padrão, o IBM® Cloud Private Cloud Foundry contém buildpacks que são suportados pelo IBM. É possível usar os buildpacks para implementar seus aplicativos na nuvem.

Sobre os buildpacks

Os buildpacks do Cloud Foundry fornecem o suporte de tempo de execução para aplicativos no ambiente do Cloud Foundry. Ao implementar um aplicativo na nuvem, ele inicia um buildpack que suporta seu tipo de aplicativo. O IBM Cloud Private Cloud Foundry fornece suporte de buildpack para Java™ EE, Node.js, ASP.Net, Swift e outros tipos de aplicativos.

É possível usar os buildpacks que são incluídos com o IBM Cloud Private Cloud Foundry para implementar aplicativos e ligá-los aos serviços fornecidos pelo usuário. Os buildpacks do Cloud Foundry a seguir estão disponíveis para o IBM Cloud e o IBM Cloud Private. Os buildpacks no IBM Cloud Private trabalham de forma diferente do que no IBM Cloud, dependendo de como o Cloud Foundry está configurado.

`cflinuxfs3` é a pilha padrão para uso com buildpacks. `cflinuxfs3` cria um ambiente de aplicativos quando combinado com um buildpack baseado em Ubuntu Xenial

Os buildpacks a seguir não funcionam com o `cflinuxfs3`:

1. Swift
2. dotNET
3. Contêineres de Docker

Buildpacks disponíveis

É possível usar o Liberty for Java, o SDK for Node.js, o Runtime for Swift ou buildpacks de núcleo ASP.NET para implementar seus aplicativos na nuvem com o procedimento a seguir. Ou, é possível começar a trabalhar com serviços fornecidos pelo usuário usando os aplicativos iniciadores para os buildpacks a seguir:

- Liberty for Java
- SDK for Node.js

Instalando e atualizando buildpacks

Os buildpacks são incluídos com o IBM Cloud Private Cloud Foundry. Para obter mais informações sobre a instalação e as atualizações do buildpack, consulte Administrando buildpacks no IBM Cloud Private Cloud Foundry.

Procedimento geral para implementação de um aplicativo com buildpacks

Implemente um aplicativo para sua instância do IBM Cloud Private Cloud Foundry com buildpacks

1. Aponte sua CLI do Cloud Foundry para seu ambiente usando o terminal da API ou a URL de destino para sua instância do IBM Cloud Private Cloud Foundry.

   ```
   cf api <API endpoint of IBM Cloud Private Cloud Foundry>
   ```

   Nota: para obter mais informações sobre como identificar seu terminal de API ou URL de destino, consulte Usando `bluemix_env_domain` para localizar o terminal de API e o terminal de Login.

2. Use a linha de comandos cf para efetuar login no Cloud Foundry.

   ```
   cf login
   ```

3. De dentro do diretório de seu aplicativo, envie o aplicativo por push.

   ```
   cf push
   ```

4. A implementação de seu aplicativo pode levar alguns minutos. Quando a implementação for concluída, você verá uma mensagem de que o seu aplicativo está em execução. Visualize seu aplicativo na URL que é listada na saída do comando push ou visualize o status de implementação do aplicativo e a URL executando o comando a seguir:

   ```
   cf app <Your-App-Name>
   ```

Dica: É possível resolver problemas de erros no processo de implementação usando o comando `cf logs <Your-App-Name> --recent`.

Variáveis de ambiente para buildpacks no IBM Cloud Private Cloud Foundry

Como é possível configurar o IBM Cloud Private Cloud Foundry de forma diferente em comparação com o IBM Cloud, algumas variáveis de ambiente podem funcionar de forma diferente e requerem que você as configure manualmente.

Usando o `bluemix_env_domain` para localizar seu terminal de API e o terminal de

login Ao instalar o Cloud Foundry, o administrador define a variável `bluemix_env_domain`. Essa variável é usada para definir o terminal de API ou a URL de destino e pode ser adaptada para definir o terminal de login para sua instância do IBM Cloud Private Cloud Foundry.

Por exemplo, se seu `bluemix_env_domain` estivesse definido como `local.bluemix.net`, sua URL de destino ou terminal de API seria `https://api.local.bluemix.net`. O terminal de login para seu domínio de ambiente seria `https://login.local.bluemix.net/UAALoginServerWAR`. Uma URL de terminal de login sempre inclui `login.` antes de `bluemix_env_domain` e termina com `/UAALoginServerWAR`. Um terminal de API ou URL de destino inclui `api.` antes de `bluemix_env_domain`.

Para saber mais sobre como instalar o Cloud Foundry e configurar o IBM Cloud Private, consulte Visão geral de instalação da plataforma do IBM Cloud Private Cloud Foundry.

PORT

Executando aplicativos Node.js localmente antes de implementar o IBM Cloud Private Cloud Foundry

Antes de implementar seu aplicativo ou ao depurar seu aplicativo, talvez você queira executá-lo localmente. Ao usar um buildpack, o ambiente do Cloud Foundry aloca uma porta e transmite essas informações ao aplicativo na variável de ambiente `PORT`. Para executar o aplicativo no Cloud Foundry, o aplicativo deve ser codificado para ligar-se à variável de ambiente `PORT`. No entanto, quando você executa um aplicativo localmente, essa variável de ambiente pode não ser configurada. É necessário codificar seu aplicativo para ligar-se a uma porta padrão se ele não localizar a variável de ambiente `PORT`.

Ao trabalhar com o aplicativo iniciador Node.js, você configura a variável de ambiente `PORT` como `3000`. Para obter mais informações sobre como executar aplicativos Node.js localmente, consulte o tutorial *Trabalhando com serviços fornecidos pelo usuário e o aplicativo iniciador Node.js* e a documentação do IBM Cloud sobre como Executar o aplicativo Node.js localmente.

Links relacionados

Para saber mais sobre serviços fornecidos pelo usuário, serviços fornecidos pelo broker de serviço, buildpacks de administração e aplicativos iniciadores para o IBM Cloud Private Cloud Foundry, consulte os tópicos a seguir:

- Administrando buildpacks no IBM Cloud Private Cloud Foundry
- Trabalhando com serviços fornecidos pelo usuário no IBM Cloud Private Cloud Foundry
- Gerenciando aplicativos Liberty e Node.js no IBM Cloud Private Cloud Foundry
- Trabalhando com serviços fornecidos por um broker de serviço no IBM Cloud Private Cloud Foundry

Consulte os links a seguir para saber mais sobre os buildpacks no IBM Cloud:

- Liberty for Java
- SDK for Node.js
- IBM Cloud Runtime for Swift
- ASP.NET Core

Administrando buildpacks no IBM Cloud Private Cloud Foundry

Dependendo de como o IBM® Cloud Private foi configurado, os buildpacks que estão contidos no IBM Cloud Private Cloud Foundry poderão requerer gerenciamento de atualização, exclusão de buildpacks descontinuados e outra manutenção.

Manutenção e ciclo de vida do buildpack

Como proprietário, é possível escolher como deseja ordenar e organizar seus buildpacks. Cada versão de buildpack requer memória e, à medida que você atualiza seus buildpacks, deve-se excluir buildpacks não utilizados ou descontinuados de seu ambiente.

Para saber mais sobre o ciclo de vida do buildpack, incluindo como criar, atualizar e excluir buildpacks no IBM Cloud Private Cloud Foundry, consulte os recursos de Buildpacks do Cloud Foundry.

Dica: se você tiver um sistema de nomenclatura consistente que é usado para todos os seus buildpacks, ele ajudará a controlar as versões que são instaladas. Se for necessário suporte para seu buildpacks, você poderá ajudar a diagnosticar e corrigir quaisquer problemas com um sistema de nomenclatura que inclua o idioma e a versão de seus buildpacks. É possível optar por ter o idioma e a versão no próprio nome do buildpack ou no nome do arquivo do buildpack.

Instalando buildpacks e atualizações buildpack

Cada liberação do IBM Cloud Private Cloud Foundry contém buildpacks do IBM. Quando você atualiza o IBM Cloud Private Cloud Foundry periodicamente, quaisquer atualizações nos componentes do Cloud Foundry também incluem atualizações para os buildpacks. Depois de instalar o IBM Cloud Private Cloud Foundry ou cada vez que você fizer atualizações no Cloud Foundry, novos buildpacks podem ser instalados.

Consulte Instalando o IBM Cloud Private Cloud Foundry para saber mais sobre a instalação do Cloud Foundry.

Removendo buildpacks

Como os buildpacks não são excluídos automaticamente, você decide quando removê-los. Deve-se remover periodicamente os buildpacks para que o Cloud Foundry não fique sem espaço de memória.

Remova buildpacks com as etapas a seguir:

1. Na CLI do Cloud Foundry, liste os buildpacks.

   ```
   cf buildpacks
   ```

2. Selecione qual buildpack remover e, em seguida, use o comando a seguir para remover um buildpack.

   ```
   cf delete-buildpack <buildpack_name>
   ```

Buildpacks no modo off-line

Os buildpacks do Liberty for Java™ e Node.js podem acessar sites e fazer download de artefatos de origens que são externas para o ambiente de nuvem no qual elas operam. Para saber mais sobre sites de lista de aplicativos confiáveis, consulte Modo off-line para Liberty nos ambientes IBM Cloud Dedicated, Bluemix Local e IBM Cloud Private. Além disso, para obter mais informações sobre como executar aplicativos Node.js localmente, é possível explorar Trabalhando off-line com Node.js.

Trabalhando com serviços fornecidos pelo usuário no IBM® Cloud Private Cloud Foundry

O IBM Cloud Private Cloud Foundry fornece um mecanismo para você se conectar e usar Serviços fornecidos pelo usuário, que são serviços que não podem ser fornecidos ou disponibilizados em seu ambiente de nuvem.

Introdução

Para usar um serviço fornecido pelo usuário com seu aplicativo, as contas e as ferramentas a seguir são necessárias:

- Um serviço disponível fora de seu ambiente
- Credenciais para acessar ou usar esse serviço
- Um aplicativo que pode ligar-se a esse serviço

Com os serviços fornecidos pelo usuário no Cloud Foundry, é possível ligar seus aplicativos a serviços fora de seu mercado do ambiente do IBM Cloud Private usando os buildpacks do Liberty for Java™, do SDK for Node.js, do Runtime for Swift ou do ASP.NET Core.

Como usar as guias

O procedimento a seguir fornece o processo geral para você criar um serviço externo, implementar seu aplicativo e ligar seu aplicativo ao seu serviço externo usando os buildpacks do Cloud Foundry. Se você tiver as credenciais de serviço, será possível adaptar este procedimento para seus próprios aplicativos e para qualquer serviço que esteja hospedado em uma nuvem pública ou no IBM Cloud Private para criar um serviço fornecido pelo usuário. O Cloud Foundry fornece as opções para criar seu serviço e

fornecer credenciais para ele. Para obter mais informações sobre como customizar os serviços fornecidos pelo usuário para seu ambiente de nuvem, consulte a documentação do Cloud Foundry em Instâncias de serviço fornecidas pelo usuário.

Também é possível usar os aplicativos iniciadores para Liberty ou Node.js, que fornecem exemplos específicos de configuração de serviços fornecidos pelo usuário em um ambiente de nuvem. Os procedimentos mostram como criar um serviço fornecido pelo usuário com um aplicativo iniciador e um serviço do Cloudant® NoSQL DB no IBM Cloud.

Antes de iniciar

Dica: Se você estiver trabalhando sem acesso à Internet externo, leia completamente o procedimento antes de começar a trabalhar com buildpacks. Assegure-se de que você tenha acesso a todas as documentações e recursos necessários.

Etapa 1: criar um serviço externo

É possível ligar-se a qualquer serviço fora de seu ambiente se você tiver as credenciais necessárias. Por exemplo, é possível criar uma instância de serviço no IBM Cloud e, em seguida, implementar um aplicativo no IBM Cloud Private Cloud Foundry que liga ao seu serviço. do IBM Cloud

Etapa 2: salvar as credenciais de serviço

Depois de criar sua instância de serviço, salve as credenciais necessárias para ligar seu aplicativo. É possível salvar essas credenciais em um arquivo JSON. Embora serviços diferentes possam ter requisitos e formatos de credenciais específicos, o modelo de exemplo a seguir mostra as credenciais de que você precisa para um Cloudant NoSQL DB.

```
{
  "username": "<username>",
    "password": "<password>",
    "host": "<host.dns.name>",
    "port": <port>,
    "url": "https://<username>:<password>@<host.dns.name>"
  }
```

Etapa 3: criar um serviço fornecido pelo usuário

É possível usar o comando `cf cups` do Cloud Foundry para criar um serviço fornecido pelo usuário.

```
cf create-user-provided service <service name> -p <path to JSON file>
```

Nota: o Cloud Foundry fornece informações sobre outros métodos para criar serviços fornecidos pelo usuário que você pode acessar por meio da linha de comandos usando o comando `cf create-user-provided-service`.

Etapa 4: ligar seu aplicativo a um serviço fornecido pelo usuário

Ligue o serviço fornecido pelo usuário ao aplicativo usando a linha de comandos.

```
cf bind-service <app-name> <service-name>
```

Em seguida, remonte seu aplicativo.

```
cf restage <app-name>
```

Após a remontagem, será possível confirmar se seu aplicativo funciona conforme o esperado procurando a URL de seu aplicativo. Ou, será possível usar o comando `cf service` para visualizar os serviços e aplicativos de limite.

Exemplos e informações de serviços fornecidos pelo usuário no IBM Cloud Private Cloud Foundry

Os aplicativos iniciadores de buildpack o guiam através da criação de um serviço fornecido pelo usuário, da criação de um banco de dadosCloudant no IBM Cloud e da ligação de um aplicativo em seu serviço.

Tente criar serviços fornecidos pelo usuário e aplicativos de ligação:

- Trabalhando com serviços fornecidos pelo usuário e o aplicativo iniciador do Liberty
- Trabalhando com serviços fornecidos pelo usuário e o aplicativo iniciador Node.js

Trabalhando com serviços fornecidos pelo usuário e o aplicativo iniciador do Liberty

É possível ligar os seus aplicativos em IBM® Cloud Private Cloud Foundry a serviços que existem fora do seu ambiente do IBM Cloud Private Cloud Foundry.

O Cloud Foundry fornece um mecanismo para conectar serviços e aplicativos que não podem ser fornecidos por ou disponíveis dentro de sua instância da nuvem. Para saber mais sobre os recursos do Cloud Foundry, consulte Serviços fornecidos pelo usuário.

Esse exemplo usa uma instância do Cloudant® NoSQL DB e o orienta através do processo para preparar e implementar seu aplicativo, criar um serviço fornecido pelo usuário e conectar-se a um banco de dados do Cloudant usando o aplicativo de introdução do Liberty for Java™.

Antes de iniciar

São necessárias as contas e ferramentas a seguir:

- Acesso ao ambiente do IBM Cloud Private Cloud Foundry
- Conta do IBM Cloud
- Cloud Foundry CLI
- Git
- Maven

Etapa 1: clonar o aplicativo de amostra

Primeiro, clone o repositório GitHub do app de amostra.

```
git clone https://github.com/IBM-Bluemix/get-started-java
```

Etapa 2: Executar o aplicativo localmente usando a linha de comandos

Use Maven para construir seu código-fonte e executar o aplicativo.

1. Na linha de comandos, mude o diretório para onde o app de amostra está localizado.

   ```
   cd get-started-java
   ```

2. Use Maven para instalar as dependências e construir o arquivo .war.

   ```
   mvn clean install
   ```

3. Execute o app localmente no Liberty.

   ```
   mvn install liberty:run-server
   ```

Ao ver a mensagem: `O servidor defaultServer está pronto para executar um planeta mais inteligente`, será possível visualizar seu aplicativo em: `http://localhost:9080/GetStartedJava`.

Para parar seu app, pressione Ctrl-C na janela de linha de comandos na qual você iniciou o app.

Etapa 3: preparar o app para implementação

Para implementá-lo no IBM Cloud Private, pode ser útil configurar um arquivo `manifest.yml`. O arquivo `manifest.yml` inclui informações básicas sobre seu aplicativo, como o nome, quanta memória alocar para cada instância e a rota. É possível localizar um arquivo `manifest.yml` de amostra no diretório `get-started-java`.

Abra o arquivo `manifest.yml` e mude o `name` de `GetStartedJava` para o nome do app, `<var class="keyword varname" data-hd-keyref="app_name">app_name</var>`.

```
applications:
 - name: GetStartedJava
   random-route: true
   path: target/GetStartedJava.war
```

```
memory: 512M
instances: 1
```

Dica: nesse arquivo `manifest.yml`, `random-route: true` gera uma rota aleatória para seu app para evitar que sua rota colida com outras. Se você escolher isso, será possível substituir `random-route: true` por `host: myChosenHostName` e fornecer um nome do host de sua opção.

Etapa 4: implementar no IBM Cloud Private Cloud Foundry

1. Implemente seu app em sua instância do IBM Cloud Private Cloud Foundry usando sua URL do IBM Cloud Private Cloud Foundry.

   ```
   cf api <API endpoint of IBM Cloud Private Cloud Foundry>
   ```

2. Efetue login em sua conta do IBM Cloud Private Cloud Foundry.

   ```
   cf login
   ```

3. No diretório `get-started-java`, envie seu aplicativo por push para o IBM Cloud Private.

   ```
   cf push
   ```

4. A implementação de seu aplicativo pode levar alguns minutos. Quando a implementação for concluída, você verá uma mensagem de que o seu aplicativo está em execução. Visualize seu aplicativo na URL que é listada na saída do comando push ou visualize o status de implementação do aplicativo e a URL executando o comando a seguir:

   ```
   cf apps
   ```

Dica: É possível resolver problemas de erros no processo de implementação usando o comando `cf logs <Your-App-Name> --recent`.

Etapa 5: criar uma instância de serviço do Cloudant NoSQL DB

Em seguida, você inclui um banco de dados do Cloudant NoSQL DB no IBM Cloud para esse aplicativo e configura o aplicativo para executar localmente e no IBM Cloud Private. Você precisa criar o Cloudant banco de dados no IBM Cloud. Em seguida, você vincula o seu aplicativo iniciador ao serviço do IBM Cloud de dentro do IBM Cloud Private Cloud Foundry.

1. Em seu navegador, efetue login no IBM Cloud e acesse o Catalog.
2. Na seção **Dados & Analytics**, selecione o **Cloudant NoSQL DB** e, em seguida, crie o serviço.
3. Acesse **Credenciais de serviço** e visualize as credenciais para o serviço.
4. Salve suas credenciais do Cloudant NoSQL DB em um arquivo JSON.

   ```
   {
    "username": "<username>",
    "password": "<password>",
    "host": "<host.dns.name>",
    "port": <port>,
    "url": "https://<username>:<password>@<host.dns.name>"
   }
   ```

5. Para trabalhar com o aplicativo de introdução, deve-se incluir `cloudant` no **nome do serviço** fornecido pelo usuário que você cria. O aplicativo de introdução analisa a variável de serviços do VCAP e procura por um serviço fornecido pelo usuário com um nome que contém `cloudant`, como `my-cloudantNoSQLDB-ups`.

   Use o comando `create-user-provided-service` para criar o serviço fornecido pelo usuário. Especifique o **nome do serviço** e o caminho para o arquivo `cloudant-credentials.json` que você criou.

   ```
   cf create-user-provided-service <Your-Service-Name> -p <path to json file>
   ```

Etapa 6: usar o Cloudant NoSQL DB

1. Ligue o serviço fornecido pelo usuário ao aplicativo de introdução.

   ```
   cf bs <Your-App-Name> <Your-Service-Name>
   ```

2. Remonte o aplicativo.

   ```
   cf restage <Your-App-Name>
   ```

Dica: é possível usar variáveis de ambiente para separar as configurações de implementação de seu código de origem. Por exemplo, em vez de codificar permanentemente uma senha do banco de dados, é possível armazená-la em uma variável de ambiente que você referenciar em seu código-fonte.

Etapa 7: Use o Cloudant NoSQL DB localmente

Em seguida, você atualiza o código local para apontar para esse banco de dados. É possível armazenar as credenciais para os serviços em um arquivo de propriedades. Esse arquivo é usado apenas quando o aplicativo está em execução localmente. Ao executar o aplicativo no IBM Cloud Private Cloud Foundry, as credenciais são lidas na variável de ambiente `VCAP_SERVICES`.

1. Abra o arquivo `src/main/resources/cloudant.properties`:

   ```
   cloudant_url=
   ```

2. Copie e cole o valor da `url` das **Credenciais de serviço**, que você salvou em um arquivo na Etapa 5, para o campo `url` do arquivo `cloudant.properties`. Salve as alterações.

   ```
   cloudant_url=https://123456789 ... bluemix.cloudant.com
   ```

3. Pare o servidor Liberty local e, em seguida, no diretório `get-started-java`, reinicie-o com o comando a seguir:

   ```
   mvn install liberty:run-server
   ```

Etapa 8: confirmar

Atualize a visualização do navegador em `http://localhost: 9080/GetStartedJava/`. Quaisquer nomes que você inserir no aplicativo serão incluídos no banco de dados.

Seu aplicativo local e o aplicativo IBM Cloud Private Cloud Foundry compartilham o banco de dados do Cloudant NoSQL DB. Nomes que você inclui de qualquer aplicativo aparecerão em ambos quando você atualizar os navegadores.

Trabalhando com serviços fornecidos pelo usuário e o aplicativo iniciador Node.js

É possível ligar os seus aplicativos em IBM® Cloud Private Cloud Foundry a serviços que existem fora do seu ambiente do IBM Cloud Private Cloud Foundry.

O Cloud Foundry fornece um mecanismo para conectar serviços e aplicativos que não podem ser fornecidos por ou disponíveis dentro de sua instância da nuvem. Para saber mais sobre os recursos do Cloud Foundry, consulte Serviços fornecidos pelo usuário.

Este exemplo usa uma instância do Cloudant® NoSQL DB e o orienta por meio do processo para preparar e implementar seu aplicativo, criar um serviço fornecido pelo usuário e conectar-se a um banco de dados do Cloudant usando o aplicativo Node.js sendo iniciado.

Antes de iniciar

São necessárias as contas e ferramentas a seguir:

- Acesso ao ambiente do IBM Cloud Private Cloud Foundry
- Conta do IBM Cloud
- Cloud Foundry CLI
- Git
- Node.js

Prepare e implemente seu aplicativo

Etapa 1: clonar o aplicativo de amostra

Primeiro, clone o repositório GitHub do app de amostra.

```
git clone https://github.com/IBM-Bluemix/get-started-node
```

Etapa 2: Executar o aplicativo localmente usando a linha de comandos

1. Na linha de comandos, mude o diretório para onde o app de amostra está localizado.

   ```
   cd get-started-node
   ```

2. Instale as dependências que estão listadas no arquivo package.json para executar o app localmente.

   ```
   npm install
   ```

3. Execute o aplicativo.

   ```
   npm start
   ```

É possível visualizar seu app em `http://localhost:3000`.

Dica: use nodemon para reinicialização automática do aplicativo com mudanças de arquivo.

Etapa 3: preparar o app para implementação

Para implementar seu aplicativo no IBM Cloud Private, pode ser útil configurar um arquivo `manifest.yml`. O arquivo `manifest.yml` inclui informações básicas sobre seu aplicativo, como o nome, quanta memória alocar para cada instância e a rota. É possível localizar um arquivo `manifest.yml` de amostra no diretório `get-started-node`.

Abra o arquivo `manifest.yml` e mude o `name` de `GetStartedNode` para o nome do app, `<var class="keyword varname" data-hd-keyref="app_name">app_name</var>`.

```
applications:
- name: GetStartedNode
  random-route: true
  memory: 128M
```

Dica: nesse arquivo `manifest.yml`, `random-route: true` gera uma rota aleatória para seu app para evitar que sua rota colida com outras. Se você escolher isso, será possível substituir `random-route: true` por `host: myChosenHostName` e fornecer um nome do host de sua opção.

Etapa 4: implementar no IBM Cloud Private Cloud Foundry

1. Implemente o seu aplicativo IBM Cloud Private Cloud Foundry usando a URL do IBM Cloud Private Cloud Foundry.

   ```
   cf api <url of IBM Cloud Private Cloud Foundry>
   ```

2. Efetue login em sua conta do IBM Cloud Private Cloud Foundry.

   ```
   cf login
   ```

3. No diretório `get-started-node`, envie seu aplicativo por push para o IBM Cloud Private.

   ```
   cf push
   ```

4. A implementação de seu aplicativo pode levar alguns minutos. Quando a implementação for concluída, você verá uma mensagem de que o seu aplicativo está em execução. Visualize o app na URL listada na saída do comando push ou visualize o status de implementação do app e a URL, executando o comando a seguir:

   ```
   cf apps
   ```

Dica: É possível resolver problemas de erros no processo de implementação usando o comando `cf logs <Your-App-Name> --recent`.

Etapa 5: incluir um banco de dados

Em seguida, você inclui um banco de dados Cloudant NoSQL DB no IBM Cloud para seu aplicativo e configura o aplicativo para executar localmente e no IBM Cloud Private. Você precisa criar o Cloudant banco de dados no IBM Cloud. Em seguida, você vincula o seu aplicativo iniciador ao serviço do IBM Cloud de dentro do IBM Cloud Private Cloud Foundry.

1. Em seu navegador, efetue login no IBM Cloud e acesse o Catalog.
2. Na seção **Dados & Analytics**, selecione o **Cloudant NoSQL DB** e, em seguida, crie o serviço.
3. Acesse **Credenciais de serviço** e visualize as credenciais para o serviço.

4. Após a instância do Cloudant ser criada, deve-se salvar as credenciais para ligar seu aplicativo a ela. Salve suas credenciais do Cloudant NoSQL DB em um arquivo JSON, com um nome de arquivo como `cloudant_credentials.json`.

```
{
 "username": "<username>",
 "password": "<password>",
 "host": "<host.dns.name>",
 "port": <port>,
 "url": "https://<username>:<password>@<host.dns.name>"
}
```

Opcional: execute o aplicativo localmente

Para executar o aplicativo localmente, é necessário atualizar seu código local para apontar para esse banco de dados. Crie um arquivo JSON para armazenar as credenciais para os serviços que o aplicativo usa. Esse arquivo é usado apenas quando o aplicativo está em execução localmente. Quando o aplicativo está em execução na nuvem, as credenciais são lidas a partir da variável de ambiente `VCAP_SERVICES`.

Dica: esse arquivo JSON `vcap-local.json` não é o mesmo que o arquivo `cloudant-credentials.json`. O arquivo `vcap-local.json` é usado apenas para executar seu aplicativo localmente.

1. No diretório `get-started-node`, crie um arquivo que seja chamado `vcap-local.json` com o conteúdo a seguir:

```
{
 "services": {
   "cloudantNoSQLDB": [
     {
       "credentials": {
         "url":"CLOUDANT_DATABASE_URL"
       },
       "label": "cloudantNoSQLDB"
     }
   ]
 }
}
```

   Dica: é possível usar variáveis de ambiente para separar as configurações de implementação de seu código de origem. Por exemplo, em vez de codificar permanentemente uma senha do banco de dados, é possível armazená-la em uma variável de ambiente que você referenciar em seu código-fonte.

2. No arquivo `cloudant-credentials.json` que você salvou na etapa 5 ou nas **Credenciais de serviço** na etapa 5, copie e cole apenas a `url` das credenciais para o campo `url` do arquivo `vcap-local.json`, substituindo **CLOUDANT_DATABASE_URL**.

3. Pare o aplicativo local e, em seguida, reinicie-o.

```
npm start
```

   Visualize seu aplicativo local em `http://localhost: 3000`. Quaisquer nomes que você inserir no aplicativo serão incluídos no banco de dados.

Dica: o IBM Cloud define a variável de ambiente PORT quando seu app é executado na nuvem. Quando você executa o aplicativo localmente, a variável de ambiente PORT não está definida, portanto, 3000 é usado como o número da porta. Para obter mais informações sobre como executar aplicativos localmente, consulte Execute seu aplicativo localmente.

Etapa 6: criar um serviço fornecido pelo usuário

Para trabalhar com o aplicativo de introdução, deve-se incluir `cloudant` no **nome do serviço** fornecido pelo usuário que você cria. O aplicativo de introdução analisa a variável de serviços do VCAP e procura por um serviço fornecido pelo usuário com um nome que contém `cloudant`, como `my-cloudantNoSQLDB-ups`.

Use o comando `create-user-provided-service` para criar o serviço fornecido pelo usuário. Especifique o **nome do serviço** e o caminho para o arquivo `cloudant-credentials.json` que você criou na etapa 5.

```
cf create-user-provided-service <service name> -p <path to json file>
```

Etapa 7: ligar o serviço fornecido pelo usuário e reorganizar o aplicativo

Ligue o serviço fornecido pelo usuário ao aplicativo de introdução e, em seguida, prepare-o.

```
cf bs <Your-App-Name> <Your-Service-Name>
cf restage <Your-App-Name>
```

Etapa 8: confirmar

Navegue até seu aplicativo e confirme se você pode incluir vários nomes no campo `nome` de seu aplicativo inicial.

Seu aplicativo local e o aplicativo IBM Cloud Private Cloud Foundry compartilham o banco de dados Cloudant. Nomes que você inclui de qualquer aplicativo aparecerão em ambos quando você atualizar os navegadores.

Trabalhando com serviços fornecidos por um broker de serviço no

IBM Cloud Private Cloud Foundry

Os brokers de serviço divulgam um catálogo de ofertas de serviços e planos para o mercado e agem de acordo com solicitações do mercado para fornecer, ligar, desvincular e desaprovisionar serviços.

IBM® Cloud Private Cloud Foundry fornece um mecanismo para você registrar um broker de serviço e usar serviços que possam não ser fornecidos ou disponibilizados em seu ambiente de nuvem.

Introdução

Para usar um serviço fornecido pelo broker de serviço com seu aplicativo, deve-se concluir as tarefas a seguir:

- Crie um broker de serviço disponível que esteja fora de seu ambiente. Consulte Implementando o banco de dados do Open Service Broker (OSB) no IBM Cloud Private Cloud Foundry.
- Registre o broker de serviço em seu ambiente. Consulte Usando serviços de banco de dados do IBM Cloud Private no IBM Cloud Private Cloud Foundry .
- Crie um aplicativo que possa ser ligado ao serviço que o broker de serviço provisiona.

Com os serviços fornecidos pelo broker de serviço no Cloud Foundry, é possível ligar seus aplicativos a serviços que estejam fora do mercado de trabalho do ambiente do IBM Cloud Private usando os buildpacks Liberty for Java™, SDK for Node.js e Runtime for Swift.

Como usar as guias

O procedimento a seguir fornece o processo geral para criação de uma instância de serviço por meio de um broker de serviço, implementação de seu aplicativo e ligação de seu aplicativo à instância de serviço usando os buildpacks do Cloud Foundry. Se você tiver outros brokers de serviço, é possível adaptar este procedimento para seus próprios aplicativos e para qualquer serviço que seja fornecido por um broker de serviço em uma nuvem pública ou no IBM Cloud Private.

Para obter mais informações sobre como gerenciar brokers de serviço em seu ambiente do Cloud Foundry, consulte a documentação do Cloud Foundry em Gerenciando brokers de serviço.

Também é possível usar os aplicativos iniciadores para Liberty ou Node.js, que fornecem exemplos específicos de configuração de serviços fornecidos pelo broker de serviço em um ambiente do IBM Cloud Private. Os procedimentos mostram como registrar um broker de serviço, provisionar uma instância de serviço do Mongo® DB e ligar a instância de serviço a um aplicativo iniciador.

Antes de iniciar

Dica: Se você estiver trabalhando sem acesso à Internet externo, leia completamente o procedimento antes de começar a trabalhar com buildpacks. Assegure-se de que você tenha acesso a todas as documentações e recursos necessários.

Etapa 1: Implementar um broker de serviço

Implemente um broker de serviço em um ambiente externo. Consulte Implementando o banco de dados do Open Service Broker (OSB) no IBM Cloud Private Cloud Foundry para implementar o broker de serviço que o IBM Cloud Private Cloud Foundry oferece para fornecer serviços de banco de dados a partir de um ambiente do IBM Cloud Private.

Etapa 2: Registrar um broker de serviço

É possível usar o comando `cf create-service-broker` do Cloud Foundry para registrar um broker de serviço.

```
$ cf create-service-broker SERVICE-BROKER-NAME USER PASSWORD SERVICE-BROKER-URL
```

Etapa 3: tornar ofertas de serviços e planos públicos

É possível usar o comando `cf service-access` do Cloud Foundry para ver as configurações de acesso nos serviços no mercado de trabalho.

```
$cf service-access
```

É possível usar o comando `cf enable-service-access` do Cloud Foundry para ativar o acesso a um serviço.

```
$cf enable-service-access SERVICE-NAME
```

Etapa 4: criar uma instância de serviço

Crie uma instância de serviço usando o comando `cf create-service`.

```
$cf create-service SERVICE-NAME PLAN SERVICE-INSTANCE-NAME
```

Etapa 5: Ligar seu aplicativo a uma instância de serviço

Ligue o aplicativo a uma instância de serviço usando o comando `cf bind-service`.

```
$cf bind-service APP-NAME SERVICE-INSTANCE-NAME
```

Exemplos e informações de broker de serviço no IBM Cloud Private

Para criar serviços e aplicativos de ligação, consulte os tópicos a seguir:

- Trabalhando com um broker de serviço e com o app iniciador do Liberty
- Trabalhando com um broker de serviço e com o app iniciador do Node.js

Trabalhando com um broker de serviço e com o app Liberty Starter

É possível ligar seus aplicativos no IBM® Cloud Private Cloud Foundry aos serviços que existem fora de seu ambiente do IBM Cloud Private Cloud Foundry usando o aplicativo de introdução do Liberty for Java™.

Os brokers de serviço divulgam um catálogo de ofertas de serviços e planos para o mercado e agem de acordo com solicitações do mercado para fornecer, ligar, desvincular e desaprovisionar serviços. IBM Cloud Private Cloud Foundry fornece um mecanismo para você registrar um broker de serviço e usar serviços que possam não ser fornecidos ou disponibilizados em seu ambiente de nuvem.

Esse exemplo usa o pacote do iniciador do Open Service Broker que o IBM Cloud Private Cloud Foundry fornece. Ele o orienta por meio dos processos de preparação e implementação de seu aplicativo, registro de um broker de serviço, criação de uma instância de serviço do Mongo® DB e conexão com o banco de dados Mongo usando o aplicativo de introdução do Liberty for Java™.

Antes de iniciar

Tipo de usuário ou nível de acesso necessário: para IBM Cloud Private: administrador de cluster, para IBM Cloud Private Cloud Foundry: gerenciador de espaço

Você precisa das ferramentas a seguir:

- Cloud Foundry CLI
- Git
- Maven

Etapa 1: implementar um broker de serviço em seu ambiente do IBM Cloud Private

Consulte Implementando o banco de dados Open Service Broker (OSB) no IBM Cloud Private Cloud Foundry para implementar o pacote do iniciador do Open Service Broker no ambiente do IBM Cloud Private. A partir da última etapa deste documento, anote o número da porta do nó externo.

Etapa 2: Registrar um broker de serviço

É possível usar o comando `cf create-service-broker` do Cloud Foundry para registrar um broker de serviço. Substitua ICP-IP e NODE-PORT-NUMBER.

```
$ cf create-service-broker icp-service-broker admin password https://ICP-IP:NODE-PORT-NUMBER
```

Etapa 3: tornar ofertas de serviços e planos públicos

Use o comando `cf service-access` do Cloud Foundry para ver as configurações de acesso nos serviços no mercado.

```
$cf service-access
```

Use o comando `cf enable-service-access` do Cloud Foundry para ativar o acesso no serviço do Mongo DB.

```
$cf enable-service-access mongodb-service
```

Etapa 4: criar uma instância de serviço

Crie uma instância de serviço usando o comando `cf create-service`.

```
$cf create-service mongodb-service default my-mongodb-service
```

Etapa 5: Clone o app de amostra

Clone o repo do GitHub de app de amostra.

```
git clone https://github.com/IBM-Bluemix/get-started-java
```

Etapa 6: executar o aplicativo localmente usando a linha de comandos

Use Maven para construir seu código-fonte e executar o aplicativo.

Na linha de comandos, mude o diretório para onde o app de amostra está localizado.

```
cd get-started-java
```

Use Maven para instalar as dependências e construir o arquivo .war.

```
mvn clean install
```

Execute o app localmente no Liberty.

```
mvn install liberty:run-server
```

Quando a mensagem for exibida, `O servidor defaultServer está pronto para executar um Smarter Planet`, seu app poderá ser visualizado em: `http://localhost:9080/GetStartedJava`.

Para parar seu app, pressione Ctrl-C na janela de linha de comandos na qual você iniciou o app.

Etapa 7: preparar o aplicativo para implementação

Para implementá-lo no IBM Cloud Private, pode ser útil configurar um arquivo `manifest.yml`. O arquivo `manifest.yml` inclui informações básicas sobre seu aplicativo, como o nome, quanta memória alocar para cada instância e a rota. É possível localizar um arquivo `manifest.yml` de amostra no diretório `get-started-java`.

Abra o arquivo `manifest.yml` e mude o nome de `GetStartedJava` para seu nome de app, `app_name`.

```
applications:

name: GetStartedJava
random-route: true
path: target/GetStartedJava.war
memory: 512M
instances: 1
```

Dica: nesse arquivo `manifest.yml`, `random-route: true` gera uma rota aleatória para seu app para evitar que sua rota colida com outras. Se você escolher isso, será possível substituir `random-route: true` por `host: myChosenHostName` e fornecer um nome do host de sua opção.

Etapa 8: Implementar no IBM Cloud Private Cloud Foundry

Implemente seu app em sua instância do IBM Cloud Private Cloud Foundry usando sua URL do IBM Cloud Private Cloud Foundry.

```
cf api
```

Efetue login em sua conta do IBM Cloud Private Cloud Foundry.

```
cf login
```

De dentro do diretório get-started-java, envie seu aplicativo por push para o IBM Cloud Private.

```
cf push
```

A implementação de seu aplicativo pode levar alguns minutos. Quando a implementação for concluída, você verá uma mensagem de que o seu aplicativo está em execução. Visualize seu aplicativo na URL que é listada na saída do comando push ou visualize o status de implementação do aplicativo e a URL executando o comando a seguir:

```
cf apps
```

Dica: é possível solucionar problemas de erros no processo de implementação usando o comando `cf logs --recent`.

Etapa 9: usar serviço Mongo DB MongoDB

Ligue a instância de serviço do Mongo DB ao aplicativo de introdução.

```
cf bind-service GetStartedJava my-mongodb-service
```

Remonte o aplicativo.

```
cf restage GetStartedJava
```

Etapa 10: Confirmar

Atualize a visualização do navegador em `http://localhost: 9080/GetStartedJava/`. Quaisquer nomes que você inserir no aplicativo serão incluídos no banco de dados.

Trabalhando com um broker de serviço e com o app iniciador Node.js

É possível ligar seus aplicativos no IBM® Cloud Private Cloud Foundry a serviços que existem fora de seu ambiente do IBM Cloud Private Cloud Foundry usando o aplicativo de introdução do Node.js.

Os brokers de serviço divulgam um catálogo de ofertas de serviços e planos para o mercado e agem de acordo com solicitações do mercado para fornecer, ligar, desvincular e desaprovisionar serviços. IBM Cloud Private Cloud Foundry fornece um mecanismo para você registrar um broker de serviço e usar serviços que possam não ser fornecidos ou disponibilizados em seu ambiente de nuvem.

Esse exemplo usa o pacote do iniciador do Open Service Broker que o IBM Cloud Private Cloud Foundry fornece. Ele o orienta por meio dos processos de preparação e implementação de seu aplicativo, registro de um broker de serviço, criação de uma instância de serviço do Mongo® DB e conexão com o banco de dados Mongo usando o aplicativo de introdução do Node.js.

Antes de iniciar

Tipo de usuário ou nível de acesso necessário: para IBM Cloud Private: administrador de cluster, para IBM Cloud Private Cloud Foundry: gerenciador de espaço

Você precisa das ferramentas a seguir:

- Cloud Foundry CLI
- Git
- Maven

Etapa 1: implementar um broker de serviço em seu ambiente do IBM Cloud Private

Consulte Implementando o banco de dados Open Service Broker (OSB) no IBM Cloud Private Cloud Foundry para implementar o pacote do iniciador do Open Service Broker no ambiente do IBM Cloud Private.

Etapa 2: Registrar um broker de serviço

É possível usar o comando `cf create-service-broker` do Cloud Foundry para registrar um broker de serviço. Substitua ICP-IP e NODE-PORT-NUMBER.

```
$ cf create-service-broker icp-service-broker admin password https://ICP-IP:NODE-PORT-NUMBER
```

Etapa 3: tornar ofertas de serviços e planos públicos

Use o comando `cf service-access` do Cloud Foundry para ver as configurações de acesso nos serviços no mercado.

```
$cf service-access
```

Use o comando `cf enable-service-access` do Cloud Foundry para ativar o acesso no serviço do Mongo DB.

```
$cf enable-service-access mongodb-service
```

Etapa 4: criar uma instância de serviço

Crie uma instância de serviço usando o comando `cf create-service`.

```
$cf create-service mongodb-service default my-mongodb-service
```

Etapa 5: Clone o app de amostra

Clone o repo do GitHub de app de amostra.

```
git clone https://github.com/IBM-Bluemix/get-started-node
```

Etapa 6: executar o aplicativo localmente usando a linha de comandos

Na linha de comandos, mude o diretório para onde o app de amostra está localizado.

```
cd get-started-node
```

Instale as dependências que são listadas no arquivo `package.json` para executar o aplicativo localmente.

```
npm install
```

Execute o aplicativo.

```
npm start
```

É possível visualizar seu app em `http://localhost:3000`.

Dica: use o nodemon para reinicialização automática do aplicativo com mudanças de arquivo.

Etapa 7: preparar o aplicativo para implementação

Para implementar seu aplicativo no IBM Cloud Private, pode ser útil configurar um arquivo `manifest.yml`. O arquivo `manifest.yml` inclui informações básicas sobre seu aplicativo, como o nome, quanta memória alocar para cada instância e a rota. É possível localizar um arquivo `manifest.yml` de amostra no diretório `get-started-node`.

Abra o arquivo `manifest.yml` e mude o nome de `GetStartedNode` para o nome de seu app, `app_name`.

```
applications:

name: GetStartedNode
random-route: true
memory: 128M
Copy
```

Dica: nesse arquivo `manifest.yml`, `random-route: true` gera uma rota aleatória para seu app para evitar que sua rota colida com outras. Se você escolher isso, será possível substituir `random-route: true` por `host: myChosenHostName` e fornecer um nome do host de sua opção.

Etapa 8: Implementar no IBM Cloud Private Cloud Foundry

Implemente o seu aplicativo IBM Cloud Private Cloud Foundry usando a URL do IBM Cloud Private Cloud Foundry.

```
cf api
```

Efetue login em sua conta do IBM Cloud Private Cloud Foundry.

```
cf login
```

De dentro do diretório get-started-node, envie seu aplicativo por push para o IBM Cloud Private.

```
cf push
```

A implementação de seu aplicativo pode levar alguns minutos. Quando a implementação for concluída, você verá uma mensagem de que o seu aplicativo está em execução. Visualize seu aplicativo na URL que é listada na saída do comando push ou visualize o status de implementação do aplicativo e a URL executando o comando a seguir:

```
cf apps
```

Dica: é possível solucionar problemas de erros no processo de implementação usando o comando `cf logs --recent`.

Etapa 9: usar serviço Mongo DB MongoDB

Ligue a instância de serviço do Mongo DB ao aplicativo de introdução.

```
cf bind-service GetStartedNode my-mongodb-service
```

Remonte o aplicativo.

```
cf restage GetStartedNode
```

Etapa 10: Confirmar

Atualize a visualização do navegador em `http://localhost`. Quaisquer nomes que você inserir no aplicativo serão incluídos no banco de dados. Anote o número da porta do nó externo.

Gerenciando aplicativos Liberty e Node.js no IBM Cloud Private Cloud Foundry

O App Management é um conjunto de utilitários de desenvolvimento e depuração que estão disponíveis para seus aplicativos Liberty e Node.js no IBM® Cloud Private Cloud Foundry.

Descontinuação: todos os utilitários do App Management foram descontinuados para aplicativos Node.js e são elegíveis para remoção em uma liberação futura. Aprenda mais em O que há de novo no IBM Cloud Private Cloud Foundry versão 3.2.0.

Utilitários do App Management

Os buildpacks fornecem os utilitários de gerenciamento de aplicativo a seguir.

Utilitários para o Liberty e o Node.js (descontinuados para o Node.js)

- proxy
- noproxy
- hc

Utilitários do Liberty

- depuração
- jmx
- localjmx

Utilitários Node.js (descontinuado)

- inspector

Restrições

- As mudanças que você faz em seu aplicativo usando App Management são temporárias e são perdidas após você sair desse modo. Esse modo é apenas para uso de desenvolvimento provisório e não se destina a ser usado como um ambiente de produção devido ao desempenho.
- A maioria dos utilitários App Management não funcionará se você configurar seu comando `start` no arquivo `manifest.yml` (comando) ou `CF CLI (-c)`. Esses métodos são substituições de buildpack e não são as melhores práticas para iniciar aplicativos Node.js. Para obter melhores resultados, configure o comando `start` no arquivo `package.json` ou `Procfile`.

Como configurar o App Management

Para ativar os utilitários do App Management, configure a variável de ambiente `BLUEMIX_APP_MGMT_ENABLE` e remonte seu aplicativo. Vários utilitários podem ser ativados, separando-os com um **+**.

Por exemplo, para ativar os utilitários hc e debug, execute o comando a seguir:

```
cf set-env myApp BLUEMIX_APP_MGMT_ENABLE hc+debug
```

Remonte seu aplicativo após configurar a variável de ambiente:

```
cf restage myApp
```

Se você não deseja que os utilitários do App Management sejam instalados com seu aplicativo, configure a variável de ambiente `BLUEMIX_APP_MGMT_INSTALL` como `false` e prepare novamente seu aplicativo.

Por exemplo, execute os comandos a seguir para montar seu aplicativo sem utilitários App Management:

```
cf set-env myApp BLUEMIX_APP_MGMT_INSTALL false
cf restage myApp
```

Utilitários Liberty e Node.js (descontinuados para o Node.js)

proxy

O utilitário proxy fornece gerenciamento de aplicativo mínimo entre seu aplicativo e seu ambiente de nuvem.

Quando ativado, o buildpack inicia um agente de proxy que está localizado entre o tempo de execução e o contêiner do aplicativo. O utilitário de proxy manipula todas as solicitações que o aplicativo recebe. Com base no tipo de solicitação, ele executa uma ação do App Management ou encaminha a solicitação para seu aplicativo. Ao usar o proxy, é possível ativar a maioria dos outros utilitários do App Management. Ao ativar o proxy, seu contêiner de aplicativo continua ativo mesmo quando o aplicativo trava. Ao usar o agente de proxy, é possível configurar atualizações de arquivo incrementais, que ativam o modo de Edição em Tempo Real para aplicativos Node.js.

noproxy

Alguns utilitários, como o utilitário inspetor para Node.js, iniciam automaticamente o utilitário de proxy. Quando o proxy é iniciado automaticamente, é possível usar o utilitário noproxy para desativar o proxy. Com as células do Diego, o proxy não é necessário, já que o Diego fornece a capacidade de SSH diretamente para seu aplicativo e configura o encaminhamento de porta.

O utilitário noproxy se aplica apenas a aplicativos que são executados em uma célula do Diego.

hc

O agente do Health Center (hc) permite que seu aplicativo seja monitorado pelo cliente do Health Center. O agente hc está disponível apenas com versões de tempo de execução do IBM SDK for Node.js. Consulte Atualizações mais recentes para o buildpack sdk-for-nodejs para o conjunto atual de tempos de execução.

O Health Center analisa o desempenho de seus aplicativos Liberty e Node.js usando o IBM Monitoring and Diagnostic Tools. Para obter mais informações, consulte Como analisar o desempenho dos aplicativos Liberty Java™ ou Node.js no IBM Cloud.

Importante: o utilitário hc inicia o proxy.

É possível usar o utilitário hc com noproxy. Para usar o Health Center com noproxy, primeiro estabeleça o encaminhamento de porta usando o comando `cf ssh`. Por exemplo:

```
cf ssh -N -T -L 1883:127.0.0.1:1883 <appName>
```

Em seguida, para conectar-se com o cliente do Health Center, use uma Conexão MQTT. e especifique o host como 127.0.0.1 e a porta como 1883.

Modo de desenvolvimento para ferramentas Eclipse

O modo de desenvolvimento é um recurso do Eclipse Tools for IBM Cloud que fornece aos desenvolvedores a capacidade de trabalhar com seus aplicativos enquanto eles estão em execução na nuvem. O Modo de Desenvolvimento no Eclipse Tools fornece uma maneira para você trabalhar em seus aplicativos no IBM Cloud Private com uma área de trabalho temporária e segura.

O modo de desenvolvimento é suportado para aplicativos Liberty e Node.js. Se você ativar o modo de desenvolvimento para o aplicativo Liberty ou Node.js, será possível atualizar arquivos de aplicativos incrementalmente sem precisar enviar seu aplicativo por push. Também é possível estabelecer uma sessão de depuração com seu aplicativo. O modo de desenvolvimento para aplicativos Liberty é equivalente e ativar os utilitários debug e jmx do App Management. Para aplicativos Node.js, isso é equivalente a ativar o utilitário inspetor.

Utilitários do Liberty

debug

Para usar o utilitário de depuração, é necessário instalar o Eclipse Tools for IBM Cloud.

O utilitário de depuração coloca o aplicativo Liberty no modo de depuração e permite que os clientes, como o IBM Eclipse Tools for IBM Cloud, estabeleçam uma sessão de Depuração remota do com o aplicativo.

Importante: o utilitário de depuração inicia o proxy.

O utilitário de depuração pode ser usado com noproxy. Para usar o utilitário de depuração com noproxy, primeiro estabeleça o encaminhamento de porta usando o comando `cf ssh`. O fragmento de código a seguir mostra um exemplo do formato de comando `cf ssh`:

```
cf ssh -N -T -L 7777:127.0.0.1:7777 <appName>
```

Em seguida, para se conectar no Eclipse, use **Configuração Java Remota** e especifique o host como `127.0.0.1` e a porta como `7777`.

jmx

O utilitário jmx ativa o JMX REST Connector para permitir que um cliente JMX remoto gerencie o aplicativo usando credenciais do usuário do IBM Cloud.

Para obter mais informações sobre como configurar um conector JMX, consulte Configurando conexão JMX segura com o perfil Liberty.

Importante: o utilitário jmx não inicia o proxy.

localjmx

O utilitário localjmx ativa o recurso Liberty localConnector-1.0. Combinar esse utilitário com o encaminhamento de porta local cria uma maneira alternativa de permitir que um cliente JMX remoto gerencie o aplicativo.

Antes de iniciar: o utilitário localjmx requer que você instale o JConsole.

O utilitário localjmx se aplica apenas a aplicativos que são executados em uma célula do Diego. Para usar o localjmx, primeiro estabeleça o encaminhamento de porta usando o comando `cf ssh`. Por exemplo:

```
cf ssh -N -T -L 5000:127.0.0.1:5000 <appName>
```

Em seguida, para conectar com o JConsole, escolha **Processo remoto**, especifique `127.0.0.1:5000` e use uma conexão insegura.

Utilitários Node.js (descontinuado)

inspector

O utilitário inspector pode ser usado para criar perfis de uso da CPU, incluir pontos de interrupção e depurar código, tudo enquanto seu aplicativo é executado no IBM Cloud Private. Para versões do Node.js anteriores à 6.3.0, o inspector ativa a interface do depurador do inspetor do Node. Para obter mais informações sobre o inspetor do Node, consulte o arquivo leia-me para node-inspector no GitHub. Para o Node.js versões 6.3.0 e superiores, o utilitário inspetor usa a Integração do Inspetor V8 para Node.js.

Para versões do Node.js posteriores à 6.3.0

Ao iniciar o modo de depuração, o proxy é ativado automaticamente, mesmo que você use uma versão do Node.js que não inclua proxy. As versões do Node.js posteriores à 6.3.0 não incluem proxy. Se você usar o utilitário inspetor com versões do Node.js posteriores à 6.3.0, é possível desativar o proxy usando noproxy.

Em vez de usar proxy para acessar a interface do inspetor, é possível usar o recurso Ferramentas do Desenvolvedor do navegador da web Google Chrome.

Ative o acesso à URL com encaminhamento de porta local com o comando a seguir:

```
cf ssh -N -T -L 9229:127.0.0.1:9229 <appName>
```

Obtenha o log de inicialização para o aplicativo usando o comando a seguir:

```
cf logs <appName> --recent
```

Se o utilitário inspector estiver ativo, o log mostrará uma saída semelhante à mensagem a seguir:

```
2017-10-16T14:37:44.75-0400 [APP/PROC/WEB/0] ERR
```

Use uma versão atualizada do navegador da web Chrome para navegar para `chrome://inspect`. Nessa URL, é possível ver seu app que está listado com um link para seus arquivos de aplicativo, como `file://home/vcap/app/app.js`. Selecione **inspecionar** para acessar a interface do inspetor.

Para versões do Node.js anteriores à 6.3.0

Se você usar o proxy, é possível acessar a interface do inspetor em `https://myApp.mybluemix.net/bluemix-debug/inspector`.

Se você não usar o utilitário de proxy, ative o acesso à URL do aplicativo usando o encaminhamento de porta local com o comando a seguir:

```
cf ssh -N -T -L 8790:127.0.0.1:8790 <appName>
```

Em seguida, acesse o inspetor por meio da URL, `http://127.0.0.1:8790`.

Criando uma Instância de Serviço

Crie uma instância de serviço e ligue-a a seu aplicativo.

Para obter mais informações sobre a configuração de serviços, consulte Usando serviços do IBM Cloud no IBM Cloud Private Cloud Foundry.

1. Confirme se você tem acesso para criar uma instância de serviço. Execute o comando a seguir para visualizar uma lista de serviços ativados. Se você não tiver acesso, entre em contato com o administrador do cluster.

   ```
   cf marketplace
   ```

2. Forneça um serviço criando uma instância de serviço:

   ```
   cf create-service <service-name> <service-plan> <your-service-instance-name>
   ```

3. Ligue a instância de serviço a seu aplicativo usando este comando:

   ```
   cf bind-service <application-name> <service-instance-name>
   ```

Resolução de problemas

Saiba como isolar e resolver problemas com os aplicativos IBM® Cloud Private Cloud Foundry, IBM Cloud Private Cloud Foundry e o Cloud Foundry Enterprise Environment.

Resolução de Problemas do IBM® Cloud Private Cloud Foundry

Saiba como isolar e resolver problemas com o IBM Cloud Private Cloud Foundry.

Resolução de Problemas de Instalação e Upgrade

Resolva problemas que possam ocorrer quando estiver instalando ou fazendo upgrade do IBM® Cloud Private Cloud Foundry.

O contêiner de concepção não inicia

O contêiner de concepção não inicia.

Sintomas

Durante a instalação, a mensagem a seguir é exibida:

```
Error response from daemon: driver failed programming external connectivity on the endpoint...
```

Resolvendo o problema

Ative a porta que você especificou no comando `launch.sh` durante a instalação. Execute este comando:

```
ufw enable <port_number>
```

Em que `<port_number>` é o valor especificado. O número da porta padrão é 8483.

Também é possível usar `iptables` para abrir a porta que você especificou no host para comunicação de entrada.

O grupo da porta não possui permissão de Administrador

Os privilégios do VMware para o usuário que está definido no arquivo `/<installation_directory>/uiconfig_<iaas_type>_template.yml` não são suficientes no grupo da porta. A execução do script `deploy-director.sh` falha.

Sintomas

A saída do comando a seguir não contém informações do grupo da porta.

```
docker exec -it inception-<name> "cat /tmp/bosh-init-run.log"
```

Em que `<name>` é o valor fornecido na execução do script `launch.sh -n <name>` durante a instalação.

Resolvendo o problema

Se você usar um comutador vSphere Distributed, inclua o usuário definido como Administrador no arquivo `/<installation_directory>/uiconfig_<iaas_type>_template.yml` para a pasta que contém o grupo da porta. Se você usar um Comutador Virtual Padrão, inclua o usuário no grupo da porta.

Cloud Foundry launch_deployment.sh falha devido a um caractere inválido

Cloud Foundry launch_deployment.sh falha devido a um caractere inválido.

Sintomas

Ao executar `launch_deployment.sh`, um erro semelhante à mensagem a seguir é exibido:

```
invalid character 'N' looking for beginning of value
strconv.Atoi: parsing "invalid character 'N' looking for beginning of value": invalid syntax
```

Resolvendo o problema

1. Revise o arquivo `uiconfig_<iaas_type>_template.yml` para ver se ele contém erros de sintaxe do YAML ou chaves inválidas. O arquivo deve atender às características a seguir:
   - O arquivo usa indentação YAML adequada.
   - Cada instância de um caractere especial está delimitada por aspas duplas. As senhas geralmente incluem caracteres especiais.
   - Números que incluem zeros iniciais ou finais são cercados por aspas duplas.
   - Todas as chaves são válidas.
2. Execute o script `launch_deployment.sh` novamente.

A implementação falha porque os FQDNs do vCenter ou ESXs não podem ser resolvidos

A implementação do IBM® Cloud Private Cloud Foundry falha porque os nomes completos de domínio do vCenter ou ESXs não podem ser resolvidos.

Sintomas

Durante a fase de validação da instalação, um erro semelhante à mensagem a seguir é exibido:

```
=>
Name         : vmware_address
Value        : vcenter.mycompany.com
Message type: error
Message      : Can not reach VMware, please check connectivity. dial tcp: lookup
vcenter.mycompany.com on 8.8.8.8:53: no such host
```

Causas

O contêiner de concepção não pode aplicar o valor do servidor DNS correto porque o valor errado está configurado no arquivo `/etc/resolv.conf` do contêiner.

A maneira como o Docker fornece informações de DNS para contêineres é uma possível causa. O Docker aplica os valores no arquivo `/etc/resolv.conf` no computador host para o contêiner de concepção. Se esse arquivo não contiver o valor de parâmetro `nameserver`, ele não será aplicado ao contêiner de concepção. Se o endereço IP do `nameserver` também estiver no arquivo `/etc/hosts` no computador host, o valor não será fornecido para o contêiner. O Docker filtra o endereço. Se o contêiner

não tiver um bom valor de `nameserver` para aplicar ou se o Docker filtrou o endereço, ele usará o endereço de DNS do Google. Consulte Configurar DNS do contêiner na documentação do Docker.

Outra causa possível é que você configurou o valor de parâmetro `dnsmasq` para `emulate` um curinga que suporta DNS. Nesse caso, o valor no arquivo `/etc/resolv.conf` do host será sobrescrito pelo endereço IP 127.0.0.1 que está no arquivo `/etc/hosts`. Devido à maneira como o Docker filtra o valor, o valor de `nameserver` no arquivo `/etc/resolv.conf` no contêiner será configurado com o endereço de DNS do Google.

Resolvendo o problema

1. Verifique se é possível resolver o nome completo do domínio (FQDN) para o vCenter do host.
2. Verifique se é possível resolver o FQDN para o vCenter no contêiner.
3. Abra o arquivo `/etc/resolv.conf` em seu contêiner de concepção e confirme se ele não contém o endereço IP do DNS.
4. Se o arquivo `/etc/resolv.conf` não contiver o endereço IP do DNS, verifique se o arquivo `/etc/hosts` no host o contém. Se o arquivo `/etc/hosts` contiver o endereço IP do DNS, remova-o.
5. No contêiner de concepção, abra o `/etc/docker/daemon.json` e confirme se você não substituiu a especificação do DNS. Se você o fez, remova essa especificação para que o arquivo `/etc/resolv.conf` no contêiner reflita a versão desse arquivo no host.
6. Execute o comando `launch.sh` novamente.
7. Execute o comando `launch_deployment.sh` novamente.

A tarefa consul falha durante uma implementação BOSH

A tarefa consul falha durante uma implementação BOSH.

Sintomas

Depois de executar uma implementação do BOSH, efetue login na máquina virtual com o comando a seguir, em que ou identifica a máquina virtual específica para a conexão SSH:

```
bosh -e IBMCloudPrivate -d Bluemix ssh <job name>/<uuid>

ou

bosh -e IBMCloudPrivate -d Bluemix ssh <job name>/<index>
```

Visualize o status do processo:

```
sudo su -
monit summary
```

E, em seguida, revise os logs na pasta `/var/vcap/sys/log/consul_agent`. Os logs `stderr` ou `stdout` mostram que um ou mais agentes consul não podem ser atingidos.

Resolvendo o problema

1. Modifique o arquivo `custom-cf.yml`:

   ```
   fqdn_groups:
    consul_agent_ips:
    - (( delete "<instance_name>" ))
   ```

   Em que `<instance_name>` é o nome da tarefa do BOSH com falha.

2. Inicie a implementação novamente.
3. Se a implementação for bem-sucedida, remova as linhas de código incluídas no arquivo `custom-cf.yml`.
4. Inicie a implementação novamente e confirme se ela foi bem-sucedida.

A implementação do Cloud Foundry atinge o tempo limite

A implementação do Cloud Foundry não é concluída porque atinge o tempo limite.

Sintomas

Cada etapa de uma implementação possui um tempo limite padrão. Quando a implementação da etapa leva mais tempo do que o valor de tempo limite padrão, você vê um erro que é semelhante à mensagem a seguir:

```
State name: director
Label     : Director
Phase     :
Script    : /repo_local/cfp-bosh-templates/2.3.2c-ff920bc/scripts/deploy-director.sh
Timeout   : 60
LogPath   : /data/logs/CloudFoundry/deploy-director.log
Status    : FAILED
Start time: Wed Jan 10 18:06:52 UTC 2018
End time  : Wed Jan 10 19:06:52 UTC 2018
Reason    : Cmd failed:State director killed as timeout reached
```

Causas

Você geralmente encontra esse erro quando possui uma conexão de rede lenta entre o contêiner de concepção e seu ambiente do vSphere.

Resolvendo o problema

É possível aumentar esse valor de tempo limite configurando um valor de tempo limite maior para essa etapa.

1. Em seu diretório de instalação, execute o comando a seguir:

   ```
   ./cm state -s <state_name> set --timeout <new_timeout>
   ```

   Em que `<state_name>` é o nome do estado que falhou e `<new_timeout>` é o tempo, em minutos, em que a etapa deve ser executada.

2. Execute o comando `launch_deployment.sh` novamente.

Falhas de Validação do Cloud Foundry

A validação de Cloud Foundry falha.

Sintomas

A validação mostra erros, não avisos, mas após a verificação, a equipe de suporte e você deduziram que o ambiente está bem configurado.

Resolvendo o problema

1. Verifique o erro e confirme se a infraestrutura foi configurada corretamente.
2. Verifique se o arquivo uiconfig contém as informações corretas.
3. Se sua configuração estiver correta, execute a implementação sem validação. Execute o comando a seguir:

   ```
   launch_deployment.sh --no-validation
   ```

A implementação do Cloud Foundry falha em uma tarefa de máquina virtual específica

A implementação do Cloud Foundry falha em uma tarefa de máquina virtual específica.

Sintomas

Ao implementar um aplicativo Cloud Foundry, um erro semelhante à mensagem a seguir é exibido:

```
Started updating job debian_nfs_server > debian_nfs_server/0 (c9a7c3c2-1d3b-41d2-a14c-f3c4f1ec742d)
(canary)..... Done (00:01:13)
 Started updating job consul > consul/0 (7ade5ffb-a450-46e3-b94a-c1f8be310ef7) (canary)..... Done
(00:01:08)
 Started updating job nats > nats/0 (76fe6c8e-1d21-4422-a407-8f79a78e03b7) (canary)..... Done
```

```
(00:01:15)
 Started updating job ccdb_ng > ccdb_ng/0 (d4b559cb-b1e8-4b20-87de-37182e2b30ec) (canary)..... Done
(00:01:10)
 Started updating job uaadb > uaadb/0 (e68e8783-049b-4f10-95c4-e922d2eccff6) (canary)..... Done
(00:01:11)
 Started updating job router > router/0 (20d6e777-2ffe-42f7-8fbd-8b1675c5fa3c) (canary).... Done
(00:00:47)
 Started updating job dea_next > dea_next/0 (75dae58e-de27-4f61-b396-1618fcc1ac04) (canary).....
Done (00:01:05)  Started updating job cc_core > cc_core/0 (87de372a-ac40-471c-b621-b4a6ac3ca602)
(canary)..................................... Failed: 'cc_core/0 (87de372a-ac40-471c-b621-
b4a6ac3ca602)' is not running after update. Review logs for failed jobs: cloud_controller_ng,
cloud_controller_worker_local_1, cloud_controller_worker_local_2, nginx_cc,
cloud_controller_migration, cloud_controller_worker_1, cloud_controller_clock (00:08:48)
Error 400007: 'cc_core/0 (87de372a-ac40-471c-b621-b4a6ac3ca602)' is not running after update. Review
logs for failed jobs: cloud_controller_ng, cloud_controller_worker_local_1,
cloud_controller_worker_local_2, nginx_cc, cloud_controller_migration, cloud_controller_worker_1,
cloud_controller_clock
Task 9 error
```

Nessa mensagem, a tarefa com falha é denominada **cloud_controller_ng**.

Resolvendo o problema

1. Efetue login no BOSH Director com as credenciais administrativas.
2. Assegure-se de que sua implementação BOSH esteja configurada. Se ela não estiver configurada, instale a CLI do BOSH.
3. Determine o ID de seu `cc_core`. Execute o comando a seguir:

   ```
   bosh -e IBMCloudPrivate -d Bluemix vms
   ```

4. Para determinar qual serviço falhou, execute os comandos a seguir:

   ```
   bosh -e IBMCloudPrivate -d Bluemix ssh cc_core/<uuid>
   sudo su
   monit summary
   ```

   A saída é semelhante ao texto a seguir

   ```
   The Monit daemon 5.2.5 uptime: 27m

   Process 'unbound'                   running
   Process 'consul_agent'              running
   Process 'cloud_controller_ng'       Execution failed
   Process 'cloud_controller_worker_local_1' initializing
   Process 'cloud_controller_worker_local_2' initializing
   Process 'nginx_cc'                  initializing
   Process 'cloud_controller_migration' Does not exist
   Process 'cloud_controller_worker_1' Does not exist
   File 'nfs_mounter'                  accessible
   Process 'cloud_controller_clock'    Does not exist
   Process 'statsd-injector'           running
   Process 'route_registrar'           running
   Process 'loginserver'               running
   Process 'uaa'                       running
   Process 'mod_vms'                   running
   Process 'metron_agent'              running
   System 'system_localhost'           running
   ```

   Nesse exemplo, o processo do **cloud_controller_ng** falhou.

5. Localize o diretório para o processo com falha:

   ```
   cd /var/vcap/sys/log/<failed_process> ; ls
   ```

   Em que `<failed_process>` é o nome do processo que falhou. A saída se assemelha ao código a seguir:

   ```
   -rw-r--r--  1 root root 20250 Aug 22 17:30 cloud_controller_ng_ctl.err.log
   -rw-r--r--  1 root root 24159 Aug 22 17:30 cloud_controller_ng_ctl.log
   -rw-r--r--  1 root root 33988 Aug 22 17:30 cloud_controller_worker_ctl.err.log
   -rw-r--r--  1 root root 21678 Aug 22 17:30 cloud_controller_worker_ctl.log
   -rw-r-----  1 root root   506 Aug 22 17:00 pre-start.stderr.log
   -rw-r-----  1 root root     0 Aug 22 17:00 pre-start.stdout.log
   ```

6. Revise cada log nesse diretório até localizar o erro.
7. Corrija o erro.

A implementação do Cloud Foundry falha devido a conflitos de endereço IP

A implementação do Cloud Foundry falha devido a conflitos de endereço IP.

Sintomas

Durante o estágio de criação da máquina virtual (VM) da implementação do Cloud Foundry, um erro semelhante à mensagem a seguir é exibido:

```
11:52:30 | Compiling packages: tps/ef9adb86728a43959bd8c3549106f7582513a294 (00:00:06)
             L Error: Unknown CPI error 'Unknown' with message 'Detected IP conflicts with other VMs
on the same networks: tools-vm01 on network VM Network with ip 172.16.215.160' in 'create_vm' CPI
method..................
```

Resolvendo o problema

1. Abra sua instância do vCenter e veja se uma VM já usa o endereço IP que foi exibido na mensagem de erro. Uma VM que não faz parte de sua implementação do IBM Cloud Private Cloud Foundry pode usar o endereço IP
2. Se o endereço IP estiver em uso, determine se ele é usado por uma VM que faz parte de sua implementação do IBM Cloud Private Cloud Foundry. Se o nome da VM iniciar com o prefixo `vm-`, ela faz parte da implementação do IBM Cloud Private Cloud Foundry.
3. Corrija o problema.
   - Se a VM não faz parte da implementação do IBM Cloud Private Cloud Foundry, a `subnet` ou o `address_range` que você especificou durante a instalação do IBM Cloud Private Cloud Foundry contém VMs. Determine os valores de `subnet` e `address_range` que você reserva para IBM Cloud Private Cloud Foundry e atualize sua instalação do IBM Cloud Private Cloud Foundry. Consulte Instalando IBM® Cloud Private Cloud Foundry.
   - Se a VM faz parte da implementação do IBM Cloud Private Cloud Foundry e o erro não ocorreu durante a fase de **compilação**, pode ser necessário limpar seu conjunto de recursos do VMware. Se você reiniciou a implementação excluindo o diretório de dados, as VMs que são criadas durante a instalação não podem ser removidas. Limpe o conjunto de recursos e execute a implementação novamente.
   - Se a VM faz parte da implementação do IBM Cloud Private Cloud Foundry e o erro ocorreu durante a fase de **compilação**, entre em contato com o suporte IBM.

O IBM Cloud Private Cloud Foundry registro automático do console falha

O console da web do IBM Cloud Private Cloud Foundry não registra automaticamente.

Sintomas

Ao tentar acessar o console da web do IBM Cloud Private Cloud Foundry, você é solicitado a inserir detalhes da UAA.

Resolvendo o problema

Antes de iniciar

1. Deve-se fornecer um endereço IP na propriedade `console_ip` do arquivo `uiconfig` para o console do IBM Cloud Private Cloud Foundry. Se você não fornecer um endereço IP, sua validação falhará e a instalação não será iniciada. Para obter informações sobre a propriedade `console_ip`, consulte Parâmetros comuns.

2. Após a conclusão da implementação, recupere os valores necessários de seus arquivos de configuração. **Dica**: esses valores são necessários ao configurar as **Credenciais do console para o UAA.**

   1. Navegue para o diretório de configuração usado durante sua implementação do IBM Cloud Private Cloud Foundry.

   2. Procure o `./uiconfig.yml` para recuperar valores para as propriedades a seguir:

```
bluemix_env_domain
main_user_name
main_user_password
```

3. Em `./CloudFoundry/deployment-vars.yml`, recupere o valor para `ibm_uaa_cf_ui_secret`.

```
ibm_uaa_cf_ui_secret:
  password: ""
  complexity: 12
  description: "Secret for cf_ui client-id, that is used by the Console to access Cloud
Foundry"
```

Conectando-se ao seu IBM Cloud Private Cloud Foundry console

1. Navegue para `https://<console_ip>:4443`.
2. Aceite o certificado autoassinado.
3. Na janela **Introdução** , clique em **Avançar** .
4. Especifique valores para as informações de ambiente a seguir na janela **Terminal do UAA**
   - **UAA Terminal**
     - **URL da API:** especifique a URL que aponta para o UAA `https://uaa.BLUEMIX_ENV_DOMAIN`, por exemplo: `https://uaa.cf.ibm.com`.
     - **Ignorar a validação de SSL para o terminal**: selecione somente quando você estiver usando certificados autoassinados com o IBM Cloud Private Cloud Foundry.
   - **Console credenciais para UAA**
     - **ID do cliente:** o `cf_ui` é criado automaticamente durante a implementação do IBM Cloud Private Cloud Foundry.
     - **Segredo do cliente**: especifique o valor da senha `ibm_uaa_cf_ui_secret`.
     - **Nome do usuário administrativo**: especifique o valor `main_user_name`.
     - **Senha do Administrador**: Especifique o `main_user_password` valor.
   - Depois de ter especificado todos os campos obrigatórios, clique em **Avançar**.
   - Se você receber um erro de SSL, marque **Ignorar verificação de SSL** nesta etapa.
5. Na janela **Escopo do administrador do console**, selecione `stratos.admin` no menu **Escopo** e clique em **Concluído**.

Quando bem-sucedido, a janela de login do console do IBM Cloud Private Cloud Foundry é aberta.

Nota: é necessário registrar o terminal do IBM Cloud Private Cloud Foundry seguindo as instruções em Registrando o IBM Cloud e plataformas adicionais do Cloud Foundry com o console.

Resolução de Problemas de Login

Resolva problemas que possam estar ocorrendo ao efetuar login.

- O usuário administrativo do Cloud Foundry está bloqueado

O usuário administrativo do Cloud Foundry está bloqueado

O usuário administrativo do Cloud Foundry está bloqueado.

Sintomas

O usuário administrativo do Cloud Foundry está bloqueado no IBM® Cloud Private Cloud Foundry.

Resolvendo o problema

1. Obtenha o valor `uaa_admin_client_secret`.
   - Para o IBM Cloud Private Cloud Foundry, use o valor do arquivo `/data/CloudFoundry/deployment-vars.yml`.
   - Para Cloud Foundry Enterprise Environment, use o seguinte comando:

     ```
     kubectl get -n uaa secret secrets -o jsonpath='{.data.uaa-admin-client-secret}' | base64
     --decode
     ```

2. Usar UAAC para conectar-se ao UAA do Cloud Foundry usando o valor **uaa_admin_client_secret**. Execute os comandos a seguir:

```
uaac target uaa.<bluemix_env_domain>
uaac token client get admin -s <uaa_admin_client_secret>
uaac password set <user_name> -p <new_password>
```

3.
   - Para o IBM Cloud Private Cloud Foundry, no arquivo `/data/CloudFoundry/bmxconfig.yml`, atualize as credenciais do usuário administrativo.
   - Para o Cloud Foundry Enterprise Environment, se necessário, use o Ferramenta de implementação do Cloud Foundry para atualizar a senha do administrador do UAA.

Resolução de Problemas de Configuração

Resolva problemas que possam estar ocorrendo quando estiver configurando seu ambiente.

- A máquina virtual BOSH é mostrada como não responsiva
- O BOSH ssh falha no Openstack

A máquina virtual BOSH mostra não responsivo

Sintomas (Detecção)

1. Um alerta do Prometheus mostra uma máquina virtual com uma mensagem `failure`.
2. O `bosh -e IBMCloudPrivate vms` mostra uma máquina virtual (tarefa) com uma mensagem `fail`.

Determine se o uso do disco está em 100%

1. Efetue login no cliente bosh.
2. Verifique o uso do disco executando o comando a seguir:

   ```
   bosh -e IBMCloudPrivate vms -- vitals
   ```

3. Conecte-se à máquina virtual em questão usando bosh, em que `0` é a instância da máquina virtual em questão:

   ```
   bosh -e IBMCloudPrivate -d Bluemix ssh JOB_NAME/0
   ```

Comandos Úteis

Os comandos a seguir podem ser executados quando você está conectado à máquina virtual:

```
df -k             # List all disk usage for the virtual machine
du --max-depth=1  # List the sizes for all files and directories in the current location.
```

Corrigindo o uso do disco persistente em 100% para o banco de dados [/var/vcap/store]

1. Efetue login como um usuário raiz.

   ```
   Sudo su-
   ```

2. Pare o banco de dados. **AVISO:** isso causará alguns problemas de interrupção do CloudFoundry, no entanto, se o banco de dados já estiver lidando com um disco cheio, já há problemas de indisponibilidade.

   ```
   monit stop postgres
   ```

3. Execute o comando a seguir para se tornar o usuário `vcap`:

   ```
   sudo su vcap
   ```

4. Limpe os logs de transações:

   ```
   /var/vcap/packages/postgres-9.6.6/bin/pg_resetxlog -f /var/vcap/store/postgres/postgres-9.6.6/
   ```

Nota: este comando pode demorar um pouco, mas reduzirá o tamanho de `/var/vcap/store`.

5. Efetue logoff como vcap.

   ```
   sair
   ```

6. Inicie o banco de dados.

   ```
   monit start postgres
   ```

7. Valide se o uso do disco não é mais 100%.

Corrigindo o uso do disco Efêmero em 100% [`/var/vcap/data`]

1. Efetue login como um usuário raiz.

   ```
   Sudo su-
   ```

2. Na máquina virtual, emita o comando a seguir para mudar o diretório: `cd /var/vcap/data`.

3. Siga os maiores tamanhos de arquivos para determinar se quaisquer arquivos ou diretórios grandes podem ser removidos.

   ```
   du --max-depth=1
   ```

4. A maioria dos arquivos em `/var/vcap/data/sys/log` pode ser removida. Se os logs forem necessários, copie-os para um local externo, em seguida, remova as cópias locais.

5. Valide se o uso do disco não é mais 100%.

Correção alternativa para uso de disco efêmero em 100% [`/var/vcap/data`]

NOTA: esta solução corrige apenas o disco efêmero e não deverá ser executada se o uso do disco persistente for 100%

1. Como o disco efêmero não contém dados persistentes, a máquina virtual pode ser reconstruída.

2. Recrie a máquina virtual usando bosh.

   ```
   bosh -e IBMCloudPrivate -d Bluemix recreate JOB_NAME/INDEX  # Example: JOB_NAME/INDEX =
   ccdb_ng/0
   Continue? [yN]: y
   ```

3. Depois que a máquina virtual for recriada, verifique se o uso do disco não está mais em 100%.

Ssh BOSH falha em Openstack

O BOSH ssh falha no Openstack.

Sintomas

Ao executar no Openstack e tentar o comando `bosh ssh` para uma máquina virtual bosh, ele falha com a notificação a seguir:

```
verificação de chave do host falhou.
```

Resolvendo o problema

1. Efetue login no BOSH Director com as credenciais administrativas.
2. Assegure-se de que sua implementação BOSH esteja configurada. Se ela não estiver configurada, instale a CLI do BOSH.
3. Descubra o endereço IP do diretor BOSH executando o comando a seguir:

   ```
   envs bosh -e IBMCloudPrivate
   ```

4. Determine o ID de `Instance` da máquina virtual na qual você deseja efetuar login. Execute o comando a seguir:

   ```
   bosh -e IBMCloudPrivate -d Bluemix vms
   ```

5. Assegure-se de que a chave privada esteja disponível para acessar a máquina virtual. Esta chave é a mesma chave que é especificada no `uiconfig.yml` como `openstack_key_pair_private`. Também é a mesma chave que é referenciada em `uiconfig.yml` pela chave `openstack_key_pair_name` e importada para o Openstack como um `Key Pair`.
   Se você precisar construir o arquivo de chave privado no Contêiner de concepção, será possível executar os comandos a seguir:

   ```
   export KEYNAME= ` bosh int /data/uiconfig.yml -- path /uiconfig/openstack_key_pair_name `
   bosh int /data/uiconfig.yml --path /uiconfig/openstack_key_pair_private > /data/${KEYNAME}.pem
   chmod 600  /data/${KEYNAME}.pem
   ```

6. Use o comando ssh BOSH com alguns parâmetros extras. Execute o comando a seguir:

   ```
   bosh -e IBMCloudPrivate -d Bluemix ssh <Instance ID>  --gw-private-key=<private key> --gw-host=
   <director IP>  --gw-user=vcap
   ```

Resolução de Problemas de Aplicativos

Resolva problemas que possam estar ocorrendo quando estiver configurando aplicativos.

- A implementação do aplicativo Cloud Foundry falha devido ao erro `EHOSTUNREACH`
- A implementação do aplicativo Docker no Cloud Foundry falha devido ao erro `no route to host`
- Os comandos cf push e log do Cloud Foundry retornam um erro
- O comando cf push do Cloud Foundry falha ao fazer download de buildpacks externos

A implementação do aplicativo Cloud Foundry falha devido ao erro EHOSTUNREACH

A implementação do aplicativo Cloud Foundry falha devido ao erro `EHOSTUNREACH`.

Sintomas

Ao implementar um aplicativo Cloud Foundry, um erro semelhante à mensagem a seguir é exibido:

```
-----> Building dependencies
       Installing node modules (package.json)
       npm ERR! Linux 4.4.0-75-generic
       npm ERR! argv "/tmp/app/vendor/node/bin/node" "/tmp/app/vendor/node/bin/npm" "install" "--
unsafe-perm" "--userconfig" "/tmp/app/.npmrc"
       npm ERR! node v6.12.2
       npm ERR! npm  v3.10.10
       npm ERR! code EHOSTUNREACH
       npm ERR! errno EHOSTUNREACH
       npm ERR! syscall connect

       npm ERR! connect EHOSTUNREACH 151.101.40.162:443
       npm ERR!
       npm ERR! If you need help, you may report this error at:
       npm ERR!     <https://github.com/npm/npm/issues>

       npm ERR! Please include the following file with any support request:
       npm ERR!     /tmp/app/npm-debug.log
-----> Build failed
       Some possible problems:

       - Node version not specified in package.json
       http://docs.cloudfoundry.org/buildpacks/node/node-tips.html

Failed to compile droplet
Exit status 223
Staging failed: Exited with status 223
Destroying container
Successfully destroyed container

FAILED
Error restarting application: BuildpackCompileFailed
```

Nesta mensagem, o aplicativo ou buildpack com falha que é usado pelo aplicativo está tentando acessar a Internet para fazer download do conteúdo. Se precisar usar um proxy HTTP para acessar a Internet, é necessário configurar seu aplicativo para usar o proxy.

Resolvendo o problema

1. Efetue login no Cloud Foundry:

   ```
   cf login
   ```

2. Mude para a organização e o espaço em que o aplicativo está implementado:

   ```
   cf target -o <org> -s <space>
   ```

3. Configure as variáveis do proxy HTTP para o aplicativo. Por exemplo, se o proxy for `http://myproxy.com:3128`, execute o comando a seguir:

   ```
   cf set-env myApp https_proxy "http://myproxy.com:3128"
   cf set-env myApp http_proxy "http://myproxy.com:3128"
   ```

4. Reinicie o aplicativo. Execute o comando a seguir:

   ```
   cf restart myApp
   ```

A implementação do aplicativo Docker no Cloud Foundry falha devido ao erro nenhuma rota para o host

A implementação do aplicativo Docker no Cloud Foundry falha devido ao erro `no route to host`.

Sintomas

Ao implementar um aplicativo Docker no Cloud Foundry, um erro semelhante à mensagem a seguir será exibido:

```
Creating container
Successfully created container
Staging...
Staging process started ...
Failed to talk to docker registry: Get https://registry-1.docker.io/v2/: dial tcp
54.152.209.167:443: getsockopt: no route to host
Failed to talk to docker registry: Get http://registry-1.docker.io/v2/: dial tcp 54.152.209.167:80:
getsockopt: no route to host
Staging process failed: Exit trace for group:
builder exited with error: failed to fetch metadata from [cloudfoundry/lattice-app] with tag
[latest] and insecure registries [] due to Get http://registry-1.docker.io/v2/: dial tcp
54.152.209.167:80: getsockopt: no route to host
Exit status 2
Staging Failed: Exited with status 2
Destroying container
Successfully destroyed container

FAILED
Error restarting application: StagingError
```

Nessa mensagem, o aplicativo com falha está tentando acessar a Internet para fazer download do conteúdo. Se você precisar usar um proxy HTTP para acessar a Internet, deverá configurar seu aplicativo para usar o proxy.

Resolvendo o problema

1. Efetue login no Cloud Foundry:

   ```
   cf login
   ```

2. Mude para a organização e o espaço em que o aplicativo está implementado:

   ```
   cf target -o <org> -s <space>
   ```

3. Envie o aplicativo Docker por push, mas não o inicie:

```
cf push myApp --docker-image --no-start <docker container image>
```

4. Configure as variáveis do proxy HTTP para o aplicativo. Por exemplo, se o proxy for `http://myproxy.com:3128`, execute o comando a seguir:

```
cf set-env myApp https_proxy "http://myproxy.com:3128"
cf set-env myApp http_proxy "http://myproxy.com:3128"
```

5. Inicie o aplicativo. Execute o comando a seguir:

```
cf restart myApp
```

Os comandos cf push e log do Cloud Foundry retornam um erro

Os comandos `cf push` e `cf log` do Cloud Foundry falham.

Sintomas

O `cf push APPLICATION_NAME` e o `cf log APPLICATION_NAME --recent` do Cloud Foundry retornam erros quando tentam alcançar o Loggregator do Cloud Foundry.

A execução de `cf push APPLICATION_NAME` exibe a mensagem de erro a seguir:

```
Warning: error tailing logs
```

A execução de `cf logs APPLICATION_NAME --recent` exibe a mensagem de erro a seguir:

```
unknown issue when making HTTP request to Loggregator
```

Resolvendo o problema

1. Efetue login no BOSH. Consulte Interfaces da linha de comandos para IBM® Cloud Private Cloud Foundry.

2. Determine qual instância de Loggregator está falhando. Para cada instância nats em sua instalação, execute o comando a seguir. Para instalações do desenvolvedor, verifique `nats/0` e para instalações corporativas, verifique `nats/0`, `nats/1` e `nats/2`.

```
bosh -e IBMCloudPrivate -d Bluemix ssh <nats> -c "tail
/var/vcap/sys/log/loggregator_trafficcontroller/loggregator_trafficcontroller.stderr.log"
```

   Em que `<nats>` é a instância a ser verificada.

   A saída de comando para a instância `nats` que causa o problema contém uma das mensagens de erro a seguir:

```
2017/12/05 18:37:12 Could not get app information: [Get http://api.local.bluemixx
.net/internal/log_access/f3c629e8-0bf0-4ecb-98a6-5dd42b707acb: dial tcp: lookup
api.local.bluemix.net on 127.0.0.1:53: no such host]
```

   ou

```
2017/12/05 18:45:04 Error while reading from stream (192.168.248.11:8082): rpc ee
rror: code = 1 desc = context canceled
2017/12/05 18:45:04 Unable to connect to doppler (192.168.248.11:8082): rpc erroo
r: code = 1 desc = context canceled
2017/12/05 18:45:04 Disconnecting from stream (192.168.248.10:8082) (doppler.diss
connect=false) (ctx.disconnect=1)
2017/12/05 18:45:04 Disconnecting from stream (192.168.248.9:8082) (doppler.discc
onnect=false) (ctx.disconnect=1)
2017/12/05 18:45:04 Disconnecting from stream (192.168.248.11:8082) (doppler.diss
connect=false) (ctx.disconnect=1)
```

3. Reinicie o Loggregator. Execute os comandos a seguir:

```
bosh -e IBMCloudPrivate -d Bluemix ssh <nats>
sudo su -
monit restart unbound
monit summary
monit restart loggregator_trafficcontroller
exit
```

Em que `<nats>` é a instância nats que produziu o erro.

4. Repita a etapa anterior para cada instância nats que exibiu a mensagem de erro.

Cloud Foundry O comando cf push falha ao fazer download de buildpacks externos

Os comandos Cloud Foundry `cf push -b https://<buildpack_path>` falham.

Sintomas

A mensagem de erro a seguir é exibida durante a implementação de um aplicativo que usa um buildpack externo.

```
Staging...
 Failed to clone git repository at https://github.com/cloudfoundry-community/jboss-buildpack.git
 Exit status 1
```

Resolvendo o problema

Conclua as etapas a seguir para identificar a causa raiz do problema:

1. Implemente um aplicativo simples.
2. Abra uma sessão no contêiner que contém o aplicativo: `cf ssh <simple_app>`.
3. Navegue para o diretório `tmp`: `cd /tmp`.
4. Clone o buildpack para o diretório `tmp`: `git clone <buildpack>`.
5. Verifique se há erros, conforme mostrado no exemplo a seguir:

```
git clone https://github.com/cloudfoundry-community/jboss-buildpack.git Clonando em 'jboss-
buildpack' ...
fatal: unable to access 'https://github.com/cloudfoundry-community/jboss-buildpack.git/':
Problem with the SSL CA cert (path? direitos de acesso?)
```

Emissão de certificado de CA SSL

Os certificados que estão sendo usados provavelmente são autoassinados. Conclua as etapas a seguir como uma solução alternativa do problema:

1. Faça download do buildpack em um local, como o contêiner de concepção, em que a CLI do Cloud Foundry está configurada.

2. Inclua os buildpacks. Para obter informações sobre como incluir buildpacks, consulte Incluir um buildpack. Certifique-se de escolher a posição correta para o buildpack ao colocar o buildpack na lista de buildpacks do Cloud Foundry atualmente disponíveis. Observe que o Cloud Foundry faz seleções de buildpack com base na extensão do aplicativo de pacote configurável. Por exemplo, se você deseja usar o buildpack-jboss e implementar um arquivo .war, deve-se verificar se o buildpack-jboss está posicionado antes de quaisquer outros buildpacks que também gerenciem arquivos .war.

   ```
   cf create-buildpack <BUILDPACK_NAME> <DOWNLOADED_BUILDPACK_PATH> <POSITION>
   ```

3. Use o comando a seguir para reimplementar o aplicativo:

   ```
   cf push -b <BUILDPACK_NAME>
   ```

Resolução de Problemas do Cloud Foundry Enterprise Environment

Saiba como isolar e resolver problemas com o Cloud Foundry Enterprise Environment.

- A implementação do Cloud Foundry Enterprise Environment falha
- O Stager está indisponível
- Não é possível desvincular um serviço OSB de um aplicativo Cloud Foundry}

A implementação do Cloud Foundry Enterprise Environment falha

A implementação do Cloud Foundry Enterprise Environment falha.

A implementação do Cloud Foundry Enterprise Environment falha devido a um arquivo .tgz inválido

Sintomas

`load_cloudpak.sh` falha com a mensagem:

```
manifest.json: nenhum arquivo ou diretório desse tipo
```

Resolvendo o problema

Um arquivo .tgz inválido foi especificado na linha de comandos. Especifique o arquivo `ibm-cfee-installer-installer-archive.tgz` correto.

A implementação do Cloud Foundry Enterprise Environment falha devido à configuração incorreta do Docker

Sintomas

`load_cloudpak.sh` falha com a mensagem:

```
Error response from daemon: Get https://9.37.33.88:8500/v2/: x509: certificate signed by unknown
authority
Unable to docker login to 9.37.33.88
```

Resolvendo o problema

O Docker não está configurado adequadamente. Consulte Instalando o software IBM no IBM Cloud Private.

A implementação do Cloud Foundry Enterprise Environment falha devido a nós do trabalhador

insuficientes

Sintomas

A ferramenta de implementação do Cloud Foundry mostra que o estado `Implementar o Cloud Foundry com base no Kubernetes` falhou. A mensagem de erro a seguir aparece no log:

```
[ERROR] Current number of worker node size: 4 is less than the required number of worker node size:
5 ( 2 control plane nodes and 3 cell nodes), exit...
```

Resolvendo o problema

O número de nós do trabalhador do Kubernetes não é suficiente para o número de instâncias do plano de controle mais o número de instâncias de célula que são necessárias para o Cloud Foundry Enterprise Environment. Aumente o número de nós do trabalhador do Kubernetes ou diminua o número de instâncias de célula ou de instâncias do plano de controle e tente novamente.

A implementação do Cloud Foundry Enterprise Environment falha porque a célula Diego não está iniciando

Sintomas

Os pods de célula Diego, em `namespace cf`, permanecem em `NotReadyState` para sempre e, eventualmente, a implementação falha.

Resolvendo o problema

Lista o estado dos pods do Cloud Foundry:

```
kubectl get  pods -n cf
```

Efetue login no pod ou pods de célula Diego que mostram `NotReadyState`:

```
kubectl exec -n cf -it diego-cell-0 /bin/bash
```

Verifique se `monit` está em execução:

```
monit summary
```

Inicie `monit` se ele não estiver em execução:

```
monit -lv
```

NOTA: Se você perceber este sintoma, conclua as etapas antes de a implementação do Cloud Foundry Enterprise Environment atingir o tempo limite e falhar.

Comandos variados

Exiba os Cloud Foundry Enterprise Environment pods

Use o comando a seguir:

```
kubectl get pods -n uaa
kubectl get pods -n cf
```

Verifique os logs de um pod de célula do Cloud Foundry Enterprise Environment

Use o comando a seguir:

```
kubectl log -n cf -c diego-cell diego-cell-0
```

Efetue login no pod do Cloud Foundry Enterprise Environment.

1. Determine o namespace e o nome do pod usando o comando na seção anterior e, em seguida, use o comando a seguir:

   ```
   kubectl exec -it -n <namespace> <pod name> bash
   ```

   Exemplo:

   ```
   kubectl exec -it -n cf diego-cell-0 bash
   ```

2. Use Cloud Foundry Enterprise Environment comandos e procedimentos de depuração padrão.

   ```
   sudo su -
   monit summary
   ls -al /var/vcap/sys/log
   ```

Verifique as configurações kube-dns para os domínios curinga

1. Edite as configurações de kube-dns.

   ```
   kubectl get configmap kube-dns -n kube-system -o=yaml
   ```

2. Edite as configurações de kube-dns. **Nota**: tenha cuidado ao editar esse arquivo, já que ele pode causar uma configuração incorreta de seu ambiente.

   ```
   kubectl edit configmap kube-dns -n kube-system
   ```

O Stager está indisponível

Siga as etapas para depurar um stager indisponível no ambiente Cloud Foundry Enterprise Environment.

Sintomas

- A API do Diego BBS não é acessível. Não é possível usar o `cf push app` e a preparação do aplicativo Cloud Foundry falha com o erro `O Stager está indisponível`, como no exemplo a seguir:

  ```
  Response code: 503
  CC code:       0
  CC error code:
  Request ID:    xxxxxxx-xxxxx-xxx-xxxx-xxxxxxxx
  ```

```
Description:   {
"description": "Stager is unavailable: execution expired",
"error_code": "CF-StagerUnavailable",
"code": 170010
}
Server error, status code: 503, error code: 170010, message: Stager is unavailable: execution
expired
```

- Ao executar `cf apps` e a coluna `instances` mostrar `?/<number of instances>`, por exemplo, `?/1`.

Resolvendo o problema

A mensagem de erro `O Stager está indisponível` geralmente significa que o Cloud Controller não pode se comunicar com a API do Diego BBS. Siga as etapas para recuperar o ambiente:

1. Certifique-se de que o Cloud Controller esteja disponível verificando o status do pod `diego-api-x` com o seguinte comando:

   ```
   $ kubectl get pod -n cf
   ```

   Por exemplo, a saída a seguir indica que dois pods diego-api-x estão em execução:

   ```
   NAME                              READY   STATUS      RESTARTS   AGE
   diego-api-0                       ?/2     Running     0          1d
   diego-api-1                       ?/2     Running     0          1d
   ```

2. Use o seguinte comando para acessar o pod `diego-api-0`:

   ```
   kubectl exec diego-api-0 -n cf -it -- bash
   ```

3. No pod `diego-api-0`, um erro semelhante ao seguinte texto indica que o Cloud Controller não pode acessar o BBS.

   ```
   nc -zv diego-api.cf.svc.cluster.local 8889
   ```

   Por exemplo, a saída a seguir indica que a conexão falhou:

   ```
   nc: connect to diego-api.cf.svc.cluster.local port 8889 (tcp) failed: Connection timed out
   ```

4. Saia do pod `diego-api-0` e, em seguida, exclua todos os pods `diego-api-x` usando os seguintes comandos. O Kubernetes recria os pods automaticamente.

   ```
   kubectl delete pod diego-api-0 -n cf
   ```

5. Espere até que todos os pods `diego-api-x` estejam em execução e 1/1 estejam prontos. Certifique-se de que um dos pods `diego-api-x` esteja rotulado como ativo usando o seguinte comando:

   ```
   kubectl get pod -n cf -L skiff-role-active
   ```

6. Repita as etapas 2 e 3 para verificar se o Cloud Controller pode acessar a API do BBS. Você deve obter uma saída semelhante ao seguinte texto:

   ```
   diego-api/0:/$ nc -zv diego-api-bbs.cf.svc.cluster.local 8889
   Connection to diego-api-bbs.cf.svc.cluster.local 8889 port [tcp/*] succeeded!
   ```

   O ambiente agora está pronto para a preparação do aplicativo Cloud Foundry.

Não é possível desvincular um serviço OSB de um aplicativo Cloud Foundry

Resolução de problemas para desvincular um serviço open service broker (OSB) de um aplicativo Cloud Foundry quando a liberação do Helm do broker de serviço é excluída do IBM® Cloud Private.

Sintomas

Quando estiver tentando desvincular um serviço no Cloud Foundry usando o comando `cf unbind-service`, você pode receber um erro semelhante à seguinte mensagem:

```
  root@cfee-5d495b885f-j9xhr:/tmp# cf unbind-service GetStartedJava-04171543 mongodb-04171543
  Unbinding app GetStartedJava-04171543 from service mongodb-04171543 in org org / space dev as
admin...
  Unexpected Response
  Response code: 502
  CC code:       0
  CC error code:
  Request ID:    0b02061434afa1b3714bc211f66c7224::89ad36f4-fae1-4834-a58c-6579262b7682
  Description:   {
    "description": "Service instance mongodb-04171543: The service broker could not be reached:
http://9.30.194.174:30015/v2/service_instances/1c02bfad-014b-4b95-86dc-
791d33b6169c/service_bindings/9ba27173-e909-454c-ab1a-04282bfdd26f?plan_id=3a41389b-739e-496d-87d0-
162e95bde385&service_id=d808cc2b-ed0a-41b5-aebc-d2dfd8d72801",
    "error_code": "CF-ServiceBrokerApiUnreachable",
    "code": 10001,
    "http": {
      "uri": "http://9.30.194.174:30015/v2/service_instances/1c02bfad-014b-4b95-86dc-
791d33b6169c/service_bindings/9ba27173-e909-454c-ab1a-04282bfdd26f?plan_id=3a41389b-739e-496d-87d0-
162e95bde385&service_id=d808cc2b-ed0a-41b5-aebc-d2dfd8d72801",
      "method": "DELETE"
    }
  }

  FAILED
```

Resolvendo o problema

A mensagem de erro geralmente ocorre quando a liberação do Helm do broker de serviço é excluída do IBM Cloud Private. Use as seguintes etapas para recuperar o ambiente:

1. Efetue login no IBM Cloud Private usando o cloudctl. Selecione o namespace no qual o Cloud Foundry Enterprise Environment é implementado usando o seguinte comando:

   ```
   cloudctl login -a https://9.30.194.174:8443 --skip-ssl-validation
   ```

2. Use o seguinte comando para localizar o nome do POD no qual o Cloud Foundry Enterprise Environment é implementado:

   ```
   kubectl get pods | grep cfee
   ```

3. Efetue login no pod e substitua `<POD_NAME>` pelo nome do POD encontrado na Etapa 2:

   ```
   kubectl exec -it <POD_NAME> -- bash
   ```

4. Abra o Ferramenta de implementação do Cloud Foundry.
5. Navegue para **Menu** > **3. Estados**.
6. Clique em `cfp-ext-osb-installer`.
7. Visualize os logs clicando no ícone de papel.
8. Nos logs, localize um comando semelhante ao seguinte texto:

   ```
   helm install /repo_local/ibm-osb-database-chart/1.0.0-009/ibm-osb-database-1.0.0.tgz --name
   cfee-osb --namespace default  --set brokerconfig.servicebrokersecret=cf-osb-broker-
   secret,brokerconfig.icpsecret=cf-osb-icp-
   secret,brokerconfig.externalClusterIp=9.30.194.174,brokerconfig.namespace=default,image=easy-
   glowworm-cloudfoundry-cluster.icp:8500/default/ibm/servicebroker-cf:1.0.0-009 --tls
   ```

9. Execute o comando encontrado na Etapa 8 no pod da Etapa 3.

Essas etapas reimplementam a liberação do Helm do broker de serviço para que seja possível desvincular com sucesso um serviço OSB de um aplicativo Cloud Foundry.

IBM Multicloud Manager

As tecnologias de contêineres transformam o modo como as empresas constroem aplicativos para suportar os modelos de desenvolvimento do Agile. Kubernetes é padrão para orquestrar e gerenciar contêineres. Com o IBM Multicloud Manager, as tecnologias de contêiner são mais fáceis de visualizar e de gerenciar.

O IBM Multicloud Manager fornece visibilidade do usuário, gerenciamento centrado no aplicativo (política, implementações, funcionamento, operações) e conformidade com base em política entre nuvens e clusters. Com o IBM Multicloud Manager, você

tem o controle de seus clusters do Kubernetes. É possível garantir que seus clusters estejam seguros, operando de forma eficiente e entregando os níveis de serviço que os aplicativos esperam.

Para obter mais informações sobre a liberação mais recente, consulte O que há de novo na versão 3.2.0.

Inventário de cluster

Depois de configurar o IBM Multicloud Manager, é possível consultar informações sobre todos os clusters que estão conectados ao sistema. Usando os rótulos de cluster, é possível organizar seus clusters de acordo com diferentes provedores em nuvem, regiões geográficas, data centers e o propósito funcional de clusters individuais. No console, é possível visualizar o status de funcionamento de pods, nós, volumes persistentes e aplicativos que são executados nesses clusters.

Operações e visibilidade multicluster

O IBM Multicloud Manager fornece uma maneira de executar consultas paralelas com relação a múltiplos clusters e agregar essas informações por vários critérios. As informações são aumentadas por meio de visualizações de tráfego de pod quase em tempo real usando o Weave Scope, que permite entender como os pods se intercomunicam.

Usando o IBM Multicloud Manager, você tem visualizações ricas de como os clusters operam dentro do ambiente. É possível ver os painéis de informações agregadas em múltiplos clusters, visualizar a topologia de recursos dentro desses clusters ou navegar em consoles de clusters individuais para obter visualizações mais detalhadas.

Consulte os exemplos a seguir de IBM Multicloud Manager uso:

- Visualize o funcionamento de todos os clusters em uma região específica.
- Saiba quantos nós estão inativos em todos os clusters.
- Visualize os pods no namespace em todos os clusters de desenvolvimento
- Visualizar todos os pods com falha em um data center específico

Recursos de aplicativo

Com o IBM Multicloud Manager, é possível implementar recursos de aplicativo em múltiplos clusters. O IBM Multicloud Manager suporta gráficos Helm como o modelo de implementação, estendendo o catálogo do IBM Cloud Private para o destino de múltiplos clusters para implementação. Isso complementa quaisquer pipelines de CI/CD existentes, permitindo aos operadores a capacidade de implementar gráficos de aplicativo que podem ser criados em mais de um cluster.

O IBM Multicloud Manager também permite definir modelos de aplicativo que combinam cargas de trabalho implementáveis juntamente com políticas de localização. Isso permite que você gerencie todos os componentes de um aplicativo como uma única unidade. As políticas de localização definem onde os componentes de aplicativo devem ser implementados e quantas réplicas devem existir.

Consulte Trabalhando com aplicativos do IBM Multicloud Manager para obter mais tópicos de aplicativos.

Políticas

O IBM Multicloud Manager permite verificar se seus clusters estão operando corretamente comparando a configuração atual de vários recursos com relação ao seu estado desejado. O sistema permite criar modelos de conformidade que podem verificar políticas com relação às funções ou objetos de pod dentro dos clusters. Consulte Trabalhando com políticas do IBM Multicloud Manager para obter informações de introdução.

Consulte Introdução do IBM Multicloud Manager e Preparando-se para o IBM Multicloud Manager para obter mais informações.

IBM Multicloud Manager introdução

À medida que as soluções corporativas mudam para múltiplos provedores em nuvem que usam clusters do Kubernetes no local e baseados em nuvem, os usuários precisam de um plano de controle de múltiplos clusters para gerenciar clusters do Kubernetes.

Consulte a documentação do produto a seguir para aprender sobre o IBM Multicloud Manager:

- IBM Multicloud Manager arquitetura
- Limitações e problemas conhecidos do IBM Multicloud Manager

IBM Multicloud ManagerArquitetura

O IBM Multicloud Manager consiste em vários componentes, que são usados para acessar e gerenciar seus clusters. Saiba mais sobre os componentes para o IBM Multicloud Manager.

O hub-cluster

O hub-cluster é o termo comum que é usado para definir o IBM Multicloud Manager controlador, que é um controlador central que é executado em um cluster do IBM Cloud Private 3.2.0.

O hub-cluster agrega informações de múltiplos clusters usando um modelo de solicitação de trabalho assíncrono. Com um banco de dados de gráfico, o hub-cluster mantém o estado dos clusters e dos aplicativos que são executados nele. O hub-cluster também usa o `etcd`, que é um armazenamento de valor de chave distribuído, para armazenar o estado das solicitações de trabalho e os resultados de múltiplos clusters e fornece um conjunto de APIs de REST para as várias funções que ele suporta.

O managed-cluster

O managed-cluster é usado para definir o IBM Multicloud Manager Klusterlet, que é o agente que é responsável por um único cluster do Kubernetes. O managed-cluster inicia uma conexão com o hub-cluster, recebe solicitações de trabalho, aplica essas solicitações de trabalho e, em seguida, retorna os resultados. O managed-cluster se conecta a vários serviços dentro do cluster para operações, incluindo o serviço de API do Kuberenetes, o serviço do Tiller (Helm) e o Weave para topologia.

Consulte Visão geral da configuração do IBM Multicloud Manager para obter informações de configuração e de importação.

IBM Multicloud Manager Recursos de aplicativos

Depois de configurar um hub-cluster e um managed-cluster do IBM Multicloud Manager, será possível visualizar e implementar aplicativos com recursos de aplicativos. Seu *Aplicativo* é usado apenas para *visualizar* seu recurso, enquanto que outros exemplos de recurso de aplicativo são usados para implementação. Um aplicativo de múltiplos clusters usa uma especificação do Kubernetes, mas com automação adicional da implementação e do gerenciamento de ciclo de vida de recursos para clusters individuais. É possível incluir `PlacementPolicy` para implementar recursos de aplicativo.

Consulte Trabalhando com aplicativos do IBM Multicloud Manager para obter mais tópicos de aplicativos.

Documentos de política do IBM Multicloud Manager

Depois de configurar um hub-cluster e um managed-cluster do IBM Multicloud Manager, será possível definir a conformidade e as políticas do IBM Multicloud Manager com modelos. O PlacementPolicy define seus clusters aos quais o documento de política é aplicado e o PlacementBinding liga seu cluster a uma política. Para obter mais detalhes sobre políticas de conformidade, consulte Trabalhando com a conformidade do IBM Multicloud Manager.

Consulte Preparando-se para o IBM Multicloud Manager para preparar seu cluster e obter informações de configuração.

Limitações e problemas conhecidos do IBM Multicloud Manager

Revise os problemas conhecidos para IBM Multicloud Manager

- Limitações e problemas conhecidos do IBM Multicloud Manager
  - Não é possível criar uma liberação do Helm em um cluster remoto
  - Os aplicativos falham ao serem instalados durante a implementação do Helm
  - Ao instalar o PPA, as imagens são transferidas por download do registro errado

Não é possível criar uma liberação do Helm em um cluster remoto

Não é possível implementar gráficos Helm que contenham imagens em um cluster remoto. Para corrigir esse erro, deve-se configurar `ClusterImagePolicy`. Execute o comando a seguir para configurar a `ClusterImagePolicy`:

```
apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
kind: ClusterImagePolicy
metadata:
  annotations:
    helm.sh/hook: post-install
```

```
    helm.sh/hook-weight: "1"
  name: ibmcloud-default-cluster-image-policy
spec:
  repositories:
  - name: <repo_name>
```

Os aplicativos falham ao serem instalados durante a implementação do Helm

Os aplicativos falham ao serem instalados durante a implementação quando o ClusterImagePolicy não está configurado.

Nota: Certifique-se de configurar o ClusterImagePolicy. Visualize a seção *Não é possível criar uma liberação do Helm em um cluster remoto* para obter informações sobre como configurar a política.

Para corrigir esse erro, reinstale seu aplicativo seguindo as tarefas:

1. Verifique o status de seu aplicativo executando o comando a seguir:

   ```
   helm list --tls
   ```

2. Para excluir seu aplicativo, execute o comando a seguir:

   ```
   helm delete releaseName -- purge
   ```

3. Edite e localize o `ClusterImagePolicy` para enviar por push suas imagens para seu aplicativo. Execute o comando a seguir:

   ```
   kubectl get clusterimagepolicy
   ```

4. Edite o ClusterImagePolicy ao executar o comando a seguir:

   ```
   kubectl edit clusterimagepolicy <policyname>
   ```

5. Reinstale seu aplicativo. Execute o comando a seguir:

   ```
   helm install chartName
   ```

Para obter mais detalhes, consulte o Problema da comunidade do Helm.

Ao instalar o PPA, as imagens são transferidas por download do registro errado

Ao instalar o Passport Archive (*PPA*), as imagens são transferidas por download do registro errado. O registro padrão é `mycluster`.

Deve-se especificar o registro no PPA.

Nota: certifique-se de efetuar login em seu cluster no Docker executando o comando a seguir:

```
docker login <cluster_CA_domain>:8500/kube-system
```

Visão Geral da Configuração do IBM Multicloud Manager

Revise os procedimentos de configuração para o IBM Multicloud Manager e as opções de multinuvem para a importação do IBM Multicloud Manager.

- Preparando-se para o IBM Multicloud Manager
- Configurando o IBM Multicloud Manager durante a instalação do IBM Cloud Private
- Configurando o IBM Multicloud Manager após a instalação do IBM Cloud Private
- Instalando o pacote do IBM Multicloud Manager opcional
- Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager

Preparando-se para a instalação do IBM Multicloud Manager

Antes de instalar o IBM Multicloud Manager, revise os requisitos do sistema e as informações da porta para preparar seu cluster. Depois de preparar seus clusters, consulte a IBM Multicloud ManagerVisão geral de configuração para obter mais informações de configuração para o hub e o managed-cluster. Consulte Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager para obter informações sobre como importar managed-clusters.

Requisitos do Sistema

Os requisitos do IBM Cloud Private estão localizados na documentação do produto do IBM Cloud Private em Requisitos do sistema.

Para o IBM Multicloud Manager, são necessários pelo menos os valores a seguir:

- ETCD: 1GB

Consulte os requisitos mínimos a seguir para seu hub-cluster do IBM Multicloud Manager:

- CPU: 16 núcleos
- Memória: 32 GB
- Espaço em disco: 300 GB

À medida que você inclui mais clusters, seus requisitos de resiliência aumentam. Para ajudar a determinar a quantidade de capacidade que você precisa para ambientes de pequeno, médio ou grande porte, consulte Dimensionando seu cluster do IBM Cloud Private.

Portas Obrigatórias

Consulte a seguinte lista de portas necessárias que devem estar disponíveis para comunicação bidirecional entre seu cluster do hub e o cluster gerenciado.

Tipos de acesso de porta

- Interno - a porta deve ser aberta para permitir conexões dentro do cluster.
- Externo - a porta deve ser aberta para permitir conexões de fora do cluster.

Portas necessárias para o cluster do hub do IBM Multicloud Manager

A tabela a seguir lista as portas que precisam estar abertas para comunicação:

Porta	Utilização
8001	padrão para o cluster gerenciado comunicar-se com a porta do servidor da API do Kubernetes no cluster do hub
8500	padrão para que o managed-cluster se comunique com o registro do

Docker do IBM Cloud Private no hub-cluster|

Portas necessárias para o cluster gerenciado do IBM Multicloud Manager no IBM Cloud Private

A tabela a seguir lista as portas que precisam estar abertas para comunicação:

Porta	Utilização
443	padrão para que o hub-cluster se comunique com o serviço Klusterlet no

ingresso nginx do IBM Cloud Private|

Configurando o IBM Multicloud Manager durante a instalação do

IBM Cloud Private

É possível configurar o IBM Multicloud Manager durante a instalação do IBM Cloud Private customizando seu arquivo `config.yaml`. Consulte Customizando o cluster com o arquivo config.yaml para saber mais sobre as definições de configuração que estão disponíveis durante a instalação. Para obter mais tópicos do IBM Multicloud Manager e, para assegurar que seus clusters estejam preparados, consulte Preparando-se para a instalação do IBM Multicloud Manager.

- Configurando o hub-cluster do IBM Multicloud Manager com o arquivo config.yaml
- Configurando o managed-cluster do IBM Multicloud Manager com o arquivo config.yaml

Configurando o hub-cluster do IBM Multicloud Manager com o arquivo

`config.yaml`

Siga o processo para mudar suas configurações em seu arquivo `config.yaml`, que está localizado na pasta `/<installation_directory>/cluster`.

Por padrão, a opção `multicluster-hub` é `enabled` e a opção `single_cluster_mode` é `true`, mas não é possível usar o IBM Multicloud Manager com a configuração padrão `single_cluster_mode`.

1. Localize a opção `single_cluster_mode` no arquivo `config.yaml` e configure o valor como `false`, conforme exibido no exemplo a seguir:

   ```
   single_cluster_mode: false
   ```

2. Opcional: para usar o armazenamento persistente, é necessário configurar o volume de persistência local para o ETCD do IBM Multicloud Manager.

3. Crie a configuração a seguir para `multicluster-hub` incluindo a sub-rotina no arquivo `config.yaml`. É possível incluir o valor em qualquer lugar fora de sua seção `management_services`:

   ```
   multicluster-hub:
    etcd:
      persistence: true
      localPath: /var/lib/etcd-mcm
   ```

4. Efetue login em seu nó de gerenciamento e crie o diretório `/var/lib/etcd-mcm`. Deve-se repetir esta etapa para todos os seus nós de gerenciamento.

Opcional: também é possível gerenciar seu hub-cluster com o procedimento a seguir, que ativa o `multicluster-endpoint`.

Configurando o managed-cluster do IBM Multicloud Manager com o arquivo

`config.yaml`

Continue com o procedimento para ativar o `multicluster-endpoint` em seu cluster.

1. No arquivo `config.yaml` para o novo cluster do IBM Cloud Private, que está localizado na pasta `/<installation_directory>/cluster`, ative o `multicluster-endpoint`, como no seguinte exemplo:

   ```
   management_services:
    multicluster-endpoint: enabled
   ```

2. Continue para criar a sub-rotina com as seguintes configurações para `multicluster-endpoint`:

   ```
   multicluster-endpoint:
    global:
      clusterName: "{{ cluster_name }}"
      clusterNamespace: "{{ cluster_name }}"
    clusterLabels:
      environment: "Dev"
      region: "US"
      datacenter: "toronto"
      owner: "marketing"
    operator:
      bootstrapConfig:
        hub0:
          name: hub0
          secret: kube-system/klusterlet-bootstrap
        hub1:
          name: null
          secret: null
    klusterlet:
      host: null
    prometheusIntegration:
      enabled: true
    policy:
      cemIntegration: false
    topology:
      enabled: true
    serviceRegistry:
      enabled: true
      dnsSuffix: "mcm.svc"
      plugins: "kube-service"
   ```

3. Salve e saia do arquivo. Conclua o procedimento de instalação.

Configurando a instalação do IBM Multicloud Manager IBM Cloud Private

Se você já instalou o IBM Cloud Private com a opção `single_cluster_mode` configurada para o valor padrão `false`, não será possível usar o IBM Multicloud Manager. No entanto, é possível ativar e usar o IBM Multicloud Manager após a instalação. Para obter mais tópicos do IBM Multicloud Manager e, para assegurar que seus clusters estejam preparados, consulte Preparando-se para a instalação do IBM Multicloud Manager.

Configurando o hub-cluster do IBM Multicloud Manager após a instalação

1. Efetue login na console de gerenciamento do IBM Cloud Private e clique em **Cargas de Trabalho > Liberações do Helm**. Localize o `multicluster-hub`, que é o nome da liberação do hub-cluster.
2. Clique em **Fazer upgrade** para a liberação do multicluster-hub e marque **Ativar back-end > Ativar ETCD** e outras funcionalidades.
3. Atualize a console de gerenciamento do IBM Cloud Private.
4. Visualize a console de gerenciamento do IBM Cloud Private, na qual o *Multicloud Manager* é exibido na página *Introdução*.

Configurando o managed-cluster do IBM Multicloud Manager

Configure os managed-clusters com o comando `cloudctl mc cluster import` e com os arquivos de configuração. É possível importar clusters de diferentes provedores de nuvem do Kubernetes, incluindo o IBM Cloud Private.

Para saber mais sobre as opções de importação, consulte Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager.

Instalando pacotes do IBM Multicloud Manager opcionais

É possível instalar o IBM Multicloud Manager para usar pacotes opcionais.

- Pré-requisitos
- Carregando o IBM Multicloud Manager archive PPA
- Pacotes opcionais do IBM Multicloud Manager

Pré-requisito

- Você precisa de acesso a um ambiente do IBM Cloud Private . Para obter mais detalhes sobre o IBM Cloud Private e a configuração de seu ambiente, consulte a Visão geral.
- Você deve instalar o Docker. Para instalar o Docker, consulte Instalar o Docker.
- É necessário instalar a ferramenta de linha de comandos do Kubernetes. Para instalar o `kubectl`, consulte Instalando a CLI do Kubernetes (kubectl)l.
- Você deve instalar o Helm. Para obter mais informações, consulte Instalando a CLI do Helm (helm) para IBM Cloud Private para instalar o Helm.
- Você deve instalar a CLI do IBM Cloud Private , `cloudctl`. Para obter mais informações, consulte Instalando a CLI do IBM Cloud Private para instalar a CLI.

  Nota: é possível fazer download do arquivo de instalação para as ferramentas de CLI a partir da console de gerenciamento do IBM Cloud Private.

Carregando o IBM Multicloud Manager Archive de PPA

1. Execute o comando a seguir para efetuar login no registro de imagem do Docker do IBM Cloud Private:

   ```
   login docker < cluster_ca_domain>: 8500
   ```

   Para configurar a autenticação para acessar o registro de imagem privado de fora de seu cluster do IBM Cloud Private, consulte Configurando a autenticação para a CLI do Docker. Para configurar em sua máquina, conclua as etapas adicionais que são necessárias para configurar a autenticação.

2. Efetue login na CLI do IBM Cloud Private ou no `cloudctl`, para configurar o `helm` e o `kubectl`. Execute o comando a seguir:

   ```
   cloudctl login -a https://<cluster_ca_domain>:8443 --skip-ssl-validation
   ```

3. Carregue os archives do Passport Advantage (PPA).

   Execute o comando a seguir para descompactar o archive:

   ```
   tar zxvf mcm-3.2.tgz
   ```

   Em seguida, execute o comando a seguir para carregar o archive PPA:

   ```
   cloudctl catalog load-ppa-archive -a mcm-3.2/mcm-ppa-3.2.tgz --registry
   <cluster_ca_domain>:8500/kube-system --username <username> --password <password>
   ```

   Veja que há os seguintes pacotes em seu diretório:

   ```
   cem-mcm-3.2-ppa.tar.gz
   mcm-3.2-ppa-alertmanager-alerttargetcontroller.tar.gz
   mcm-ppa-3.2.tgz
   ```

4. Execute o comando a seguir para descompactar o archive:

   ```
   tar zxvf mcm-optional-components-X86-64-3.2.tgz
   ```

   Veja os arquivos a seguir:

   ```
   federation-v2-0.0.3-amd64.tgz
   ibm-argocd-3.1.2-amd64.tgz
   ```

Pacotes opcionais do IBM Multicloud Manager

Carregue os pacotes opcionais a seguir:

Federation-v2

```
  cloudctl catalog load-ppa-archive -a mcm-optional-components-X86-64-3.2/federation-v2-0.0.3-
amd64.tgz --registry <cluster_ca_domain>:8500/kube-system --username <username> --password
<password>
```

Recursos do GitOps

```
  cloudctl catalog load-ppa-archive -a mcm-optional-components-X86-64-3.2/ibm-argocd-3.1.2-amd64.tgz
--registry <cluster_ca_domain>:8500/kube-system --username <username> --password <password>
```

Gerenciamento de eventos para o IBM Multicloud Manager

```
  cloudctl catalog load-ppa-archive -a mcm-3.2/cem-mcm-3.2-ppa.tar.gz --registry
<cluster_ca_domain>:8500/kube-system --username <username> --password <password>
```

Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager

É possível importar clusters de diferentes provedores de nuvem do Kubernetes, incluindo o IBM Cloud Private. Depois de configurar seu arquivo e executar o `cloudctl mc cluster import`, o cluster de destino se torna um managed-cluster para o hub-cluster do IBM Multicloud Manager.

Escolha dentre as instruções a seguir para configurar seu managed-cluster:

- Importando um cluster do IBM Cloud Private
- Importando um cluster do IBM Cloud Private with OpenShift
- Importando um cluster do IBM Cloud Kubernetes Service
- Importando um cluster do Amazon Elastic Container Service for Kubernetes
- Importando um cluster do Azure Kubernetes Service
- Importando um cluster do Google Kubernetes Engine
- Importando um cluster do OpenShift
- Removendo um managed-cluster importado

Consulte a Visão geral da configuração do IBM Multicloud Manager para obter mais tópicos.

Importando um cluster do IBM Cloud Private

Depois de instalar seu IBM Cloud Private com o IBM Multicloud Manager ativado, configure seu arquivo e execute `cloudctl mc cluster import` para criar um managed-cluster.

- Pré-requisitos
- Prepare-se para importação
- Crie seu arquivo `config.yaml` do IBM Multicloud Manager
- Importando um cluster
- Pós-importação

Pré-requisito

- Deve-se ter um hub-cluster do IBM Multicloud Manager, que é um cluster do IBM Cloud Private com o serviço de gerenciamento `multicluster-hub` ativado e com `single_cluster_mode` configurado como `false`. Para obter mais detalhes sobre o IBM Cloud Private e sobre a configuração de seu ambiente, consulte Configurando o IBM Multicloud Manager.
- Deve-se ter um cluster do IBM Cloud Private que você deseja que seja gerenciado pelo hub-cluster do IBM Multicloud Manager.
- Você deve instalar o Docker. Para instalar o Docker, consulte Instalar o Docker.
- É necessário instalar a CLI do Kubernetes, `kubectl`. Para instalar o `kubectl`, consulte Instalando a CLI do Kubernetes (kubectl).
- Você deve instalar a CLI do IBM Cloud Private , `cloudctl`. Para obter mais informações, consulte Instalando a CLI do IBM Cloud Private para instalar a CLI.

  Nota: faça download do arquivo de instalação para as ferramentas de CLI a partir da console de gerenciamento do IBM Cloud Private.

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

   ```
   export KUBECONFIG=kubeconfig
   ```

3. Em um navegador, como um administrador de cluster, efetue login na console de gerenciamento do IBM Cloud Private para o managed-cluster de destino.
4. Selecione o ícone do usuário e, em seguida, clique em **Configurar cliente**. Os detalhes da configuração de cluster são exibidos e são semelhantes ao código a seguir:

```
  kubectl config set-cluster {cluster_name} --server=<Cluster Master Host>:<Cluster Master API Port>
--insecure-skip-tls-verify=true
  kubectl config set-context {cluster_name}-context --cluster={cluster_name}
  kubectl config set-credentials {username} --token={token}
  kubectl config set-context {cluster_name}-context --user={username} --namespace=default
  kubectl config use-context {cluster_name}-context
```

1. Clique em **Copiar para a área de transferência** para copiar os comandos de configuração `kubectl`.
2. Volte para o mesmo terminal da Etapa 1 e cole os comandos de configuração. Liste os arquivos em seu diretório e confirme se o arquivo `kubeconfig` foi criado.

3. Verifique o conteúdo do arquivo `kubeconfig`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

```
apiVersion: v1
clusters:
- cluster:
   insecure-skip-tls-verify: true
   server: <targeted-managed-cluster-kubernete-api-server>
 name: <cluster_name>
contexts:
- context:
   cluster: <cluster_name>
   namespace: default
   user: <username>
 name: <cluster_name>-context
current-context: <cluster_name>-context
kind: Config
preferences: {}
users:
- name: <username>
 user:
   token:  <authentication token>
```

4. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.

5. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

```
unset KUBECONFIG
```

Crie seu arquivo de configuração `cluster-import.yaml` do

IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.

2. Efetue login em seu *hub*-cluster com `cloudctl login` com o comando a seguir:

```
cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port>
--skip-ssl-validation-->
```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub.

```
cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
```

1. Opcional: abra seu `cluster-import.yaml` e configure a seção a seguir se você estiver direcionando para um cluster que não tenha acesso ao DockerHub público. Se você tiver acesso, ignore esta etapa:

   Para o IBM Cloud Private EE, é possível usar o registro do Docker privado do IBM Cloud Private.

```
inception_image: <cluster_CA_domain>:8500/ibmcom/icp-inception:3.2.0-ee
image_repo: <cluster_CA_domain>:8500/ibmcom
private_registry_enabled: true
docker_username: <username>
docker_password: <password>
```

2. Configure os parâmetros a seguir no arquivo `cluster-import.yaml`:

   - `default_admin_user`: o nome do usuário administrador do cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

   Consulte a seção *Configurações do multicluster-endpoint* de Customizando o cluster com o arquivo config.yaml para obter mais parâmetros.

   Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`.

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

3. Execute o comando a seguir para importar no managed-cluster de destino:

   ```
   cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--
   cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
   ```

4. Verifique se o cluster foi importado com sucesso.
   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.
   - Localize o novo managed-cluster importado na lista.
   - Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Pós-importação

Se você concluiu a etapa 4 opcional na seção *Criar seu arquivo de configuração `cluster-import.yaml` do IBM Multicloud Manager* para configurar seu Docker privado em seu `cluster-import.yaml`, será necessário remover as credenciais do ConfigMap no hub-cluster.

1. Efetue login em seu *hub*-cluster com `cloudctl login`. Execute o comando a seguir:

   ```
   cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port>
   --skip-ssl-validation-->
   ```

2. Edite o ConfigMap `bootstrap-config` para o managed-cluster de destino. Execute o comando a seguir:

   ```
   kubectl edit configmap -n <cluster_namespace> <cluster_name>-bootstrap-config
   ```

3. Remova as informações de autenticação privadas do Registro do Docker, conforme exibido no exemplo a seguir:

   ```
   docker_username: <docker_username>
   docker_password: <docker_password>
   ```

4. Salve e saia.

Importando um cluster do IBM Cloud Private with OpenShift

Pré-requisito

- Deve-se ter um hub-cluster do IBM Multicloud Manager, que é um cluster do IBM Cloud Private com o serviço de gerenciamento `multicluster-hub` ativado e com `single_cluster_mode` configurado como `false`. Para obter mais detalhes sobre o IBM Cloud Private e sobre a configuração de seu ambiente, consulte Configurando o IBM Multicloud Manager.
- Deve-se ter acesso a um ambiente do IBM Cloud Private com o OpenShift. Para obter informações adicionais sobre o IBM Cloud Private with OpenShift e como configurar seu ambiente, consulte a Visão geral do IBM Cloud Private with OpenShift.
- Você deve instalar o Docker. Para instalar o Docker, consulte Instalar o Docker.
- É necessário instalar a ferramenta de linha de comandos do Kubernetes, `kubectl`. Para instalar `kubectl`, consulte Instalar e configurar o kubectl.
- Você deve instalar a CLI do IBM Cloud Private , `cloudctl`. Para obter mais informações, consulte Instalando a CLI do IBM Cloud Private para instalar a CLI.
- Deve-se instalar a CLI do OpenShift, `oc`. Para obter mais informações, consulte Introdução à CLI.

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

   ```
   export KUBECONFIG=kubeconfig
   ```

3. Efetue login em seu IBM Cloud Private with OpenShift com `oc login`:

   ```
   oc login <OpenShift console URL>
   ```

4. Verifique o conteúdo do arquivo `kubeconfig`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

   ```
   apiVersion: v1
   clusters:
   - cluster:
      insecure-skip-tls-verify: true
      server: <targeted-managed-cluster-kubernete-api-server>
    name: <cluster-name>
   contexts:
   - context:
      cluster: <cluster-name>
      namespace: default
      user: <username>
    name: <context-name>
   current-context: <context-name>
   kind: Config
   preferences: {}
   users:
   - name: <username>
    user:
      token: <token>
   ```

1. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.
2. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

   ```
   unset KUBECONFIG
   ```

Crie seu arquivo de configuração

`cluster-import.yaml` do IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`. Execute o comando a seguir:

   ```
   cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-
   validation-->
   ```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub.

   ```
   cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
   ```

4. Abra seu `cluster-import.yaml` e configure a seção a seguir se você estiver direcionando para um cluster que não tenha acesso ao DockerHub público. Se você tiver acesso, ignore esta etapa.

   Para o IBM Cloud Private with OpenShift EE, é possível usar o registro de docker privado interno do OpenShift.

   ```
   inception_image: docker-registry.default.svc:5000/ibmcom/icp-inception:3.2.0-rhel-ee
   image_repo: docker-registry.default.svc:5000/ibmcom
   ```

5. Configure os parâmetros a seguir no arquivo `cluster-import.yaml`.

   - `default_admin_user`: o nome do usuário administrador de cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`.

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

3. Execute o comando a seguir para importar no managed-cluster de destino:

   ```
   cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--
   cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
   ```

4. Verifique se o cluster foi importado com sucesso.
   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.
   - Localize o novo managed-cluster importado na lista.
   - Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Pós-importação

Se você concluiu a etapa 4 opcional na seção *Criar seu arquivo de configuração `cluster-import.yaml` do IBM Multicloud Manager* para configurar seu Docker privado em seu `cluster-import.yaml`, será necessário remover as credenciais do ConfigMap no hub-cluster.

1. Efetue login em seu *hub*-cluster com `cloudctl login`. Execute o comando a seguir:

   ```
   cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-
   validation-->
   ```

2. Edite o ConfigMap `bootstrap-config` para o managed-cluster de destino. Execute o comando a seguir:

   ```
   kubectl edit configmap -n <cluster_namespace> <cluster_name>-bootstrap-config
   ```

3. Remova as informações de autenticação privadas do Registro do Docker, conforme exibido no exemplo a seguir:

   ```
   docker_username: <docker_username>
   docker_password: <docker_password>
   ```

4. Salve e saia.

Importando um cluster do IBM Cloud Kubernetes Service

Siga o procedimento para importar um cluster do IBM Cloud Kubernetes Service. Para obter mais informações sobre o IBM Cloud Kubernetes Service, consulte Introdução ao serviço do IBM Cloud Kubernetes Service.

Pré-requisito

- Deve-se ter um hub-cluster do IBM Multicloud Manager, que é um cluster do IBM Cloud Private com `multi-cluster hub` ativado e com `single_cluster_mode` configurado como `false`.

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Efetue login no IBM Cloud e, em seguida, clique em **Kubernetes** para acessar o painel.
2. Selecione **Clusters** a partir da navegação e, em seguida, selecione seu cluster de destino na lista que é exibida.
3. Na página Visão Geral, clique em **Acesso**.
4. Clique em **download** para obter o arquivo `kubeconfig.zip` para seu cluster. Coloque o arquivo no `<import_config_directory>`.
5. Em um terminal, acesse o diretório `<import_config_directory>`.
6. Execute o comando a seguir para extrair o arquivo `kube-config-<zone>-<cluster_name>.yaml` e o arquivo `ca-<zone>-<cluster_name>.pem` associado de seu arquivo compactado `kubeconfig` e coloque tudo em seu diretório atual:

   ```
   unzip -j kubeconfig.zip
   ```

7. Verifique o conteúdo do arquivo `kube-config-<zone>-<cluster_name>.yaml`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

   ```
   apiVersion: v1
   clusters:
   - cluster:
      certificate-authority: ca-<zone>-<cluster_name>.pem
      server: <target_cluster_kuberentes_api_seerver>
    name: <cluster_name>
   contexts:
   - context:
      cluster: <cluster_name>
      namespace: default
      user: <username>
    name:  <cluster_name>
   current-context: <cluster_name>
   kind: Config
   preferences: {}
   users:
   - name: <username>
    user:
      auth-provider:
        config:
          client-id: kube
          client-secret: kube
          id-token: <id-token>
          idp-issuer-url: https://iam.bluemix.net/identity
          refresh-token: <refresh-token>
        name: oidc
   ```

8. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

   ```
   export KUBECONFIG=kube-config-<zone>-<cluster_name>.yaml
   ```

9. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.
10. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

    ```
    unset KUBECONFIG
    ```

Crie seu arquivo de configuração `cluster-import.yaml` do

IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`:

   ```
   cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-
   validation-->
   ```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub:

```
cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
```

4. Abra o arquivo `cluster-import.yaml` e configure os parâmetros a seguir:

   - `default_admin_user`: o nome do usuário administrador de cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

   Consulte a seção *Configurações do multicluster-endpoint* de Customizando o cluster com o arquivo config.yaml para obter mais parâmetros.

   Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login` com o comando a seguir:

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

3. Execute o comando a seguir para importar no managed-cluster de destino:

   ```
   cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--
   cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
   ```

4. Verifique se o cluster foi importado com sucesso.

   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.
   - Localize o novo managed-cluster importado na lista.
   - Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Importando um cluster do Amazon Elastic Container Service for Kubernetes

Siga o procedimento para importar um cluster do Amazon Elastic Container Service for Kubernetes. Consulte Amazon Elastic Container Service for Kubernetes para obter mais informações sobre o serviço público do Kubernetes.

- Pré-requisitos
- Prepare-se para importação
- Crie seu arquivo config.yaml do IBM Multicloud Manager
- Importando um cluster

Pré-requisito

- Deve-se ter um hub-cluster do IBM Multicloud Manager, que é um cluster do IBM Cloud Private com `multi-cluster hub` ativado e com `single_cluster_mode` configurado como `false`.
- É necessário instalar a CLI para autenticar-se posteriormente neste procedimento. Consulte [Instalar a CLI do AWS] (https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html) para obter

instruções para instalar a CLI.

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Autentique-se à CLI do AWS com o comando a seguir:

```
aws configure
```

3. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

```
export KUBECONFIG=kubeconfig
```

4. Obtenha o `kubeconfig` com o comando a seguir:

```
 aws eks update-kubeconfig --name <cluster-name>
```

5. Liste os arquivos em seu diretório e confirme se o arquivo `kubeconfig` foi criado.

6. Verifique o conteúdo do arquivo `kubeconfig`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: <certificate-authority-data>
  server: <target-managed-cluster-kubernete-api-server>
name: <cluster-name>
contexts:
- context:
 cluster: <cluster-name>
 user: <user-profile-name>
name: <context-name>
current-context: <context-name>
kind: Config
preferences: {}
users:
- name: <user-profile-name>
user:
exec:
 apiVersion: client.authentication.k8s.io/v1alpha1
 args:
 - token
 - -i
 - <cluster-name>
 command: aws-iam-authenticator
```

7. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.

8. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

```
unset KUBECONFIG
```

Crie seu arquivo de configuração `cluster-import.yaml` do

IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.

2. Efetue login em seu *hub*-cluster com `cloudctl login`.

```
cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-
validation-->
```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub:

```
cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
```

1. Abra o arquivo `cluster-import.yaml` e configure os parâmetros a seguir:

   - `default_admin_user`: o nome do usuário administrador de cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

   Consulte a seção *Configurações do multicluster-endpoint* de Customizando o cluster com o arquivo config.yaml para obter mais parâmetros.

Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`.

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

3. Execute o comando a seguir para importar no managed-cluster de destino:

   ```
   cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--
   cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
   ```

4. Verifique se o cluster foi importado com sucesso.
   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.
   - Localize o novo managed-cluster importado na lista.
   - Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Importando um cluster do Azure Kubernetes Service

Siga o procedimento para importar um cluster do Azure Kubernetes Service. Consulte Azure Kubernetes Service para obter mais informações sobre o serviço público do Kubernetes.

- Pré-requisitos
- Prepare-se para importação
- Crie seu arquivo `config.yaml` do IBM Multicloud Manager
- Importando um cluster

Pré-requisito

- Deve-se ter um hub-cluster do IBM Multicloud Manager, que é um cluster do IBM Cloud Private com `multi-cluster hub` ativado e com `single_cluster_mode` configurado como `false`.
- É necessário instalar a CLI do Azure e autenticar. Consulte Azure Kubernetes Service para obter instruções para instalar a CLI.
- Deve-se ativar o *monitoramento* e o *RBAC* no cluster do Azure Kubernetes Service.

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login no Azure Cloud Platform com o comando a seguir, que abre um navegador para autenticação:

   ```
   az login
   ```

3. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

   ```
   export KUBECONFIG=kubeconfig
   ```

4. Obtenha suas informações de acesso ao cluster executando o comando a seguir:

   ```
   az aks get-credentials \
   --resource-group <resource_group> \
   --name <cluster_name> \
   --file ./kubeconfig
   ```

5. Verifique o conteúdo do arquivo `kubeconfig`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: <ca-data>
    server: <targeted-managed-cluster-kubernete-api-server>
  name: <cluster_name>
contexts:
- context:
    cluster: <cluster_name>
    user: <username>
  name: <cluster_name>-context
current-context: <cluster_name>-context
kind: Config
preferences: {}
users:
- name: <username>
  user:
    client-certificate-data: <client-certificate-data>
    client-key-data: <client-key-data>
    token: <token>
```

6. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.

7. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

```
unset KUBECONFIG
```

Crie seu arquivo de configuração `cluster-import.yaml` do

IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.

2. Efetue login em seu *hub*-cluster com `cloudctl login` com o comando a seguir:

```
cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub:

```
cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
```

4. Abra o arquivo `cluster-import.yaml` e configure os parâmetros a seguir:

   - `default_admin_user`: o nome do usuário administrador de cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

   Consulte a seção *Configurações do multicluster-endpoint* de Customizando o cluster com o arquivo config.yaml para obter mais parâmetros.

Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.

2. Efetue login em seu *hub*-cluster com `cloudctl login`. Execute o comando a seguir:

```
cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
```

3. Execute o comando a seguir para importar no managed-cluster de destino:

```
cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
```

4. Verifique se o cluster foi importado com sucesso.
    - Efetue login em seu hub-cluster do IBM Multicloud Manager.
    - Na barra de navegação, clique em **Clusters**.
    - Localize o novo managed-cluster importado na lista.
    - Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Importando um cluster do Google Kubernetes Engine

Siga o procedimento para importar um cluster do Google Kubernetes Engine (Kubernetes Engine). Para obter mais informações sobre o Kubernetes Engine, consulte Mecanismo do Google Kubernetes .

Pré-requisito

- Deve-se ter um hub-cluster do IBM Multicloud Manager, que é um cluster do IBM Cloud Private com `multi-cluster hub` ativado e com `single_cluster_mode` configurado como `false`.
- Deve-se ter uma conta do [Google Cloud Platform]
  Abre em uma novaguia](../../images/icons/launch-glyph.svg "Abre em uma nova guia").
- Deve-se ter a função *Administrador do Kubernetes Engine* para o cluster do Google Kubernetes Engine que está sendo importado.
- É necessário instalar a CLI e autenticar-se. Consulte [CLI do Google Cloud SDK]
  Abre em uma novaguia](../../images/icons/launch-glyph.svg "Abre em uma nova guia") para obter instruções para instalar a CLI.

Para obter mais informações sobre o Kubernetes Engine, consulte Mecanismo do Google Kubernetes .

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login no Google Cloud Platform com o comando a seguir:

   ```
   gcloud auth login <account>
   ```

3. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

   ```
   export KUBECONFIG=kubeconfig
   ```

4. Recupere as credenciais do cluster e armazene no diretório `./kubeconfig`. Execute o comando a seguir:

   ```
   gcloud container clusters get-credentials <cluster-name> --zone <zone> --project <project>
   ```

5. Liste os arquivos em seu diretório e confirme se o arquivo `kubeconfig` foi criado.
6. Verifique o conteúdo do arquivo `kubeconfig`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

   ```
   apiVersion: v1
   clusters:
   - cluster:
      certificate-authority-data: <ca-data>
      server: <targeted-managed-cluster-kubernete-api-server>
    name: <cluster_name>
   ```

```
contexts:
- context:
   cluster: <cluster_name>
   user: <username>
 name: <cluster_name>-context
current-context: <cluster_name>-context
kind: Config
preferences: {}
users:
- name: <username>
 user:
  auth-provider:
   config:
     cmd-args: config config-helper --format=json
     cmd-path: <gcloud_install_path>
     expiry-key: '{.credential.token_expiry}'
     token-key: '{.credential.access_token}'
   name: gcp
```

7. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.

8. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

   ```
   unset KUBECONFIG
   ```

Crie seu arquivo de configuração `cluster-import.yaml` do

IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.

2. Efetue login em seu *hub*-cluster com `cloudctl login`. Execute o comando a seguir:

   ```
   cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation-->
   ```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub:

   ```
   cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
   ```

4. Abra o arquivo `cluster-import.yaml` e configure os parâmetros a seguir:

   - `default_admin_user`: o nome do usuário administrador de cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

   Consulte a seção *Configurações do multicluster-endpoint* de Customizando o cluster com o arquivo config.yaml para obter mais parâmetros.

   Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.

2. Efetue login em seu *hub*-cluster com `cloudctl login`.

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

3. Execute o comando a seguir para importar no managed-cluster de destino:

   ```
   cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
   ```

4. Verifique se o cluster foi importado com sucesso.

   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.

- Localize o novo managed-cluster importado na lista.
- Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Importando um cluster do OpenShift

Siga o procedimento para importar um cluster do OpenShift. Para obter mais informações sobre o IBM Cloud Private with OpenShift e sobre como configurar seu ambiente, consulte Visão geral do IBM Cloud Private with OpenShift.

Pré-requisito

- Deve-se ter acesso a um ambiente com o Red Hat OpenShift.
- Você deve instalar o Docker. Para instalar o Docker, consulte Instalar o Docker.
- É necessário instalar a ferramenta de linha de comandos do Kubernetes, `kubectl`. Para instalar `kubectl`, consulte Instalar e configurar o kubectl.
- Deve-se instalar a CLI do OpenShift, `oc`. Para obter mais informações, consulte Introdução à CLI.

Prepare-se para importação

É necessário criar um diretório `<import_config_directory>` para armazenar seus arquivos de configuração de importação.

Gerando seu arquivo `kubeconfig` para o managed-cluster de destino

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir:

   ```
   export KUBECONFIG=kubeconfig
   ```

3. Efetue login no Red Hat OpenShift com `oc login`.

   ```
   oc login <OpenShift console URL>
   ```

4. Efetue login e gere seu arquivo `kubeconfig`:

   ```
   oc login <openshift console> -u <username> -p <password>
   ```

5. Liste os arquivos em seu diretório e confirme se o arquivo `kubeconfig` foi criado.
6. Verifique o conteúdo do arquivo `kubeconfig`. Certifique-se de que haja um valor para `clusters`, um valor para `contexts` e um valor para `users`. Consulte a seguinte saída de exemplo:

   ```
   apiVersion: v1
   clusters:
   - cluster:
      insecure-skip-tls-verify: true
      server: <target managed cluster OpenShift server URL>
    name: <cluster name>
   contexts:
   - context:
      cluster: <cluster name>
      namespace: default
      user: <user profile name>
    name: <context name>
   current-context: <context name>
   kind: Config
   preferences: {}
   users:
   - name: <user profile name>
    user:
      token: <user token>
   ```

7. Verifique se é possível se conectar ao seu managed-cluster de destino com `kubectl` usando o `kubeconfig`. Execute qualquer comando `kubectl`. Se você receber um erro `unable to connect`, exclua o arquivo e tente novamente.

8. Desconfigure a variável de ambiente `KUBECONFIG` para evitar modificação do arquivo `kubeconfig` à medida que você continua. Execute o comando a seguir:

```
unset KUBECONFIG
```

Crie seu arquivo de configuração `cluster-import.yaml` do

IBM Multicloud Manager

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`.

```
cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port>
--skip-ssl-validation-->
```

3. Execute o comando a seguir para criar o modelo de configuração, `cluster-import.yaml`, em que `<cluster_name>` é o nome do recurso de cluster no hub e `<cluster_namespace>` é o namespace dos recursos de cluster no hub:

```
cloudctl mc cluster template <cluster_name> -n <cluster_namespace> > cluster-import.yaml
```

4. Abra o arquivo `cluster-import.yaml` e configure os parâmetros a seguir:

   - `default_admin_user`: o nome do usuário administrador de cluster para o managed-cluster de destino
   - `container_runtime`: o tempo de execução do contêiner usado no cluster; as opções suportadas atualmente são `docker` e `containerd`

   Agora você está pronto para importar um cluster.

Importando o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`.
2. Efetue login em seu *hub*-cluster com `cloudctl login`.

```
cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
```

3. Execute o comando a seguir para importar no managed-cluster de destino:

```
cloudctl mc cluster import -f <cluster-import.yaml> --cluster-kubeconfig kubeconfig [-C|--
cluster-context {context}]  [-b|--bootstrap-namespace {namespace}] [-t|--timeout {time}]
```

4. Verifique se o cluster foi importado com sucesso.

   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.
   - Localize o novo managed-cluster importado na lista.
   - Assegure-se de que o status seja *Pronto*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

Pós-importação

Se você concluiu a etapa 4 opcional na seção *Criar seu arquivo de configuração `cluster-import.yaml` do IBM Multicloud Manager* para configurar seu Docker privado em seu `cluster-import.yaml`, será necessário remover as credenciais do ConfigMap no hub-cluster.

1. Efetue login em seu *hub*-cluster com `cloudctl login`. Execute o comando a seguir:

```
cloudctl login -a https://<Hub Cluster Master Host>:<Cluster Master API Port> --skip-ssl-
validation-->
```

2. Edite o ConfigMap `bootstrap-config` para o managed-cluster de destino. Execute o comando a seguir:

```
kubectl edit configmap -n <cluster_namespace> <cluster_name>-bootstrap-config
```

3. Remova as informações de autenticação privadas do Registro do Docker, conforme exibido no exemplo a seguir:

```
docker_username: <docker_username>
docker_password: <docker_password>
```

4. Salve e saia.

Remover um managed-cluster importado

É possível remover seu cluster gerenciado do IBM Multicloud Manager de seus diferentes provedores em nuvem do Kubernetes, incluindo o IBM Cloud Private.

Remover o cluster

1. Em um terminal, acesse o diretório `<import_config_directory>`, que você criou durante o processo de importação. Se você não tiver o diretório `<import_config_directory>`, repita as etapas *Preparar para importação* em seu procedimento de importação.

2. Efetue login em seu *hub*-cluster com `cloudctl login`.

   ```
   cloudctl login -a https://<Cluster Master Host>:<Cluster Master API Port> --skip-ssl-validation
   ```

3. Opcional: edite o ConfigMap `bootstrap-config` para o managed-cluster de destino e configure a seção a seguir se estiver direcionando para um cluster que não tenha acesso ao DockerHub público. Se você tiver acesso, ignore esta etapa.

   Para o IBM Cloud Private EE, é possível usar o registro do Docker privado do IBM Cloud Private.

   ```
   inception_image: <cluster_CA_domain>:8500/ibmcom/icp-inception:3.2.0-ee
   image_repo: <cluster_CA_domain>:8500/ibmcom
   private_registry_enabled: true
   docker_username: <username>
   docker_password: <password>
   ```

4. Execute o comando a seguir para remover o managed-cluster de destino:

   ```
   cloudctl mc cluster remove cloudctl mc cluster remove {name} [-n|--namespace {namespace}] [-C|-
   -cluster-context {context}] [-K|--cluster-kubeconfig {path}] [-b|--bootstrap-namespace
   {namespace}]
   ```

5. Verifique se seu cluster está *Off-line*.

   - Efetue login em seu hub-cluster do IBM Multicloud Manager.
   - Na barra de navegação, clique em **Clusters**.
   - Localize seu managed-cluster na lista de clusters.
   - Assegure-se de que o status esteja *Off-line*. Dependendo do ambiente, pode levar alguns minutos para que o status seja exibido.

6. Remova o recurso de cluster com o comando a seguir:

   ```
   kubectl delete cluster {cluster-name} -n {cluster-namespace}
   ```

Definindo configurações de failover para seus clusters do IBM Multicloud Manager

Nota: o failover é suportado para o Linux® x86_64. O failover não é suportado para o Linux on Power (ppc64le).

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Depois de preparar o Minio, é possível configurar o failover para o IBM Multicloud Manager para seus hub-clusters e managed-clusters.

- Preparar-se para failover do IBM Multicloud Manager
- Configurando failover do IBM Multicloud Manager para clusters do hub
- Configurando failover do IBM Multicloud Manager para managed-clusters
- Resolvendo problemas de configurações de failover para seus clusters do IBM Multicloud Manager

Preparar configurações de failover para seus clusters do

IBM Multicloud Manager

É necessário preparar o Minio para configurar o failover para o IBM Multicloud Manager. O Minio é um servidor de armazenamento de objeto distribuído de alto desempenho, que é projetado para infraestrutura de nuvem privada em larga escala. Consulte Minio para IBM Cloud Private para obter mais informações.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Prepare o Minio no principal e secundário para armazenar objetos

- Crie seu segredo de acesso do Minio
- Implemente o Minio
- Crie segredo do armazenamento de objetos para o Velero

Crie seu segredo de acesso do Minio

É necessário gerar o segredo de acesso usando suas próprias credenciais. Para o Minio, o hub-cluster *principal* e o hub-cluster *secundário* precisam compartilhar a mesma `accesskey` e `secretkey`.

1. Execute `echo -n "<credential>" |base64` para criptografar as credenciais `accesskey` e `secretkey`. Consulte a amostra de `mcm-minio-secret.yaml` a seguir com os valores:

```
apiVersion: v1
kind: Secret
metadata:
 name: mcm-minio-secret
 namespace: kube-system
type: Opaque
data:
 accesskey: <base64-encoded-access-key>
 secretkey: <base64-encoded-secret-key>
```

2. Execute o comando a seguir para criar o segredo de acesso do Minio no namespace `kube-system`. Seu nome secreto é usado para implementar o Minio no `kube-system`:

```
kubectl create -f mcm-minio-secret.yaml
```

Implemente o Minio

Você está pronto para implementar o Minio com seu nome secreto que você criou na etapa anterior.

1. Navegue para a console do IBM Cloud Private e clique em **Catálogo**.
2. Procure por `minio` e implemente `ibm-minio-objectstore` em *ibm-charts* no *Namespace de destino* e `kube-system` usando o segredo que você criou.

Nota: o nome da região é usado para definir configurações de failover para seu hub-cluster do IBM Multicloud Manager. O nome da região padrão para o IBM Cloud Private Minio é `us-east-1`.

1. Na página de serviços, procure por `minio` para editar o serviço Minio para mudar o tipo de serviço de `ClusterIP` para `NodePort`. É possível acessar o link da página de serviço com sua URL do IBM Cloud Private:

```
https://<Cluster Master Host>:<Cluster Master API Port>/console/access/services
```

2. Clique para *Ativar* o serviço Minio.
3. Efetue login no Minio com suas credenciais.
4. Clique no ícone de inclusão (+) na interface e, em seguida, clique em `Criar depósito` para criar seu depósito para armazenar os dados de backup.
5. Nomeie o depósito como `mcm`.

Crie o segredo de armazenamento de objetos para o Velero

Execute o comando a seguir para criar um segredo do Velero para conectar-se ao Minio:

```
kubectl create -f mcm-minio-cloud-credential.yaml
```

O `aws_access_key_id` e a `aws_secret_access_key` são os mesmos que as credenciais do Minio. O nome secreto é usado para definir configurações de failover para o hub-cluster do IBM Multicloud Manager.

Consulte a amostra a seguir do arquivo `mcm-minio-cloud-credential.yaml`:

```
apiVersion: v1
kind: Secret
metadata:
  namespace: kube-system
  name: cloud-credentials
  labels:
    component: minio
stringData:
  cloud: |
    [default]
    aws_access_key_id = admin
    aws_secret_access_key = admin1234
```

Agora é possível ir para Configurando o failover do IBM Multicloud Manager para hub-clusters e Configurando o failover do IBM Multicloud Manager para managed-clusters.

Defina as configurações de failover para hub-clusters do IBM Multicloud Manager

Depois de preparar o Minio, é possível definir as configurações de failover para o hub-cluster IBM Multicloud Manager*principal* e *secundário*.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Pré-requisito

É necessário que dois hub-clusters estejam preparados para failover do IBM Multicloud Manager. Consulte Preparar-se para failover do IBM Multicloud Manager.

Prepare o segredo do `kubeconfig` persistente para failover do hub-cluster

É necessário nomear seus dois clusters: nomeie um cluster *principal* como cluster e nomeie o outro cluster *secundário* como cluster.

Configure seu principal

1. Efetue login em seu cluster `secundário` e execute o comando a seguir para gerar o arquivo `~/.kube/kubeconfig`:

   ```
   export MASTER_HOST=<Cluster Master Host>
   export ACCESS_TOKEN=`kubectl get secret $(kubectl get sa default -n kube-system -o jsonpath="
   {.secrets[0].name}") -n kube-system -o jsonpath="{.data.token}" | base64 -d`
   export KUBECONFIG=~/.kube/kubeconfig
   kubectl config set-cluster mycluster --server=https://$MASTER_HOST:8001 --insecure-skip-tls-
   verify=true
   kubectl config set-context mycluster-context --cluster=mycluster
   kubectl config set-credentials mcm --token=$ACCESS_TOKEN
   kubectl config set-context mycluster-context --user=mcm --namespace=default
   kubectl config use-context mycluster-context
   unset KUBECONFIG
   ```

2. No cluster *secundário*, copie o arquivo `~/.kube/kubeconfig` do *secundário* para `~/.kube/secondary-kube-config` do cluster *principal*. Execute o comando a seguir:

   ```
   sudo scp ~/.kube/kubeconfig root@_primary_ip:~/.kube/secondary-kube-config
   ```

1. Em seu cluster *principal*, execute o comando a seguir para criar o segredo `mcm-secondary-kubeconf-secret`:

```
cd ~/.kube/
kubectl create secret generic mcm-secondary-kubeconf-secret --from-file=kubeconfig=./secondary-
kube-config -n kube-system
```

O segredo `mcm-secondary-kubeconf-secret` é usado para definir as configurações de failover.

Configure seu cluster secundário

1. Efetue login em seu cluster `primary` e execute o comando a seguir para gerar o arquivo `~/.kube/kubeconfig`:

```
export MASTER_HOST=<Cluster Master Host>
export ACCESS_TOKEN=`kubectl get secret $(kubectl get sa default -n kube-system -o jsonpath="
{.secrets[0].name}") -n kube-system -o jsonpath="{.data.token}" | base64 -d`
export KUBECONFIG=~/.kube/kubeconfig
kubectl config set-cluster mycluster --server=https://$MASTER_HOST:8001 --insecure-skip-tls-
verify=true
kubectl config set-context mycluster-context --cluster=mycluster
kubectl config set-credentials mcm --token=$ACCESS_TOKEN
kubectl config set-context mycluster-context --user=mcm --namespace=default
kubectl config use-context mycluster-context
unset KUBECONFIG
```

2. No cluster *principal*, copie o arquivo `~/.kube/kubeconfig` do *principal* para `~/.kube/primary-kube-config` do cluster *secundário*. Execute o comando a seguir:

```
sudo scp ~/.kube/kubeconfig root@_secondary_ip:~/.kube/primary-kube-config
```

3. No cluster *secundário*, execute os comandos a seguir para criar o segredo `mcm-primary-kubeconf-secret`:

```
cd ~/.kube/
kubectl create secret generic mcm-primary-kubeconf-secret --from-file=kubeconfig=./primary-
kube-config -n kube-system
```

O segredo `mcm-primary-kubeconf-secret` é usado para definir as configurações de failover.

Prepare IBM Multicloud Manager em ambos os hub-clusters para failover

1. Obtenha o nome da liberação do Helm instalada usando o seguinte comando:

```
helm list --tls |grep multicluster
multicluster-hub                    2           Wed May 22 21:37:27 2019    DEPLOYED    ibm-mcm-
prod-3.2.0                  1.0           kube-system
```

2. Crie um arquivo de substituição chamado `override.yaml`. Consulte a amostra YAML de substituição de cluster *principal*, em que `hub0` é o nome principal e `hub1` é o `peerHub`, que será o nome do cluster *secundário*.

Nota: certifique-se de que o valor de seu `initWeight` de seu cluster principal exceda o valor de seu cluster secundário.

```
core:
  myName: hub0
  highAvailability:
    enabled: true
    initWeight: 1
    peerHub: hub1
    peerHubSecret: mcm-secondary-kubeconf-secret
    backup:
      period: 5m
      ttl: 1h
      region: us-east-1
      url: <Secondary Hub Minio URL>
    restore:
      period: 5m
      region: us-east-1
      url: <Primary Hub Minio URL >
velero:
  cloudCredential: cloud-credentials
```

Consulte a amostra YAML de substituição de cluster *secundário*, em que `hub1` é o nome secundário e `hub0` é o `peerHub`, que será o nome do cluster *principal*.

```
core:
  myName: hub1
  highAvailability:
```

```
    enabled: true
    initWeight: 0
    peerHub: hub0
    peerHubSecret: mcm-primary-kubeconf-secret
    backup:
      period: 5m
      ttl: 1h
      region: us-east-1
      url: <Secondary Hub Minio URL>
    restore:
      period: 5m
      region: us-east-1
      url: <Primary Hub Minio URL >
velero:
  cloudCredential: cloud-credentials
```

Configure o failover para o hub-cluster do IBM Multicloud Manager

1. Em um terminal, execute o comando a seguir para definir as configurações de failover para o cluster principal e para o cluster secundário:

   ```
   helm upgrade  multicluster-hub --version=3.2.0 mgmt-charts/ibm-mcm-prod -f override.yaml --tls
   ```

2. Reinicie o pod do operador com o comando a seguir:

   ```
   kubectl get pod -n kube-system |grep core-operator
   multicluster-hub-core-operator-778567449c-bphd9                1/1     Running     0
   28h
   kubectl delete pod multicluster-hub-core-operator-778567449c-bphd9 -n kube-system
   ```

Continue com Configurando o failover do IBM Multicloud Manager para managed-clusters.

Definindo configurações de failover para seus managed-clusters do

IBM Multicloud Manager

É possível definir as configurações de failover para o managed-cluster do IBM Multicloud Manager.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

- Pré-requisitos
- Prepare o segredo do `kubeconfig` persistente para failover do managed-cluster
- Configure o failover para o managed-cluster do IBM Multicloud Manager

Pré-requisito

É necessário que dois managed-clusters estejam preparados para failover do IBM Multicloud Manager. Consulte Preparar-se para failover do IBM Multicloud Manager para obter mais informações.

Prepare o segredo do `kubeconfig` persistente para failover do managed-cluster

Dois clusters do IBM Cloud Private são necessários, um denominado o cluster *principal* e outro denominado o cluster *secundário*.

1. Copie o arquivo `~/.kube/kubeconfig` do *principal* para o diretório `~/.kube/` do managed-cluster. Renomeie o arquivo, `primary-kube-config`.

2. Copie o arquivo `~/.kube/kubeconfig` *secondary* para o diretório `~/.kube/` do managed-cluster. Renomeie o arquivo, `secondary-kube-config`.

3. Execute os comandos a seguir para criar os segredos `mcm-secondary-kubeconf-secret` e `mcm-primary-kubeconf-secret`:

   ```
   cd ~/.kube/
   kubectl create secret generic mcm-secondary-kubeconf-secret --from-file=kubeconfig=./secondary-
   kube-config -n kube-system
   kubectl create secret generic mcm-primary-kubeconf-secret --from-file=kubeconfig=./primary-
   kube-config -n kube-system
   ```

Configure o failover para o managed-cluster do IBM Multicloud Manager

1. Crie um arquivo de substituição chamado `override.yaml`. Consulte a amostra a seguir, em que `hub0` é definido com o segredo primário, `mcm-primary-kubeconf-secret` e `hub1` é definido com `mcm-secondary-kubeconf-secret`.

```
global:
 clusterName: c0
 clusterNamespace: cn0
operator:
 bootstrapConfig:
   hub0:
     name: hub0
     secret: mcm-primary-kubeconf-secret
   hub1:
     name: hub1
     secret: mcm-secondary-kubeconf-secret
tillerIntegration:
 user: admin
```

2. Em um terminal, execute o comando a seguir para definir as configurações de failover para o managed-cluster:

```
helm install -n  mcm-klusterlet  mgmt-charts/ibm-klusterlet -f klu.yaml --tls --namespace kube-system
```

3. Para verificar, efetue logon no hub-cluster *principal* e execute o comando a seguir. Veja se o status de saída de amostra do managed-cluster é `Ready`:

```
kubectl get cluster --all-namespaces
```

   Consulte a saída de amostra:

```
NAMESPACE   NAME      ENDPOINTS          STATUS   AGE
cn0          c0      9.46.72.117:8001   Ready    32m
```

4. Para verificar, efetue logon no hub-cluster *secundário* e execute o comando a seguir. Veja se o status de saída de amostra do managed-cluster é `Pending`:

```
kubectl get cluster --all-namespaces
```

   Consulte a saída de amostra:

```
NAMESPACE   NAME      ENDPOINTS          STATUS    AGE
cn0         c0        9.46.72.117:8001   Pending   3m
```

Resolução de problemas de configuração de failover para hub-clusters do

IBM Multicloud Manager

Revise o procedimento de resolução de problemas a seguir para os hub-clusters do IBM Multicloud Manager se você tiver problemas com a configuração de failover.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

1. No hub-cluster *principal* e no hub-cluster *secundário*, execute o comando a seguir para verificar o servidor de API e assegure-se de que não haja erros:

```
 kubectl get pods -n kube-system |grep core-apiserver
```

   A saída mostra os detalhes do servidor de API. Por exemplo:

```
 multicluster-hub-core-apiserver-f8d877c45-xdtql                  1/1     Running            0
50s
```

   Execute o comando a seguir com o nome do servidor de API:

```
 kubectl logs multicluster-hub-core-apiserver-f8d877c45-xdtql -n kube-system
```

2. No hub-cluster *principal* e no hub-cluster *secundário*, execute o comando a seguir para obter o log do operador principal. Se você receber um erro `Unable to get leadervote`, talvez não há conexão. Verifique o valor `peerHubSecret` que você definiu quando configurou o failover.

```
kubectl get pods -n kube-system |grep core-operator
```

A saída mostra os nomes de log do operador principal. Por exemplo:

```
multicluster-hub-core-operator-778567449c-cplv6                  1/1     Running            0
25h
```

Execute o comando a seguir com o nome do log do operador principal:

```
kubectl logs multicluster-hub-core-operator-778567449c-cplv6 -n kube-system
```

3. Assegure-se de que o pod Velero esteja executando sem erros no log. Se você receber um erro de armazenamento de backup, será necessário mudar o objeto `BackupStorageLocation`.

    1. Verifique o log do pod do Velero executando o comando a seguir:

    ```
    kubectl get pods -n kube-system | grep velero
    ```

    A saída exibe os nomes de log do pod do Velero, por exemplo:

    ```
    multicluster-hub-velero-5449cb49d7-hkpbw                          1/1     Running
    0        25h
    ```

    Execute o comando a seguir com o nome do log do pod do Velero, por exemplo:

    ```
    kubectl logs multicluster-hub-velero-5449cb49d7-hkpbw -n kube-system
    ```

    2. Edite o objeto `BackupStorageLocation` executando o comando a seguir para obter os nomes de backup:

    ```
    kubectl get backupstoragelocation -n kube-system
    ```

    A saída exibe os nomes de backup. Por exemplo:

    ```
    NAME                            AGE
    multicluster-hub-core-backup    4d
    multicluster-hub-core-restore   4d
    ```

    Execute o comando a seguir com o nome do backup:

    ```
    kubectl edit backupstoragelocation multicluster-hub-core-backup -n kube-system
    ```

4. Se houver um erro relacionado ao armazenamento de backup, deverá mudar o objeto `BackupStorageLocation`. Execute o comando a seguir:

    ```
    kubectl edit BackupStorageLocation –n kube-system
    ```

5. Verifique o `leadervote`. No hub-cluster *principal* e no hub-cluster *secundário*, execute o comando a seguir:

    ```
    kubectl get leadervote
    ```

    Saída de exemplo no hub-cluster *principal*:

    ```
    NAME   IDENTITY   VOTE   CURRENT LEADER   READY   AGE
    hub0   hub0       1                       true    1h
    ```

    Saída de exemplo no hub-cluster *secundário*:

    ```
    NAME   IDENTITY   VOTE   CURRENT LEADER   READY   AGE
    hub1   hub1       0                       false   1h
    ```

    O valor de `Vote` no hub-cluster *principal* deve ser igual ao valor do cluster de backup mais um.

6. Assegure-se de que o objeto de backups possa ser criado periodicamente em seu cluster principal. Execute o comando a seguir:

    ```
    Kubectl get backups –n kube-system –o yaml
    ```

    Você deve receber um status de `completed` ou `InProgress`.

7. Se você não receber o status `completed` ou `InProgress`, execute o comando a seguir para obter detalhes do planejamento:

    ```
    kubectl get schedule –n kube-system –oyaml
    ```

8. Se o planejamento for nulo, use o comando a seguir para reiniciar o pod do operador:

```
 kubectl get pod -n kube-system |grep core-operator
 multicluster-hub-core-operator-778567449c-bphd9                1/1     Running     0
28h

 kubectl delete pod multicluster-hub-core-operator-778567449c-bphd9 -n kube-system
```

9. Assegure-se de que o objeto de restauração possa ser criado periodicamente em seu cluster de backup executando o comando a seguir:

```
kubectl get restore -n kube-system -o yaml
```

   Você deve receber um status de `completed` ou `InProgress`.

Atualizando IBM Multicloud Manager

É possível fazer upgrade do IBM Multicloud Manager no IBM Cloud Private depois de fazer upgrade do IBM Cloud Private. Depois de fazer upgrade do IBM Cloud Private, é possível fazer upgrade de seu hub-cluster e de seu managed-cluster.

Nota: alguns recursos da console de gerenciamento do IBM Cloud Private variam dependendo da versão do hub-cluster e da versão de seu managed-cluster.

- Fazendo upgrade de seu hub-cluster do IBM Multicloud Manager
- Fazendo upgrade de seu managed-cluster do IBM Multicloud Manager

Fazendo upgrade de seu hub-cluster do IBM Multicloud Manager

É possível fazer upgrade de seu hub-cluster do IBM Multicloud Manager no IBM® Cloud Private, se o IBM Cloud Private for atual.

Consulte Fazer upgrade do cluster do IBM Cloud Private para fazer upgrade do cluster do IBM Cloud Private.

Importante: É necessário o armazenamento persistente ativado na instalação atual do IBM Multicloud Manager para evitar a perda de dados.

Caminhos de Upgrade Suportados

É possível fazer upgrade apenas dos caminhos suportados a seguir:

- IBM Cloud Private versão 3.1.2 para 3.2.0
- IBM Multicloud Manager versão 3.1.2 para 3.2.0

Fazendo upgrade de seu hub-cluster

1. Antes de fazer upgrade de seu hub-cluster, é necessário fazer upgrade do IBM Cloud Private.

   Importante: em seu arquivo `/<upgrade_directory>/cluster/config.yaml` que você editou durante a configuração, desative o `multicluster-hub`, conforme o exemplo a seguir:

```
management_services:
  ...
  multicluster-hub: disabled
  ...
```

2. Depois de fazer upgrade do IBM Cloud Private, navegue para a console de gerenciamento do IBM Cloud Private com a URL a seguir. Visualize a tabela **Liberações do Helm**.

```
https://<Cluster Master Host>:<Cluster Master API Port>/catalog/instances
```

3. Localize e clique na liberação para o gráfico `ibm-mcm-prod` para abrir a página de detalhes.

4. Clique em **Fazer upgrade** para abrir a janela de upgrade.

5. Selecione `mgmt-charts` como o repositório de gráfico e selecione `3.2.0` como a versão de destino.

6. Configure os parâmetros a seguir em `Quick start`: Configurações globais do Multicloud Manager

- Namespace do Multicloud Manager: o namespace que você configurou durante a instalação do IBM Multicloud Manager
- Nome do usuário do Tiller: o nome do usuário usado para se comunicar com o serviço tiller-deploy para implementar gráficos do Helm

7. Clique em **Fazer upgrade** para fazer upgrade do IBM Multicloud Manager versão 3.1.2 para 3.2.0.

Verificação

1. Execute o comando a seguir para verificar o status para assegurar que os pods exibem o status `Running`:

   ```
   kubectl -n kube-system get pods | grep <helm_release_name>
   ```

   Consulte a saída de amostra a seguir:

   ```
   mcm-console-mcmui-7564ffdd66-kdz8l                                1/1     Running     0
   20m
   mcm-console-mcmuiapi-947ff4b79-lm8kt                              1/1     Running     0
   20m
   mcm-core-apiserver-7cdb75d4fd-q69ls                               1/1     Running     0
   ```

2. Execute o comando a seguir para assegurar que a liberação do Helm foi submetida a upgrade para 3.2.0.

   ```
   helm list --tls | grep ibm-mcm-prod
   ```

   Consulte a seguinte saída de exemplo:

   ```
   mcm    2    Wed May 15 01:39:18 2019    DEPLOYED    ibm-mcm-prod-3.2.0    1.0    kube-system
   ```

3. Se você tiver um managed-cluster anterior, execute o comando a seguir para assegurar-se que seu cluster esteja no status `Ready`:

   ```
   kubectl get cluster --all-namespaces
   ```

   Consulte a seguinte saída de exemplo:

   ```
   NAMESPACE   NAME        ENDPOINTS          STATUS   AGE
   mycn        mycluster   9.30.0.174:8001    Ready    3h
   ```

   Agora é possível fazer upgrade de seu managed-cluster.

   Consulte Fazendo upgrade de seu managed-cluster do IBM Multicloud Manager antes de fazer upgrade do Klusterlet.

Fazendo upgrade de seu managed-cluster do IBM Multicloud Manager

Será possível fazer upgrade do managed-cluster se o upgrade do hub-cluster do IBM Multicloud Manager já foi feito.

É necessária uma versão atual do cluster do hub. Consulte Fazendo upgrade de seu hub-cluster do IBM Multicloud Manager antes de fazer upgrade de seu managed-cluster.

Caminhos de Upgrade Suportados

É possível fazer upgrade apenas dos caminhos suportados a seguir:

Fazendo upgrade de seu managed-cluster

1. Configure um managed-cluster com o comando `import`. Consulte Importando um managed-cluster de destino no hub-cluster do IBM Multicloud Manager para obter instruções.

2. Acesse o diretório `<import_config_directory>`.

3. Configure a variável de ambiente `KUBECONFIG` com o comando a seguir, que foi configurado durante o processo de importação:

   ```
   export KUBECONFIG=<kubeconfig-file>
   ```

4. Execute o comando a seguir para assegurar que os pods exibam o status `Running`:

```
kubectl get pods -n multicluster-endpoint
```

Consulte a seguinte saída de exemplo:

```
NAME                                                              READY   STATUS    RESTARTS
AGE
multicluster-endpoint-ibm-klusterlet-klusterlet-7dc57b447c9thg8   2/2     Running   0
19d
```

5. Execute o comando a seguir para obter o nome da liberação e o namespace de liberação do Helm a partir do managed-cluster anterior:

```
helm list --tls | grep ibm-mcmk-prod-3.1.2
```

Consulte a seguinte saída de exemplo:

```
NAME                REVISION      UPDATED             STATUS      CHART
NAMESPACE
<release-name>      1             <date revised>      DEPLOYED    ibm-mcmk-prod-3.1.2
<release-namespace>
```

6. Execute o comando a seguir para editar a implementação:

```
kubectl edit deployment <release-name>-ibm-mcmk-prod-klusterlet -n <release namespace>
```

7. Edite o `spec:replicas`. Mude para o valor diminuído de `0`.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
... ...
  name: <release-name>-ibm-mcmk-prod-klusterlet
... ...
spec:
  ... ...
  replicas: 0
  ... ...
```

8. Acesse a console de gerenciamento do hub-cluster do IBM Cloud Private com sua URL e efetue login com suas credenciais.

9. Na navegação, clique em **Clusters** para visualizar a tabela de cluster. Localize o `<cluster-name>` do managed-cluster e verifique o status `Ready` e a mudança de versão.

10. Volte para o seu terminal. Execute o comando a seguir para excluir a versão 3.1.2:

```
helm delete <release-name> --tls --purge --no-hooks
```

Agora você está pronto para verificar seu upgrade.

Verificação

1. Execute o comando a seguir para verificar o pod para o status `Running`:

```
kubectl get pods -n multicluster-endpoint
```

2. Atualize a console de gerenciamento do hub-cluster e certifique-se de que seu cluster esteja `Ready`.

Trabalhando com aplicativos do IBM Multicloud Manager

Saiba como usar os aplicativos IBM Multicloud Manager .

- IBM Multicloud Manager Visão Geral de Aplicativos
- Criando IBM Multicloud Manager recursos do aplicativo
- Criando um PlacementPolicy para implementar recursos do aplicativo
- Excluindo recurso de aplicativo do IBM Multicloud Manager

IBM Multicloud Manager visão geral do recurso de aplicativo

Um aplicativo IBM Multicloud Manager consiste em cinco recursos de aplicativo, que são definidos como *Application*, *Deployable*, *PlacementPolicy*, *DeployableOverride* e *ApplicationRelationship*.

Especificação do aplicativo

Um aplicativo de cluster é definido com a especificação de comunidade do Kubernetes SIG Application CRD. Um aplicativo de múltiplos clusters usa a mesma especificação do Kubernetes, mas com automação adicional da implementação e do gerenciamento de ciclo de vida de recursos para clusters individuais.

Exemplos de Recurso de Aplicativo

Visualize e implemente aplicativos com os cinco recursos. Seu *Application* é usado somente para *visualizar* seu recurso, enquanto os outros quatro exemplos de recurso de aplicativo são para implementação.

Recurso de aplicativo

Visualize seu recurso com um recurso *Application*. Consulte o exemplo a seguir da definição para um *Application*.

Edite a seção `spec` de seu YAML para definir seu aplicativo com rótulos. O Application `spec` agrupa recursos em `componentKinds` com base no `seletor`. Consulte a amostra a seguir, em que Service, Deployment e Statefulset, com o rótulo `app: details`, são agrupados no `details-app` do Application:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: Application
metadata:
  name: details-app
  labels:
    app: details
spec:
  selector:
    matchLabels:
      app: details
  componentKinds:
  - group: core
    kind: Service
  - group: apps
    kind: Deployment
  - group: apps
    kind: StatefulSet
```

Implementáveis

Um recurso *Deployable* implementa seu gráfico do Helm.

Edite a seção `spec` do YAML para definir o aplicativo. Consulte a amostra a seguir de `spec` Deployable, que define o gráfico do Helm como um recurso implementável. Aqui, `chartName`, `nginx-lego` e `chartVersion`, `0.3.1`, estão no repositório `google`. O `placementPolicy` define os critérios de implementação.

```
apiVersion: mcm.ibm.com/v1alpha1
kind: Deployable
metadata:
  name: trader
  labels:
    serviceKind: ApplicationService
    name: trader
    placementPolicy: "trader"
spec:
  deployer:
    kind: helm
    helm:
      repository: google
      chartName: nginx-lego
      chartVersion: 0.3.1
```

PlacementPolicy

PlacementPolicy define os critérios para localizar um ou mais clusters para colocação ou implementação de cargas de trabalho. Por exemplo, se você criar um PlacementPolicy, incluir ou remover um cluster ou se a conformidade de seu cluster mudar, o IBM

Multicloud Manager gerará uma nova lista de clusters de destino com base nessas mudanças. Se mais de um cluster corresponder a esses critérios, o IBM Multicloud Manager usará `resourceHint` como critério e, em seguida, excluirá o que não é mais válido.

Edite a seção `spec` de seu YAML para definir seu PlacementPolicy. Consulte o seguinte PlacementPolicy de amostra que define um cluster com `purpose` como `prod` e `resourceHint` definido como `cpu` para que o IBM Multicloud Manager escolha o cluster com os núcleos de CPU mais disponíveis.

```
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementPolicy
metadata:
  name: trader
spec:
  replicas: 1
  clusterLabels:
    matchLabels:
      purpose: prod
  resourceHint:
    type: cpu
    order: desc
```

DeployableOverride

O *DeployableOverride* é usado para definir valores diferentes do Deployable original e substituir uma implementação.

Edite a seção `spec` de seu YAML para definir seu DeployableOverride. Qualquer valor no `spec template` pode ser diferente do Deployable original.

Consulte a amostra a seguir, em que `deployer` está definido para o `cluster de produção` e para o `cluster de teste`. O arquivo de amostra contém os mesmos `chartName` e `chartVersion` do Helm, mas valores de repositório diferentes para implementação em um cluster.

```
apiVersion: mcm.ibm.com/v1alpha1
kind: DeployableOverride
metadata:
  name: trader
spec:
  overrides:
  - clustername: productioncluster
    namespace: default
    template:
        deployer:
          kind: helm
          helm:
            repository: production-repo
            chartName: app-chart
            chartVersion: 0.11.0
            values: somevalues
  - clustername: testcluster
    namespace: default
    template:
        deployer:
          kind: helm
          helm:
            repository: test-repo
            chartName: app-chart
            chartVersion: 0.11.0
            values: someothervalues
```

ApplicationRelationship

ApplicationRelationship implementa um aplicativo em outro aplicativo com base nos valores de `source` e `target`.

Edite a seção `spec` do YAML para definir o aplicativo. A amostra a seguir exibe a definição para os detalhes do *ApplicationRelationship*, em que `product-app` do `Deployable` depende de `details-app` do `Deployable` e os dois são implementados juntos.

```
apiVersion: mcm.ibm.com/v1alpha1
kind: ApplicationRelationship
metadata:
    name: product-to-details
    labels:
      app: productpage
```

```
spec:
  type: usesCreated
  source:
    clustername: crucial-owl
    kind: Deployable
    name: product-app
    namespace: default
  destination:
    kind: Deployable
    name: details-app
    clustername: crucial-owl
    namespace: default
```

Consulte Criando recursos de aplicativo IBM Multicloud Manager para aprender a criar recursos de aplicativo.

Em seguida, consulte Trabalhando com aplicativos IBM Multicloud Manager para obter mais tópicos sobre o aplicativo.

Criando IBM Multicloud Manager recursos do aplicativo

É possível aplicar seu arquivo YAML para criar os cinco recursos de aplicativo, que são definidos como *Application*, *Deployable*, *PlacementPolicy*, *DeployableOverride* ou *ApplicationRelationship*.

Criar um Aplicativo

Os recursos de aplicativo são definidos por especificações de comunidade do CRD do aplicativo SIG do Kubernetes. Aprenda a criar seus recursos de aplicativo.

1. Escolha entre os dois processos a seguir para criar seu recurso de aplicativo:
2. Edite e salve seu arquivo YAML com qualquer ferramenta, em seguida, execute o seguinte para aplicá-lo a um api-server:

   ```
   kubectl apply -f <filename>
   ```

   Você precisa de um gráfico Helm para empacotar todos os arquivos YAML para seu aplicativo.

   - Substitua a URL do gráfico por um gráfico e atualize `targetCluster` com o filtro de cluster. Este exemplo pode ser usado para quaisquer aplicativos com somente um gráfico Helm.

3. Em seguida, verifique se você criou seu recurso, executando o comando a seguir:

   ```
   kubectl get <kind>
   ```

Especificações do IBM Multicloud Manager

Aprenda sobre as especificações para Deployable, DeployableOverride e ApplicationRelationship. Consulte Criando um PlacementPolicy do IBM Multicloud Manager para implementar recursos de aplicativo para aprender a configurar esse recurso.

- Criar um Implementáveis
- Criar um DeployableOverride
- Criar um recurso ApplicationRelationship
- Criar um repositório do Helm local

Criar um Implementáveis

É possível definir sua implementação com a especificação `deployer`, como o tipo `helm`, que é suportado com a versão atual. Implemente `helm` com uma das duas opções a seguir:

1. Atualize seu .yaml com `repository`, `chartName` e `version`.
2. Forneça o `chartURL` diretamente.

Também é necessário especificar valores. Escolha entre um dos procedimentos a seguir:

1. Insira bytes codificados em base64 em valores.
2. Forneça uma URL de valores para o IBM Multicloud Manager para download.

É necessário especificar o namespace do cluster remoto no qual você deseja implementar. Consulte a amostra a seguir:

```
apiVersion: mcm.ibm.com/v1alpha1
  kind: Deployable
  metadata:
    name: trader
    labels:
      serviceKind: ApplicationService
      name: trader
      placementPolicy: "trader"
  spec:
    deployer:
      kind: helm
      helm:
        repository: google
        chartName: nginx-lego
        chartVersion: 0.3.1
```

Criar um DeployableOverride

É possível implementar configurações exclusivas em clusters diferentes, como duas configurações diferentes para um cluster de produção e do desenvolvedor com `DeployableOverride`. Essa especificação contém múltiplas substituições, cada uma delas com dois membros: `clusterName` e `template`.

```
apiVersion: mcm.ibm.com/v1alpha1
kind: DeployableOverride
metadata:
  name: trader
spec:
  overrides:
  - clustername: productioncluster
    namespace: default
    template:
        deployer:
          kind: helm
          helm:
            repository: production-repo
            chartName: app-chart
            chartVersion: 0.11.0
            values: somevalues
  - clustername: testcluster
    namespace: default
    template:
        deployer:
          kind: helm
          helm:
            repository: test-repo
            chartName: app-chart
            chartVersion: 0.11.0
            values: someothervalues
```

Criar um recurso ApplicationRelationship

É possível definir relacionamentos entre componentes dentro do Applications com `ApplicationRelationship`.

Para essa especificação, o `type` é `usesCreated`. Também é possível definir `source` e `destination`, que consistem em `clustername`, `kind`, `name` e `namespace`. Consulte a amostra a seguir:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: ApplicationRelationship
metadata:
    name: product-to-details
    labels:
      app: productpage
spec:
  type: usesCreated
  source:
    clustername: crucial-owl
    kind: Deployable
    name: product-app
    namespace: default
  destination:
    kind: Deployable
    name: details-app
```

```
    clustername: crucial-owl
    namespace: default
```

Criar um repositório do Helm local

É possível incluir gráficos do Helm no repositório interno, que é fornecido pelo IBM Cloud Private. É possível criar um repositório interno a partir de `kind: HelmRepo`.

Consulte a seguinte especificação, em que `kube-system` é o `namespace` direcionado, mas o valor pode ser qualquer namespace ao qual os usuários estão designados e em que `local-charts` é o `name`, mas o valor pode ser qualquer nome de repositório. A `anotação` deve ser configurada como `true` e deve-se incluir o `spec: url`:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: HelmRepo
metadata:
  namespace: kube-system
  name: local-charts
  annotations:
      mcm.ibm.com/hub-cluster-repo: "true"
spec:
  url: https://<mycluster.icp>:8443/helm-repo/charts
```

Consulte Criar PlacementPolicy para implementar os recursos de aplicativo IBM Multicloud Manager para implementar um aplicativo PlacementPolicy.

Consulte Trabalhando com aplicativos IBM Multicloud Manager para obter mais tópicos sobre o aplicativo.

Criando um PlacementPolicy para implementar recursos do aplicativo

Edite o `PlacementPolicy` para implementar recursos de aplicativo. Execute os comandos a seguir para implementar seu aplicativo:

1. Execute `kubectl edit PlacementPolicy <name> -n <namespace>` para mudar o cluster no qual você deseja implementar o aplicativo.
2. Para verificar, execute `kubectl get placementpolicy <name> -n <namespace>`. Após a mudança ser feita em `PlacementPolicy`, o `Status.Decisions` é atualizado.
3. Execute `kubectl get work --all-namespaces` para listar todos os trabalhos que estão associados à política de localização. O trabalho é criado no namespace do cluster de destino que não está no namespace `placementpolicy`.

Criar um PlacementPolicy

É possível definir sua implementação com as especificações de `PlacementPolicy` a seguir em seu arquivo `.yaml`:

1. Atualize `replicas` com o número de clusters nos quais você deseja implementar.
2. Escolha e insira o `clusterSelector`, que é `matchLabels` e `matchExpressions`.
3. Insira o `resourceSelector`, que é `cpu` ou `memory`.
4. Insira os `compliances` para seu cluster.

Consulte o exemplo a seguir, com `replicas` mudado para `2` para implementação em 2 clusters e com `purpose` mudado de `prod` para `dev` para reimplementação em um ambiente de produção. No exemplo, o `matchExpressions` identifica o cluster com um rótulo de `tier` que não está no `cache`. Além disso, o `local` para o cluster é definido com `us` e os `compliances` do cluster são inseridos.

```
  apiVersion: mcm.ibm.com/v1alpha1
  kind: PlacementPolicy
  metadata:
    name: trader
  spec:
    replicas: 2
    clusterSelector:
      matchLabels:
        purpose: dev
        location: us
      matchExpressions:
      - {key: tier, operator: NotIn, values: [cache]}
    resourceSelector:
```

```
    type: cpu
    order: desc
  compliances:
  - hippacompliances
  - corporatecompliancs
```

Criar um PlacementBinding para seu Deployable

Consulte Criando uma conformidade do IBM Multicloud Manager para aprender como criar PlacementPolicy e ligar com `PlacementBinding`. Depois de criar uma política, é possível ligar seu Deployable. Consulte o exemplo a seguir, que liga o Deployable `watson-conversation-app` ao PlacementPolicy `watson-conversation-app`. O `apiGroup`, `Name` e `Kind` são necessários para especificar exclusivamente o recurso:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
metadata:
  labels:
    name: watson-conversation-app-binding
    placementPolicy: watson-conversation-app
  name: watson-conversation-app-binding
  namespace: default
placementRef:
  apiGroup: mcm.ibm.com
  kind: PlacementPolicy
  name: watson-conversation-app
subjects:
- apiGroup: mcm.ibm.com
  kind: Deployable
  name: watson-conversation-app
```

Para obter mais detalhes sobre políticas de conformidade, consulte Trabalhando com a conformidade do IBM Multicloud Manager.

Consulte Trabalhando com aplicativos IBM Multicloud Manager para obter mais tópicos sobre o aplicativo.

Excluindo recursos de aplicativo do IBM Multicloud Manager

1. É possível excluir aplicativos usando um dos procedimentos a seguir:
   - Se você criou o recurso com `kubectl`, execute o comando a seguir para excluir todos os recursos de aplicativo:

     ```
     kubectl delete <kind> <name> -n <namespace>
     ```

     Após o *Deployable* ou *PlacementPolicy* ser excluído, a carga de trabalho é removida dos clusters de destino.
   - Se você criou o recurso com uma liberação do Helm, exclua essa liberação do Helm para excluir automaticamente todos os recursos e todas as cargas de trabalho.

1. Execute o comando a seguir para verificar se seu recurso foi removido:

   ```
   kubectl get <kind> <name>
   ```

Consulte Trabalhando com aplicativos IBM Multicloud Manager para obter mais tópicos sobre o aplicativo.

Trabalhando com políticas do IBM Multicloud Manager

Saiba como usar políticas do IBM Multicloud Manager

- Visão geral da política do IBM Multicloud Manager
- Exemplo de política do IBM Multicloud Manager
- Criando uma política IBM Multicloud Manager
- Gerenciando uma política de segurança
- Excluindo uma política do IBM Multicloud Manager

Visão geral de política do IBM Multicloud Manager

Um modelo do IBM Multicloud Manager é definido em um documento sobre políticas. Cada documento sobre políticas pode ter pelo menos um ou vários modelos.

Elementos de política

Cada *política* dentro da conformidade contém os elementos a seguir:

- Um seletor de `namespace` que especifica a quais namespaces dentro do cluster a política se aplica.
- Uma lista de `modelos`, como `modelos de função`, `modelos de objetos` e `modelos de política` dentro da política que descrevem como um recurso no Kubernetes deve ser definido e se ele tem permissão para existir.
    - Um `modelo da função` é usado para listar funções RBAC que devem ser avaliadas ou aplicadas aos clusters gerenciados. Os modelos de função são tratados como uma categoria especial de modelos, já que eles possuem regras internas que podem ser analisadas e comparadas para avaliar a conformidade de um cluster.
    - Um `modelo de objeto` é usado para listar qualquer outro objeto do Kubernetes que deve ser avaliado ou aplicado aos clusters gerenciados. Um exemplo de objeto pode ser uma política de segurança de pod, uma política de imagem ou um intervalo de limite.
    - Um `modelo de política` é usado para criar uma ou mais políticas para controles de segurança de terceiros ou externos. Por exemplo, é possível criar uma política de mutação com o controlador de política de mutação. Para obter informações adicionais, consulte a Visualização de tecnologia para o *Controlador de política de mutação*.

Consulte Trabalhando com políticas do IBM Multicloud Manager para obter mais tópicos de política.

Exemplo de política do IBM Multicloud Manager

Cada política do IBM Multicloud Manager pode ter pelo menos um ou vários modelos. Para obter mais detalhes sobre os elementos de política, consulte Visão geral de política do IBM Multicloud Manager.

A política requer um *PlacementPolicy* que define os clusters aos quais o documento de política se aplica, e um *PlacementBinding* que liga a política do IBM Multicloud Manager ao `PlacementPolicy`. Visualize a política de localização e a ligação de localização de exemplo:

- Política de localização de exemplo que seleciona clusters com base em um rótulo:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementPolicy
metadata:
  name: placement1
spec:
  clusterLabels:
    matchLabels:
      cloud: IBM
```

- Ligação de localização de exemplo que liga a política de localização e o documento sobre políticas:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
metadata:
  name: binding1
  namespace: mcm
placementRef:
  name: placement1
  kind: PlacementPolicy
  apiGroup: mcm.ibm.com
subjects:
- name: policy1
  kind: Policy
  apiGroup: policy.mcm.ibm.com
```

Importante: O `PlacementPolicy` e o `PlacementBinding` podem ser incluídos no mesmo arquivo `.yaml` ou em arquivos `.yaml` separados.

Exemplo de Política

```
apiVersion: policy.mcm.ibm.com/v1alpha1
kind: Policy
metadata:
  name: policy1
  annotations:
    policy.mcm.ibm.com/standards: NIST
    policy.mcm.ibm.com/categories: SystemAndInformationIntegrity, RBAC
    policy.mcm.ibm.com/controls: MutationAdvisor
spec:
  remediationAction: "enforce" # enforce or inform
  complianceType: "musthave" # used as default, when missing in a particular sub-template
  namespaces:
    include: ["default"]
    exclude: ["kube*"]
  role-templates:
    - apiVersion: roletemplate.mcm.ibm.com/v1alpha1
      metadata:
        namespace: "" # will be inferred
        name: operator-role
      selector:
        matchLabels:
          dev: "true"
      complianceType: "musthave" # at this level, it means the role must exist with the rules that
it must have below
      rules:
        - complianceType: "mustnothave" # at this level, it means if the role exists the rule is a
mustnothave
          policyRule:
            apiGroups: ["core"]
            resources: ["secrets"]
            verbs: ["get", "list", "watch","delete", "create", "update", "patch"]
        - complianceType: "musthave" # at this level, it means if the role exists the rule is a
musthave
          policyRule:
            apiGroups: ["core"]
            resources: ["pods"]
            verbs: ["get", "list", "watch"]
    object-templates:
      - complianceType: "musthave"
        objectDefinition:
          kind: RoleBinding
          apiVersion: rbac.authorization.k8s.io/v1
          metadata:
            name: operate-pods-rolebinding
            namespace: default
          subjects:
          - kind: User
            name: admin # Name is case sensitive
            apiGroup: rbac.authorization.k8s.io
          roleRef:
            kind: Role #this must be Role or ClusterRole
            name: operator # this must match the name of the Role or ClusterRole you wish to bind to
            apiGroup: rbac.authorization.k8s.io
      - complianceType: "musthave"
        objectDefinition:
          apiVersion: policy/v1beta1
         kind: PodSecurityPolicy
         metadata:
           name: restricted-mcm
           annotations:
             seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
          spec: privileged: false # no priviliedged pods allowPrivilegeEscalation: false
allowedCapabilities: - '*'
            volumes: - '*'
            hostNetwork: true
                  hostPorts:
                  - min: 1000 # ports < 1000 are reserved
                    max: 65535
                  hostIPC: false
                  hostPID: false
                  runAsUser:
                    rule: 'RunAsAny'
                  seLinux:
                    rule: 'RunAsAny'
                  supplementalGroups:
```

```
        rule: 'RunAsAny'
      fsGroup:
        rule: 'RunAsAny'
  - complianceType: "musthave"
    objectDefinition:
      kind: NetworkPolicy
      apiVersion: networking.k8s.io/v1
      metadata:
        namespace: default
        name: deny-from-other-namespaces
      spec:
        podSelector:
          matchLabels:
        ingress:
        - from:
          - podSelector: {} # accept ingress from all pods within this namespace only
  - complianceType: "musthave"
    objectDefinition:
      apiVersion: v1
      kind: LimitRange
      metadata:
        name: mem-limit-range
      spec:
        limits:
        - default:
            memory: 512Mi
          defaultRequest:
            memory: 256Mi
          type: Container
```

Consulte Trabalhando com políticas do IBM Multicloud Manager para obter mais tópicos de conformidade.

Criando uma política do IBM Multicloud Manager

Para criar uma política para o IBM Multicloud Manager, deve-se criar um arquivo YAML para criar uma política para clusters gerenciados.

Pré-requisito

1. Deve-se instalar o IBM® Cloud Private. Para obter mais informações, visualize Instalando o IBM Cloud Private.
2. Deve-se configurar o IBM Multicloud Manager com seu cluster do IBM Cloud Private. Para obter mais informações, visualize Configurando a instalação do IBM Multicloud Manager para obter mais informações.

Nível de acesso necessário: Pelo menos um Operador.

É possível criar um arquivo YAML para sua política do IBM Multicloud Manager ou criar uma política a partir da console. Visualize as seções a seguir para criar uma política:

- Criando um arquivo YAML para uma política do IBM Multicloud Manager
- Criando uma política de segurança de cluster a partir da console do IBM Multicloud Manager

Os seguintes objetos são necessários para a política do IBM Multicloud Manager:

- *PlacementPolicy*: Define um *seletor de cluster* no qual a conformidade deve ser implementada.
- *PlacementBinding*: Liga a localização a um PlacementPolicy.

Visualize mais descrições dos arquivos YAML de política no exemplo de política do IBM Multicloud Manager.

Criando um arquivo YAML para uma política do IBM Multicloud Manager

Conclua as seguintes etapas para criar uma política:

1. Crie um arquivo YAML para sua política. Sua política pode ser semelhante ao seguinte arquivo YAML:

```
apiVersion: policy.mcm.ibm.com/v1alpha1
   kind: Policy
   metadata:
     name: policy1
   spec:
```

```
    remediationAction: "enforce" # or inform
    namespaces:
      include: ["default"]
      exclude: ["kube*"]
```

- O valor *enforce* fornece correção automática com base nas políticas.
- O valor *inform* relata se o cluster está em conformidade com as políticas especificadas.

  Nota: Por exemplo, com `remediationAction` configurado como *inform* e uma política que requer um controle de acesso baseado na função (RBAC) específico, o status de conformidade relata que o cluster estará fora de conformidade se a função não estiver no cluster. A política inclui a lista de violações. Nenhuma ação de correção é executada no modo *informativo*.

  Com `remediationAction` configurado como *enforce*, o gerenciador de conformidade do IBM Multicloud Manager cria automaticamente a política ausente nos clusters gerenciados de destino.

2. Defina o modelo que a política usa. Para criar um modelo, edite o arquivo `.yaml` incluindo um campo `modelos`. Seu `modelo` pode ser semelhante ao seguinte conteúdo:

```
 role-templates:
   - kind: RoleTemplate
     apiVersion: roletemplate.mcm.ibm.com/v1alpha1
     complianceType: "musthave" # at this level, it means the role must exist and must have the
following rules
     metadata:
       namespace: "" # will be inferred
       name: operator
     selector:
       matchLabels:
         dev: "true"
     rules:
       - complianceType: "musthave" # at this level, it means if the role exists the rule is a
musthave
         policyRule:
           apiGroups: ["extensions", "apps"]
           resources: ["deployments"]
           verbs: ["get", "list", "watch", "create", "delete","patch"]
```

3. Defina um `PlacementPolicy`. Certifique-se de mudar o `PlacementPolicy` para especificar os clusters aos quais as políticas precisam ser aplicadas, por `clusterNames` ou `clusterLabels`. Seu `PlacementPolicy` pode ser semelhante ao seguinte conteúdo:

```
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementPolicy
metadata:
  name: placement1
spec:
  clusterNames:
   - "cluster1"
   - "cluster2"
  clusterLabels:
    matchLabels:
      cloud: IBM
```

4. Defina um `PlacementBinding` para ligar sua política e seu `PlacementPolicy`. Seu `PlacementBinding` pode ser semelhante à amostra YAML a seguir:

```
---
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
metadata:
  name: binding1
placementRef:
  name: placement1
  apiGroup: mcm.ibm.com
  kind: PlacementPolicy
subjects:
- name: compliance1
  apiGroup: mcm.ibm.com
  kind: Compliance
```

Também é possível ligar seu Deployable com o PlacementBinding. Para obter mais informações, visualize *Criar um PlacementBinding para seu Deployable* em Criando um PlacementPolicy para implementar recursos do aplicativo.

5. Aplique a política executando o seguinte comando:

```
kubectl apply -f <policy-file-name>  --namespace=<mcm_namespace>
```

6. Verifique e liste as políticas executando o seguinte comando:

```
kubectl get policies --namespace=<mcm_namespace>
```

7. Visualize detalhes de uma única política executando o seguinte comando:

```
kubectl get policy <policy-name> -n <mcm_namespace> -o yaml
```

Para obter um exemplo da amostra de arquivo YAML inteira, visualize o exemplo de política.

Criando uma política de segurança de cluster a partir do IBM Multicloud Managerconsole

1. No menu de navegação, clique em **Políticas**.
2. Para criar uma política, clique em **Criar política**.

3. Visualize a definição de política de segurança de exemplo do IBM Multicloud Manager. Copie e cole o arquivo YAML para sua política.

   Seu arquivo YAML pode ser semelhante à seguinte política:

```
apiVersion: policy.mcm.ibm.com/v1alpha1
kind: Policy
metadata:
  name: policy-pod
  namespace: mcm
  annotations:
    policy.mcm.ibm.com/categories:
'SystemAndCommunicationsProtections,SystemAndInformationIntegrity'
    policy.mcm.ibm.com/controls: 'MutationAdvisor,VA'
    policy.mcm.ibm.com/standards: 'NIST,HIPAA'
spec:
  complianceType: musthave
  namespaces:
    exclude:
      - kube*
    include:
      - default
  object-templates:
    - complianceType: musthave
      objectDefinition:
        apiVersion: v1
        kind: Pod
        metadata:
          name: nginx1
        spec:
          containers:
            - name: nginx
              image: 'nginx:1.7.9'
              ports:
                - containerPort: 80
  remediationAction: enforce
```

   Importante: Certifique-se de incluir valores para o `policy.mcm.ibm.com/controls` e `policy.mcm.ibm.com/standards` para exibir cartões modais de quais controles e padrões são violados na seção *Visão geral da política*.

4. Clique em **Criar política**.

Uma política de cluster é criada.

Para gerenciar suas políticas, consulte Gerenciando uma política de segurança para obter informações adicionais. Consulte Trabalhando com políticas do IBM Multicloud Manager para obter mais tópicos de conformidade.

Gerenciando uma política de segurança

Gerencie sua política de cluster para corrigir e editar sua política de segurança.

Gerenciando uma política de segurança a partir da console de gerenciamento

1. No menu de navegação, clique em **Políticas**.
2. Na guia *Visão geral*, selecione uma violação de política para visualizar quais clusters estão fora de conformidade.
3. Clique na guia *Todas as políticas* para visualizar uma tabela de suas políticas.
4. Selecione uma política para visualizar seus detalhes.
5. Para ativar a edição de seu arquivo YAML, clique em **Editar**.
6. Depois de editar o modelo YAML, clique em **Enviar**.

Visualizar Excluindo uma política do IBM Multicloud Manager.

Excluindo uma política do IBM Multicloud Manager

Exclua sua política de segurança do IBM Multicloud Manager.

Excluindo uma política do IBM Multicloud Manager a partir da linha de comandos (CLI)

Conclua as etapas a seguir para excluir sua política:

1. Exclua uma política executando o seguinte comando:

   ```
   kubectl delete policy <policy-name> -n <mcm namespace>
   ```

   Depois que a política é excluída, ela é removida de seu cluster ou clusters de destino.

2. Verifique se sua política foi removida executando o seguinte comando:

   ```
   kubectl get policy <policy-name> -n <mcm namespace>
   ```

Excluindo uma política do IBM Multicloud Manager a partir da

console de gerenciamento

Conclua as etapas a seguir para excluir sua política:

1. No menu de navegação, clique em **Políticas**.
2. Na guia *Todas as políticas*, selecione o ícone **Opções** para a política que você deseja excluir.
3. Clique em **Remover**.
4. Na caixa de diálogo *Remover política*, clique em **Remover política**.

Sua política é excluída.

Trabalhando com a IBM Multicloud Manager descoberta de serviço

Configure o IBM Multicloud Manager para que seu managed cluster do IBM Multicloud Manager possa descobrir serviços Kubernetes, serviços Ingress e serviços Istio.

- Visão geral de descoberta de serviço do IBM Multicloud Manager
- Preparando seu IBM Multicloud Manager para descobrir serviços
- Ativando um serviço Kubernetes para descoberta
- Ativando um ingresso do Kubernetes para descoberta
- Ativando um serviço Istio para descoberta

Visão geral de serviços do IBM Multicloud Manager

Um serviço do IBM Multicloud Manager pode ser um dos seguintes tipos: serviço Kubernetes, ingresso do Kubernetes ou Istio Gateway. Um serviço do IBM Multicloud Manager é executado em um único managed-cluster ou em múltiplos managed-clusters. Eles suportam serviços Kubernetes, serviços de ingresso do Kubernetes e serviços Gateway Istio.

As seções a seguir fornecem um resumo dos tipos de serviços que estão disponíveis para o IBM Multicloud Manager.

Serviço do Kubernetes

Um recurso *service* é um recurso de serviço Kubernetes. Edite a seção `spec` de seu arquivo `yaml` de definição de serviço para definir seu serviço com rótulos. O exemplo a seguir mostra um recurso de serviço Kubernetes:

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    mcm.ibm.com/service-discovery: "{}"
  name: dbservice
  namespace: database
spec:
  type: LoadBalancer
  ports:
  - name: http
    nodePort: 8080
    port: 8000
    protocol: TCP
  selector:
    app: dbservice
```

Serviço de ingresso

Um serviço *Ingress* é um ingresso do Kubernetes que define os critérios nos quais os serviços no managed-cluster podem se comunicar com outros managed-clusters. Edite a seção `spec` do arquivo `yaml` de definição de serviço de ingresso para definir as regras do Ingress. Consulte a amostra a seguir de uma definição de serviço do Ingress:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: dbing
  namespace: database
  annotations:
    mcm.ibm.com/service-discovery: "{}"
spec:
  rules:
  - host: mydb.database.mcm.svc
    http:
      paths:
      - path: /db
        backend:
         serviceName: dbservice
         servicePort: 8000
```

Serviço de Gateway

Um serviço de *gateway* é usado para definir um Gateway Ingress. Um Gateway Istio é usado para expor um serviço Istio fora da malha de serviço Istio. Deve-se ativar um Gateway Istio para expor um serviço Istio. Edite a seção `spec` de seu arquivo `yaml` de definição de gateway Istio para definir seu gateway.

Consulte o serviço de gateway de amostra a seguir:

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: dbgateway
  namespace: database
spec:
  selector:
    istio: ingressgateway
```

```
servers:
- port:
    number: 80
    name: http
    protocol: HTTP
  hosts:
  - "mydb.database.global"
```

Consulte Trabalhando com a descoberta de serviço do IBM Multicloud Manager para obter mais informações sobre os serviços.

Preparando seu IBM Multicloud Manager para descobrir serviços

É possível configurar o registro de serviço do IBM Multicloud Manager para descobrir serviços Kubernetes, serviços de ingresso do Kubernetes e serviços Istio que estejam em diferentes managed-clusters do IBM Multicloud Manager.

Quando você tem múltiplas instâncias de um serviço Kubernetes, de um serviço de ingresso do Kubernetes ou de um serviço Istio que são gerenciadas pelo IBM Multicloud Manager, é um desafio mantê-las. A função de descoberta de serviço IBM Multicloud Manager descobre apenas os serviços Kubernetes, os serviços de ingresso do Kubernetes e os serviços Istio que estão configurados para que sejam descobertos.

Tipo de usuário ou nível de acesso necessário: Administrador de cluster.

Depois que o hub-cluster e o managed-cluster do IBM Multicloud Manager são configurados, é necessário concluir as etapas a seguir para configurar seu componente de registro de serviço:

Configurar DNS

Configure o DNS para cada managed-cluster concluindo estas etapas:

1. Localize o IP do cluster de serviço `mcm-svc-registry-dns` inserindo o comando a seguir, em que é o namespace que contém seu componente de registro:

   ```
   kubectl get -n <ibm-klusterlet-namespace> service mcm-svc-registry-dns -o
   jsonpath='{.spec.clusterIP}'
   ```

2. Defina a configuração do DNS do cluster inserindo o comando a seguir:

   ```
   kubectl edit -n kube-system configmap kube-dns
   ```

3. Ative o plug-in de encaminhamento na configuração `kube-dns`, semelhante ao exemplo a seguir, em que é o endereço IP que você identificou na etapa 1:

   ```
   Corefile: |
    .: 53 {

        ...

        forward mcm.svc. <mcm-svc-registry-dns-service-cluster-ip>
    }
   ```

   O valor `mcm.svc` é o sufixo de domínio DNS padrão para o registro de serviço. Se você deseja configurá-lo para outro valor, conclua as etapas a seguir:

   1. Efetue login em seu IBM Cloud Private console de gerenciamento.
   2. Navegue para **Cargas de trabalho** -> **Liberações do Helm**.
   3. Selecione sua liberação `ibm-klusterlet`.
   4. Expanda **Todos os parâmetros**.
   5. Na seção *Configuração de registro de serviço do Multicloud Manager*, configure o sufixo DNS no campo *Sufixo DNS*.

      Dica: se o seu managed-cluster for um IBM Cloud Kubernetes Service, também será possível configurar seu DNS do cluster executando o comando a seguir:

      ```
      kubectl edit -n kube-system configmap coredns
      ```

Ativando um serviço Kubernetes para descoberta

É possível configurar o registro de serviço do IBM Multicloud Manager para descobrir serviços Kubernetes que estejam em diferentes managed-clusters do IBM Multicloud Manager.

Um serviço Kubernetes expõe um grupo de pods que estão em execução em um managed-cluster. A função de descoberta de serviço do IBM Multicloud Manager descobre apenas os serviços Kubernetes que estão configurados para que sejam descobertos.

Tipo de usuário ou nível de acesso necessário: Administrador de cluster.

Descubra o serviço Kubernetes

Para descobrir um serviço Kubernetes em seus managed-clusters, conclua as etapas a seguir:

1. Anote o serviço para ativar a descoberta de serviço.

   Deve-se ativar o serviço no managed-cluster a ser descoberto incluindo a anotação a seguir no arquivo `yaml` de definição de serviço para o serviço que você deseja descobrir:

   ```
   mcm.ibm.com/service-discovery
   ```

   O exemplo a seguir mostra o formato para o `dbservice`:

   ```
   apiVersion: v1
   kind: Service
   metadata:
    annotations:
      mcm.ibm.com/service-discovery: "{}"
    name: dbservice
    namespace: database
   spec:
    type: LoadBalancer
    ports:
    - name: http
      nodePort: 8080
      port: 8000
      protocol: TCP
    selector:
      app: dbservice
   ```

   Nota: o serviço Kubernetes que você deseja descobrir deve ser um tipo `LoadBalancer` ou `NodePort`.

2. Se o serviço para o qual você incluiu a anotação de descoberta tiver outros aplicativos que dependem dele, inclua o serviço na entrada do aplicativo como uma dependência implementável.

   O exemplo a seguir mostra como incluir essa dependência:

   ```
   apiVersion: apps.ibm.com/v1alpha1
   kind: Deployable
   metadata:
    name: name1
    namespace: workspace
   spec:
    template:
      apiVersion: extensions/v1beta1
      kind: Deployment
      metadata:
        name: ibm-websphere
        labels:
          app: ibm-websphere
      spec:
        replicas: 1
        selector:
          matchLabels:
            app: ibm-websphere
        template:
          metadata:
            labels:
              app: ibm-websphere
          spec:
            containers:
   ```

```
          - name: ibm-websphere
            image: "registry.ng.bluemix.net/seed/ibm-websphere-sample"
            imagePullPolicy: Always
 dependencies:
 - name: dbservice
   namespace: database
   kind: Service
   apiGroup: v1
placement:
   clusterNames:
   - managed-cluster1
```

O serviço dependente implementável é descoberto automaticamente no cluster (`managed-cluster1`) no qual o aplicativo é implementado.

3. Acesse o serviço descoberto usando o formato a seguir:

   ```
   <service-name>.<service-namespace>.<service-registry-dns-suffix>
   ```

Um exemplo do formato é: `dbservice.database.mcm.svc`.

Ativando um ingresso do Kubernetes para descoberta

É possível configurar o registro de serviço do IBM Multicloud Manager para descobrir ingressos do Kubernetes que estejam em managed-clusters do IBM Multicloud Manager diferentes.

Quando você tem múltiplos ingressos do Kubernetes que são gerenciados pelo IBM Multicloud Manager, é um desafio mantê-los. A função de descoberta de serviço do IBM Multicloud Manager descobre ingressos do Kubernetes que estão configurados para que sejam descobertos.

Tipo de usuário ou nível de acesso necessário: Administrador de cluster.

Ative o plug-in de descoberta kube-ingress

O plug-in para a descoberta de ingresso do Kubernetes deve ser ativado se você deseja descobrir um ingresso do Kubernetes em seus managed-clusters. Conclua as etapas a seguir para ativar o plug-in de ingresso:

1. Efetue login em seu IBM Cloud Private console de gerenciamento.
2. Navegue para **Cargas de trabalho** -> **Liberações do Helm**.
3. Selecione sua liberação `ibm-klusterlet`.
4. Expanda **Todos os parâmetros**.
5. Na seção *Configuração de registro de serviço do Multicloud Manager*, é possível ativar o plug-in que você deseja usar inserindo-o no campo **Plug-ins ativados**. Sua entrada deve ser separada por vírgulas, como o exemplo a seguir:

   ```
   kube-service,kube-ingress
   ```

Descubra o ingresso do Kubernetes

Para descobrir um ingresso do Kubernetes dentro de seus managed-clusters, conclua as etapas a seguir:

1. Anote um ingresso com a anotação de descoberta de serviço.

   Deve-se ativar o ingresso no managed-cluster a ser descoberto incluindo a anotação a seguir no arquivo `yaml` para o ingresso que você deseja descobrir:

   ```
   mcm.ibm.com/service-discovery
   ```

   O exemplo a seguir mostra como incluir isso no ingresso:

   ```
   apiVersion: extensions/v1beta1
   kind: Ingress
   metadata:
    name: dbing
    namespace: database
    annotations:
   ```

```
    mcm.ibm.com/service-discovery: "{}"
spec:
 rules:
 - host: mydb.database.mcm.svc
   http:
     paths:
     - path: /db
       backend:
        serviceName: dbservice
        servicePort: 8000
```

Dica: é possível anexar o sufixo DNS de registro de serviço (mcm.svc) ao nome do host de ingresso, o que permite acessar o host de ingresso diretamente usando o nome do host.

2. Se o ingresso no qual você incluiu a anotação de descoberta tiver outros aplicativos que dependem dele, inclua o ingresso na entrada do aplicativo como uma dependência implementável.

   O exemplo a seguir mostra como incluir essa dependência:

```
apiVersion: apps.ibm.com/v1alpha1
kind: Deployable
metadata:
 name: name1
 namespace: workspace
spec:
 template:
   apiVersion: extensions/v1beta1
   kind: Deployment
   metadata:
     name: ibm-websphere
     labels:
       app: ibm-websphere
   spec:
     replicas: 1
     selector:
       matchLabels:
         app: ibm-websphere
     template:
       metadata:
         labels:
           app: ibm-websphere
       spec:
         containers:
         - name: ibm-websphere
           image: "registry.ng.bluemix.net/seed/ibm-websphere-sample"
           imagePullPolicy: Always
 dependencies:
 - name: dbing
   namespace: database
   kind: Ingress
   apiGroup: extensions/v1beta1
 placement:
   clusterNames:
   - managed-cluster1
```

   Após a aplicação desse implementável, seu ingresso dependente é descoberto automaticamente no cluster (`managed-cluster1`) no qual o aplicativo está implementado.

3. Acesse o ingresso descoberto usando o nome do host do ingresso. Neste exemplo, o nome do host é `mydb.database.mcm.svc`.

Ativando um serviço Istio para descoberta

É possível configurar o registro de serviço do IBM Multicloud Manager para descobrir serviços Istio que estejam em diferentes managed-clusters do IBM Multicloud Manager.

Quando você tem múltiplas instâncias de um serviço Istio que são gerenciadas pelo IBM Multicloud Manager, é um desafio mantê-las. A função de descoberta de serviço do IBM Multicloud Manager descobre serviços Istio que estão configurados para que sejam descobertos.

Tipo de usuário ou nível de acesso necessário: Administrador de cluster.

Ative o plug-in de descoberta do Istio

O plug-in para a descoberta do Istio deve ser ativado se você deseja descobrir um serviço Istio em seus managed-clusters. Conclua as etapas a seguir para ativar um plug-in do Istio:

1. Efetue login em seu IBM Cloud Private console de gerenciamento.
2. Navegue para **Cargas de trabalho** -> **Liberações do Helm**.
3. Selecione sua liberação `ibm-klusterlet`.
4. Expanda **Todos os parâmetros**.
5. Na seção *Configuração de registro de serviço do Multicloud Manager*, é possível ativar o plug-in que você deseja usar inserindo-o no campo **Plug-ins ativados**. Sua entrada deve ser separada por vírgulas, como o exemplo a seguir:

   ```
   kube-service,istio
   ```

   Nota: se você ativar o plug-in Istio, será necessário instalar o istio-coredns-plugin em seu sistema Istio e certificar-se de que o sistema Istio tenha um balanceador de carga externo.

Descubra o serviço Istio

Para descobrir um serviço Istio dentro dos managed-clusters, conclua as etapas a seguir:

1. Exponha um serviço Istio fora da malha de serviço usando um gateway Istio.

   Consulte Controlar tráfego de ingresso para obter informações sobre como usar o Gateway Istio para expor seu Serviço Istio.

2. O gateway Istio que você incluiu deve ser semelhante ao exemplo a seguir:

   ```
   apiVersion: networking.istio.io/v1alpha3
   kind: Gateway
   metadata:
    name: dbgateway
    namespace: database
   spec:
    selector:
      istio: ingressgateway
    servers:
    - port:
        number: 80
        name: http
        protocol: HTTP
      hosts:
      - "mydb.database.global"
   ```

   Nota: o nome do host deve terminar com o sufixo *.global*. Isso é necessário para o `istio-coredns-plugin`.

3. Anote o gateway Istio com a anotação de descoberta de serviço.

   Deve-se ativar o gateway Istio no cluster gerenciado a ser descoberto incluindo a anotação a seguir no arquivo `yaml` de definição do Istio para o gateway que você deseja descobrir:

   ```
   mcm.ibm.com/service-discovery
   ```

   O exemplo a seguir mostra como incluir isso no ingresso:

   ```
   apiVersion: networking.istio.io/v1alpha3
   kind: Gateway
   metadata:
    name: dbgateway
    namespace:
    annotations:
      mcm.ibm.com/service-discovery: "{}"
   spec:
    selector:
      istio: ingressgateway
    servers:
    - port:
   ```

```
      number: 80
      name: http
      protocol: HTTP
    hosts:
     - "mydb.database.global"
```

4. Se o aplicativo Istio para o qual você incluiu a anotação de descoberta tiver outros aplicativos que dependem dele, inclua o gateway anotado na entrada do aplicativo como uma dependência implementável.

   O exemplo a seguir mostra como incluir essa dependência:

```
apiVersion: apps.ibm.com/v1alpha1
kind: Deployable
metadata:
 name: name1
 namespace: workspace
spec:
 template:
   apiVersion: extensions/v1beta1
   kind: Deployment
   metadata:
     name: ibm-websphere
     labels:
       app: ibm-websphere
   spec:
     replicas: 1
     selector:
       matchLabels:
         app: ibm-websphere
     template:
       metadata:
         labels:
           app: ibm-websphere
       spec:
         containers:
         - name: ibm-websphere
           image: "registry.ng.bluemix.net/seed/ibm-websphere-sample"
           imagePullPolicy: Always
 dependencies:
 - name: dbgateway
   namespace: database
   kind: Gateway
   apiGroup: networking.istio.io/v1alpha3
placement:
   clusterNames:
   - managed-cluster1
```

   Após a aplicação desse implementável em um sistema Istio, seu gateway dependente é descoberto automaticamente no cluster (`managed-cluster1`) no qual o aplicativo está implementado.

5. Acesse o ingresso descoberto usando o nome do host do gateway. Neste exemplo, o nome do host é `mydb.database.global`.

Serviços com recursos

Visualize a documentação do produto mais recente para os serviços e aplicativos integrados do IBM Multicloud Manager.

Serviços integrados do IBM Multicloud Manager

- Integração do Cloud Automation Manager e do IBM Multicloud Manager
- Gerenciamento de eventos para o IBM Multicloud Manager

Gerenciamento de eventos para o IBM Multicloud Manager

É possível visualizar e gerenciar múltiplos clusters ao instalar o Event Management for IBM Multicloud Manager. Usando o Gerenciamento de Eventos, é possível consolidar informações de seus sistemas de monitoramento e resolver problemas. Os eventos indicam que algo aconteceu em um aplicativo, serviço ou outro objeto monitorado. Todos os eventos que estão

relacionados a um único aplicativo ou a um cluster específico são correlacionados a um incidente. O Event Management pode receber eventos de várias origens de monitoramento, seja no local ou na nuvem.

Com o IBM Multicloud Manager, é possível assegurar que seus clusters do IBM Cloud Private estejam seguros, operando de forma eficiente e fornecendo os níveis de serviço que os aplicativos esperam. Para obter informações adicionais sobre como instalar o IBM Multicloud Manager, consulte Instalando o IBM Multicloud Manager em um cluster do IBM Cloud Private.

Instalando o Event Management for IBM Multicloud Manager

É possível fazer download do componente opcional IBM Cloud Event Management para o IBM Multicloud Manager a partir do website IBM Passport Advantage e, em seguida, implementar seus gráficos. Instale o IBM Multicloud Manager em seu hub-cluster e o IBM Multicloud Manager Klusterlet em todos os clusters que você deseja gerenciar.

Para obter o procedimento de instalação completo, consulte IBM Cloud Event Manager IBM Multicloud ManagerConfiguração na documentação do produto.

Para obter informações adicionais sobre o Cloud Event Management, consulte a documentação do produto Cloud Event Management.

Integração do Cloud Automation Manager e do IBM Multicloud Manager

O IBM Cloud Automation Manager pode ser integrado com o IBM Multicloud Manager. O IBM Cloud Automation Manager é uma plataforma de gerenciamento multicloud, de autoatendimento que é executada no IBM Cloud Private. Com a integração do IBM Multicloud Manager, é possível assegurar que seus clusters estejam protegidos, operando de forma eficiente e entregando níveis de serviço apropriados.

Para obter informações adicionais sobre o Cloud Automation Manager, consulte Visão geral do Cloud Automation Manager.

Pré-requisitos

Para obter uma lista completa de pré-requisitos, consulte Pré-requisitos para instalar o Cloud Automation Manager.

Configurando o Cloud Automation Manager e o IBM Multicloud Manager

Para instalar o Cloud Automation Manager, consulte Instalando.

Para obter informações adicionais sobre o IBM Multicloud Manager, consulte a visão geral do produto IBM Multicloud Manager.

Integrando com o IBM Multicloud Manager

É possível usar um modelo Terraform que é fornecido com o Cloud Automation Manager 3.1.2 e mais recente. Este modelo carrega o archive PPA e binários do IBM Multicloud Manager em um cluster do Kubernetes. Para obter o procedimento de configuração completo, consulte Integrando-se com o Multicloud Manager IBM Multicloud Manager Configuração na documentação do produto.

IBM Multicloud Manager Resolução de problemas

Saiba como isolar e resolver problemas com o IBM Multicloud Manager.

Verifique se seus problemas não estão relacionados a requisitos do sistema operacional, como disco, memória e capacidades de CPU. Para obter mais informações sobre os requisitos do sistema para o IBM® Cloud Private, consulte Requisitos do sistema. Consulte Preparando-se para a instalação do IBM Multicloud Manager para conhecer os requisitos do IBM Multicloud Manager.

Para obter suporte, consulte Suporte na documentação do produto.

- Uma política de conformidade não é aplicada a um cluster gerenciado
- Reinicie e limpe o MongoDB
- Problemas de instalação e configuração
- Problemas de integração

Uma política não é cumprida no cluster gerenciado

Depois de instalar o Klusterlet, as políticas que você criou em seu arquivo `.yaml` em seu hub-cluster não são cumpridas em seu cluster gerenciado.

Sintomas

Falha na implementação do documento sobre políticas.

Causas

Uma política criada não é aplicada no cluster remoto porque o `policy-template` não existe no seu arquivo `.yaml` do cluster gerenciado.

Resolvendo o problema

1. Verifique se o objeto `policy-template` existe no namespace IBM Multicloud Manager no hub-cluster executando o seguinte comando:

   ```
   kubectl describe pod $POD -n $POD_NS | grep "mcm-ns"
   ```

2. Execute o seguinte comando para verificar se existem políticas nos namespaces IBM Multicloud Manager:

   ```
   kubectl get policies -- all-namespaces
   ```

3. Verifique se seu cluster está registrado para os namespaces do IBM Multicloud Manager. Execute o comando a seguir:

   ```
   kubectl get clusters -- all-namespaces
   ```

4. Verifique se sua política existe em seu namespace. Execute o comando a seguir para receber detalhes sobre sua política:

   ```
   kubectl get policy -n $POD_NS
   ```

5. Verifique se seu arquivo `spec` possui os atributos a seguir:

   ```
   remediationAction: "enforce" # enforce or inform
   complianceType: "musthave" # used as default, when missing in a particular sub-template
   namespaces:
    include: ["default"]
    exclude: ["kube*"]
   ```

   Nota: o valor *inform* relata se o cluster está em conformidade com as políticas especificadas.

Sua política é cumprida em seu cluster remoto.

Reinicie e limpe o MongoDB

Ao usar o MongoDB, se você obtiver um erro de CLI a partir do comando `resourceview` ou na visualização *Topologia* do console, será necessário reiniciar o MongoDB.

Sintomas

Ao executar `kubectl get resourceview <resource_name>`, você deve obter os dados que são buscados por esse comando mas, em vez disso, é possível obter:

```
Nenhum recurso localizado.
```

Na console de gerenciamento do IBM Multicloud Manager, sua visualização *Topologia* retorna o erro a seguir:

```
An unexpected error occurred. Try again.
```

Causas

O MongoDB no hub-cluster pode estar no estado de erro e pode precisar de uma reinicialização.

Resolvendo o problema

1. É necessário reiniciar o MongoDB no hub-cluster.
    - Efetue login no IBM Multicloud Manager console de gerenciamento ou em seu cluster do hub com o comando a seguir, em que `cluster_host_name` é o nome do host externo ou o endereço IP para seu nó principal ou nós principal líder:

      ```
      cloudctl login -a https://<cluster_host_name>:8443 --skip-ssl-validation
      ```

    - Exclua o pod do mongodb com o comando a seguir:

      ```
      kubectl delete pod -l app=ibm-mcm-prod -l component=mongodb -n kube-system
      ```

2. Depois de reiniciar o MongoDB em seu hub-cluster, é necessário reiniciar seu Klusterlet em cada cluster gerenciado.
    - Efetue login no IBM Multicloud Manager console de gerenciamento ou em seu cluster do hub com o comando a seguir, em que `cluster_host_name` é o nome do host externo ou o endereço IP para seu nó principal ou nós principal líder:

      ```
      cloudctl login -a https://<cluster_host_name>:8443 --skip-ssl-validation
      ```

    - Execute o comando a seguir para excluir o pod do Klusterlet:

      ```
      kubectl delete pod -l app=ibm-mcmk-prod -l component=klusterlet -n kube-system
      ```

IBM Multicloud Manager problemas de instalação e configuração

Revise os problemas de instalação e configuração com o IBM Multicloud Manager

- O IBM Multicloud Manager é interrompido quando desinstalado
- O cluster não aparece no IBM Multicloud Manager após a reinstalação do Klusterlet

IBM Multicloud Manager problemas de integração

Os eventos não aparecem no Gerenciamento de Eventos para o IBM Multicloud Manager.

Sintomas

Os eventos aparecem no Gerenciador de Alertas, mas não aparecem no Gerenciamento de Eventos console.

Resolvendo o problema

O Gerenciamento de Eventos contém um pod que quebra os eventos recebidos. Reinicie o pod para reiniciar o fluxo de eventos.

1. Obtenha o nome do pod para o processo de brokers de Gerenciamento de Eventos. Execute o comando a seguir:

   ```
   kubectl get pods | grep brokers
   ```

2. Exclua o pod. O pod é reinicializado automaticamente. Execute o comando a seguir:

   ```
   kubectl delete pod <brokers pod name>
   ```

3. Verifique se há novos incidentes na console de Gerenciamento de Eventos. Execute o comando a seguir:

   ```
   kubectl get pods | grep brokers
   ```

Visualização de tecnologia

Alguns recursos estão disponíveis nessa versão do IBM® Cloud Private apenas como código de visualização de tecnologia (TPC).

Código de visualização de tecnologia

O TPC está incluído no software IBM Cloud Private e está sujeito aos seguintes termos de Informações sobre Licença relacionadas ao TPC:

CÓDIGO DE VISUALIZAÇÃO DE TECNOLOGIA: o Código de Visualização de Tecnologia (TPC) pode ser incluído ou distribuído com o Programa ou com atualizações para ele. O TPC será identificado como tal no Arquivo de Avisos (ou em um Arquivo de Avisos atualizado que acompanha as atualizações) ou em um arquivo ou arquivos referenciados em tal Arquivo de Avisos. Alguns ou todos do TPC não podem ser disponibilizados de maneira geral pela IBM como ou num produto. Você está autorizado a usar o TPC apenas para uso interno para propósitos de avaliação e não para uso em um ambiente de produção. O Arquivo de Avisos pode limitar este uso de avaliação a um período de avaliação. Nesse caso, no final do período de avaliação, será necessário parar o uso e desinstalar o TPC. Nenhum suporte é fornecido para TPC e este é fornecido "NO ESTADO EM QUE SE ENCONTRAM (""AS IS"")" SEM GARANTIA DE QUALQUER ESPÉCIE (EXPRESSA OU IMPLÍCITA), INCLUINDO, SEM LIMITAÇÃO, QUALQUER GARANTIA IMPLÍCITA DE COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO/FIM. Não é possível transferir o TPC para outra parte. O TPC pode conter um dispositivo de desativação que impedirá que ele seja usado após o período de avaliação terminar. Você não corromperá essa desativação do dispositivo ou do TPC. É necessário tomar as devidas precauções para evitar a perda de dados que possa ocorrer quando o TPC não puder mais ser usado.

- Incluindo um nó do trabalhador do Windows no cluster do IBM Cloud Private
- IBM Cloud Private Detector de problemas do nó e Draino
- Gerenciar o kube-proxy usando o IPVS
- Ajuste de escala horizontal automático do pod usando métricas customizadas
- Instalando o IBM® Cloud Private usando o containerd
- Restringindo o acesso aos serviços de plataforma
- Login do IBM Cloud Private com o IBM® Z
- Controlador de política de mutação
- Serviço de funcionamento do sistema IBM Cloud Private
- Instalando o IBM Cloud Private com o IBM Cloud Kubernetes Service
- Definindo configurações de failover para seus clusters do IBM Multicloud Manager
- Instalando o Knative no IBM Cloud Private

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Incluindo um nó do trabalhador do Windows no cluster do

IBM Cloud Private

Como uma visualização de tecnologia, é possível incluir um Windows™ nó do trabalhador em um cluster do IBM Cloud Private existente. Posteriormente, é possível implementar um aplicativo Windows para o nó Windows.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

- Requisitos do Sistema
- Recursos Suportados
- Pré-requisitos
- Planejando sua Topologia de Rede
- Desativando o IP do calico-in-IP
- Preparando o Windows nó do trabalhador
- Incluindo o nó do trabalhador do Windows no cluster
- Implementando um Windows serviço de amostra
- Problemas e Limitações Conhecidos
- Resolução de problemas

Requisitos do sistema

Revise os requisitos do sistema a seguir:

Requisitos de hardware:

Tabela 1. Requisitos mínimos de hardware

Tipo de Nó	N£mero	CPU	Memória	Disco
Nó do trabalhador	>=1	2	>=4 GB	> 50 GB

Sistemas operacionais e plataformas suportados

Tabela 2. Sistemas Operacionais Suportados

Plataforma	Sistema Operacional
Windows	Windows Server versão 1803, 1809

Versão e tipo de contêiner suportados do Docker

Tabela 3. Versões Suportadas do Docker

Plataforma	Docker EE	Tipo de contêiner
Windows	17.06.1-ee-2 ou mais recente	Contêineres de servidor Windows

Recursos Suportados

Tabela 4. Recursos suportados

Recurso	Windows nó do trabalhador
Implementando o aplicativo Windows no cluster do IBM Cloud Private	S
Containerd	N
IBM Cloud Private Cloud Foundry	N
Cloud Automation Manager	N
IBM Cloud Private-CE	S
Tarefas de Instalação	N
IPSec	N
Criação de Log	N
Medição	N
Aplicativo Prometheus	N
Rede: Calico	N
Rede: NSX-T	N
Suporte à GPU do Nvidia	N
Armazenamento: GlusterFS	N
Armazenamento: VMware	N
Armazenamento: Minio	N
Criptografia de	N
Consultor de Vulnerabilidade	N

Pré-requisitos

- O IBM Cloud Private cluster está configurado e em execução corretamente.
- O Calico está ativado para rede.
- O IP-in-IP está desativado para o Calico. Consulte Disabling Calico IP-in-IP .
- O nome do host do trabalhador Windows é resolvível no cluster.
- O firewall do host do trabalhador do Windows está desativado.

Planejando sua Topologia de Rede

Há várias configurações de rede suportadas com o Kubernetes no Windows. Para obter mais informações, consulte Rede do KubernetesWindows.

- Calico: Para rede de nós do Linux® e política de cumprimento de rede.
- Host-Gateway: as rotas de IP Static são configuradas diretamente em cada nó do cluster. O Host-Gateway é usado para os nós a seguir:
  - Windows nós internetworking
  - Interligação de redes de nós do Linux e nós do Windows

Nota: para minimizar o impacto em redes de cluster do IBM Cloud Private existentes, deve-se verificar e suportar que o Host-Gateway seja usado como a solução de rede para o Windows para integração com o IBM Cloud Private.

Desativando o IP do Calico-in-IP

1. Instale a CLI do Calico. Para obter detalhes, consulte Instalando a CLI do Calico (calicoctl).

2. Para obter a especificação do conjunto de IP atual (`ippool`) do ambiente, execute os comandos a seguir no nó principal:

```
export ETCD_ENDPOINTS=https://<MASTERIP>:4001
export ETCD_CERT_FILE=/etc/cfc/conf/etcd/client.pem
export ETCD_KEY_FILE=/etc/cfc/conf/etcd/client-key.pem
export ETCD_CA_CERT_FILE=/etc/cfc/conf/etcd/ca.pem

calicoctl get ippool default-ipv4-ippool -o yaml > ippool.yaml
cat ippool.yaml
```

   O conteúdo do arquivo `ippool.yaml` é mostrado no exemplo a seguir:

```
apiVersion: projectcalico.org/v3
kind: IPPool
metadata:
creationTimestamp: 2019-01-28T16:46:29Z
name: default-ipv4-ippool
resourceVersion: "94911"
uid: 42d2e92c-231c-11e9-837b-000c295cba9c
spec:
cidr: 10.1.0.0/16
ipipMode: Always
natOutgoing: true
```

3. Para desativar o modo `ipip`, mude `Always` para `Never` e, em seguida, execute o comando a seguir para aplicar a mudança:

```
calicoctl apply -f ./ippool.yaml
```

Preparando o nó do trabalhador do Windows

1. Instale o Docker no servidor Windows:

```
Install-Module -Name DockerMsftProvider -Repository PSGallery -Force
Install-Package -Name docker -ProviderName DockerMsftProvider -Force
```

   Para obter mais informações, consulte a Documentação da Microsoft.

2. Reinicie o Windows host:

```
Reiniciar-Computador-Forçar
```

3. Depois que o host estiver em execução, inicie o serviço do Docker:

```
Iniciar-docker de serviço
```

4. Verifique a instalação do Docker. Por exemplo:

```
docker version
    Client:
    Version:           18.03.1-ee-4
    API version:       1.37
    Go version:        go1.10.2
    Git commit:        0ded23c
    Built:             Thu Oct 25 00:41:52 2018
    OS/Arch:           windows/amd64
    Experimental:      false
    Server:
    Engine:
    Version:          18.03.1-ee-4
    API version:      1.37 (minimum version 1.24)
    Go version:       go1.10.2
    Git commit:       0ded23c
    Built:            Thu Oct 25 00:56:17 2018
    OS/Arch:          windows/amd64
    Experimental:     false
```

5. Ative o encaminhamento de IP no trabalhador do Windows :

```
 PS C:\Users\Administrator> reg add HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters /v
IPEnableRouter /D 1 /f
```

6. Reinicie o Windows host:

```
Reiniciar-Computador-Forçar
```

7. Configure o SSH sem uma senha do nó do trabalhador do Windows para o nó principal:

```
ssh-keygen -b 4096 -f $Env:UserProfile\.ssh\id_rsa -N '""'

function ssh-copy-id([string]$userAtMachine){
$publicKey = "$ENV:USERPROFILE" + "/.ssh/id_rsa.pub"
if (!(Test-Path "$publicKey")){
    Write-Error "ERROR: failed to open ID file '$publicKey': No such file"
} else {
    & cat "$publicKey" | ssh $userAtMachine "umask 077; test -d .ssh || mkdir .ssh ; cat >>
.ssh/authorized_keys || exit 1"
}
}

ssh-copy-id -i $Env:UserProfile\.ssh\id_rsa.pub root@<ICPMasterIp>
```

Nota: substitua os valores que estão entre sinais de maior e menor (< >) de acordo com seu ambiente.

8. Faça download do pacote **IBM Cloud Private 3.2.0 for Windows (64-bit) Docker** mais recente. É possível fazer download do arquivo a partir do website do IBM Passport Advantage®.

9. Extraia o pacote:

```
Expand-Archive \Path\TO\ibm-cloud-private-win-x64-3.2.0.zip C:\
```

Incluindo o nó do trabalhador do Windows no cluster

1. Obtenha um `podCIDR` válido para o trabalhador do Windows.

    1. Obtenha o intervalo de CIDR de pod reservado que é usado para o nó do Linux no principal. Por exemplo:

        ```
        # ip route show | grep -Eo '[0-9.]+/[ 0-9 ] + ' |grep "^ 10.1\."
        10.1.13.128/26
        10.1.120.64/26
        10.1.130.0/26
        10.1.140.0/26
        ```

        Neste exemplo, `10.1` é o prefixo do parâmetro `clusterCIDR`. Consulte a Tabela 4 para obter a definição de `clusterCIDR`.

    2. Determine um CIDR de pod não reservado para o trabalhador do Windows com base no resultado anterior. Por exemplo, `10.1.141.0/ 26`.

2. Obtenha o `KubeDnsServiceIp` para o cluster:

```
 # kubectl get svc -n kube-system |grep kube-dns
 kube-dns                       ClusterIP   10.0.0.10      <none>        53/UDP,53/TCP
87m
```

10.0.0.10 é o valor para `KubeDnsServiceIp`

3. Inicie o script a seguir para incluir o nó, especificando os valores de parâmetros corretos:

```
cd C:\ibm-cloud-private-win-x64-3.2.0\

 .\join.ps1 -masterIp <ICPMasterIp> -clusterCIDR <ClusterCidr> -serviceCIDR <ServiceCidr> -
kubeDnsServiceIp <KubeDnsServiceIp> -podCIDR <PodCIDR> -license accept
```

Nota: execute `Get-Help .\join.ps1` para usar este script. Consulte a Tabela 5 para obter as definições de parâmetro.

4. Verifique os resultados ao executar o comando a seguir no nó principal:

```
# kubectl get node
 NAME           STATUS    ROLES          AGE     VERSION
 172.16.200.184   Ready     etcd,master   127m    v1.12.4+icp-ee
 172.16.200.208   Ready     worker         89m    v1.12.4+icp-ee
 172.16.200.210   Ready     proxy          89m    v1.12.4+icp-ee
 172.16.200.239   Ready     management     89m    v1.12.4+icp-ee
 shags1         Ready     worker         29m     v1.12.3
```

Observe que `shags1` é o nó do Windows .

5. Configure as rotas de IP estáticas nos nós do cluster:

   - Para o host do Windows, inclua uma rota para o CIDR de pod do Linux no IP privado do Linux.

     Por exemplo:

     ```
     route -p add 10.1.13.128/26   172.16.200.184
     route -p add 10.1.120.64/26   172.16.200.208
     route -p add 10.1.130.0/26    172.16.200.210
     route -p add 10.1.140.0/26    172.16.200.239
     ```

     Neste exemplo, 172.16.xx.xx é o IP de um nó do cluster do Linux.

     O CIDR de pod em cada nó do Linux pode ser obtido usando o seguinte comando:

     ```
     ip route show |grep blackhole
     ```

   - Para o host do Linux, inclua uma rota para o CIDR de pod do Windows no IP privado do Windows.

     Por exemplo:

     ```
     ip route add 10.1.141.0/ 26 via 172.16.215.209
     ```

     Nesse exemplo, 172.16.215.209 é o IP do nó do Windows. O valor **10.1.141.0/26** é o valor que é usado na Etapa 1.

Parâmetro	Descrição	Valor Padrão
masterIp	Endereço IP principal do IBM Cloud Private.	
clusterCIDR	Esta é uma sub-rede global que é usada por todos os pods no cluster. Cada nó é designado a uma	

sub-rede /24 menor a partir deste para seus pods a serem usados. Ele é igual a `network_cidr` que está definido em `config.yaml`. | 10.1.0.0/16 | | serviceCIDR | Uma sub-rede puramente virtual não roteável que é usada por pods para serviços de acesso uniformemente, independentemente da topologia de rede. Ela é convertida em um espaço de endereço roteável e a partir dele pelo kube-proxy em execução nos nós. Ela é igual a `service_cluster_ip_range` que está definido em `config.yaml`. | 10.0.0.0/16 | | kubeDnsServiceIp | O endereço IP do serviço "kube-dns" que é usado para resolução de DNS e descoberta de serviço de cluster. É possível obter seu valor da Etapa 2.| 10.0.0.10 | | podCIDR| Esta é a sub-rede não reservada do conjunto de IPs do Calico para alocar IPs para contêineres individuais. | | | licença | Contrato de licença do IBM Cloud Private. | aceitar |

Implementando um serviço de amostra do Windows

1. Permita que a imagem do Windows seja implementada a partir da política de imagem do IBM Cloud Private:

   ```
   kubectl editar ClusterImagePolicy -n kube-system
   ```

   Inclua o valor a seguir na seção de repositórios:

   ```
       - name: mcr.microsoft.com/windows / *
   ```

2. Execute o aplicativo do servidor da web para criar a implementação:

   ```
    # wget
   https://raw.githubusercontent.com/Microsoft/SDN/master/Kubernetes/flannel/l2bridge/manifests/si
   mpleweb.yml -O win-webserver.yaml

    # kubectl apply -f win-webserver.yaml
   ```

   Leva alguns minutos para puxar a imagem do núcleo do servidor do Windows. Após a implementação, dois pods estão no status de execução. Por exemplo:

   ```
    # kubectl get po -o wide
    NAME                              READY    STATUS    RESTARTS   AGE    IP             NODE
   NOMINATED NODE
    win-webserver-578967f9d4-6t2sx    1/1      Running   0          88m    10.1.141.46    shags1
   <none>
    win-webserver-578967f9d4-vcwqx    1/1      Running   0          88m    10.1.141.45    shags1
   <none>
   ```

   Dois contêineres são iniciados no servidor Windows. Por exemplo:

   ```
    # docker ps |findstr powershell
    cb51f5d23630        17b224ab9b3a            "powershell.exe -com???"   About an hour ago   Up
   About an hour                       k8s_windowswebserver_win-webserver-578967f9d4-
   ```

```
vcwqx_default_bb079250-2062-11e9-9104-00163e01c8b8_0
 8de0cbb092b8        17b224ab9b3a            "powershell.exe -com???"   About an hour ago   Up
About an hour                         k8s_windowswebserver_win-webserver-578967f9d4-
6t2sx_default_bb089f7e-2062-11e9-9104-00163e01c8b8_0
```

3. Verifique o pod e o serviço:

   - Localize o IP do pod:

     ```
     # kubectl get po -o wide
     NAME                              READY    STATUS    RESTARTS   AGE    IP             NODE
   NOMINATED NODE
     win-webserver-578967f9d4-6t2sx    1/1      Running   0          88m    10.1.141.46
   shags1        <none>
     win-webserver-578967f9d4-vcwqx    1/1      Running   0          88m    10.1.141.45
   shags1        <none>
     ```

   - Acesse o aplicativo de amostra Windows por meio do IP do pod:

     ```
     # curl 10.1.141.45:80
         <html><body><H1>Windows Container Web Server</H1><p>IP 10.1.141.45 callerCount 4
   <p>IP 10.1.141.45 callerCount 1 </body></html>
     ```

   - Localize o IP de serviço:

     ```
     # kubectl get svc
     NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)         AGE
     kubernetes      ClusterIP   10.0.0.1       <none>        443/TCP         4h45m
     win-webserver   NodePort    10.0.193.104   <none>        80:31700/TCP    94m
     ```

   - Verifique o acesso pelo IP de serviço:

     ```
     # curl 10.0.193.104
         <html><body><H1>Windows Container Web Server</H1><p>IP 10.1.141.45 callerCount 5
   <p>IP 10.1.141.45 callerCount 1 </body></html>
     ```

Problemas e Limitações Conhecidos

1. Todos os nós do cluster, incluindo o nó do Windows, devem estar na mesma sub-rede.
2. O contêiner do Windows não pode acessar a Internet.
3. Não é possível acessar o aplicativo Windows com o tipo de serviço de NodePort, já que a porta do nó do serviço está inacessível a partir do nó do cluster.

Resolução de problemas

Os problemas a seguir foram identificados e as resoluções estão disponíveis:

Erro lançado ao instalar o Docker

Sintomas:

```
PS C:\Users\Administrator> Install-Package -Name docker -ProviderName DockerMsftProvider -Force
WARNING: A restart is required to enable the containers feature. Reinicie sua máquina.
Install-Package : Cannot rename because item at 'C:\Program Files\dummyName' does not exist.
At line:1 char:1
+ Install-Package -Name docker -ProviderName DockerMsftProvider -Force
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (Microsoft.Power....InstallPackage:InstallPackage)
[Install-Package],
   Exception
    + FullyQualifiedErrorId :
InvalidOperation,Microsoft.PowerShell.Commands.RenameItemCommand,Microsoft.PowerShell.Pa
   ckageManagement.Cmdlets.InstallPackage
```

Causa :

Esse é um problema conhecido do Windows Server 2019. Para obter mais informações, consulte Problema do MicrosoftDockerProvider.

Resolvendo o problema :

Ao ver esse erro, é possível ignorá-lo e continuar o procedimento.

A junção do cluster falhou

Sintomas:

```
PLAY [Join Windows to ICP cluster] *****************************************************

TASK [Label the node shags1 as worker role]
Error from server (NotFound): nodes "shags1" not found
```

Causa :

O nó principal não pode reconhecer o nó Windows.

Resolvendo o problema :

É possível incluir o IP e nome do host do Windows em `/etc/hosts` de nós do Linux. Em seguida, execute `rm c:k\` no PowerShell do Windows e, em seguida, reúna os nós.

IBM Cloud Private detector de problemas do nó e Draino

Problemas podem surgir em nós que afetam os pods que estão em execução neles. Quando problemas são detectados, o IBM Cloud Private usa o detector de problemas do nó e o Draino para identificar nós com problemas e, em seguida, desfazer o planejamento deles e drená-los para que os problemas possam ser resolvidos e os pods replanejados.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

O detector de problemas do nó e o Draino coletam problemas do nó a partir de vários daemons e tornam os problemas visíveis para as camadas de envio de dados na pilha de gerenciamento de cluster. Quando ocorrem problemas, o IBM Cloud Private desfaz o planejamento (isola) os nós problemáticos imediatamente e os drena após uma quantia de tempo configurável. O padrão é 10 minutos. É possível replanejá-los (desfazer o isolamento) após os problemas serem resolvidos.

Para saber mais, consulte Projeto do Draino e Projeto node-problem-detector do Kubernetes para tentar este procedimento.

Pré-requisitos

Assegure-se de que cada nó no cluster do IBM Cloud Private tenha o diretório `/var/log/journal`. Se o diretório não existir, crie-o.

Tarefas de Instalação

É possível ativar o detector de problema de nó e o Draino durante a instalação por meio do arquivo `config.yaml` ou após a instalação na console de gerenciamento usando um gráfico do Helm.

Ativando o parâmetro node-problem-detector-draino durante a instalação do cluster

1. Seguindo o procedimento de instalação, durante a Etapa 3, customize seu cluster, abra o arquivo `/<installation_directory>/cluster/config.yaml`.
2. Na lista de serviços de gerenciamento, configure `node-problem-detector-draino` como `enabled`. Por exemplo:

   ```
   management_services:
    istio: disabled
    vulnerability-advisor: disabled
    storage-glusterfs: disabled
    storage-minio: disabled
    key-management-hsm: disabled
    platform-security-netpols: disabled
    node-problem-detector-draino: enabled
   ```

3. Salve e saia do arquivo.

O detector de problemas do nó e o Draino são instalados pelo instalador do IBM Cloud Private durante a instalação do cluster.

Instalando o gráfico node-problem-detector-draino para um cluster existente

Tipo de usuário ou nível de acesso necessário: administrador de cluster, administrador de equipe ou operador

1. Efetue login no console de gerenciamento IBM Cloud Private.
2. Clique em **Catálogo** .
3. Localize o gráfico node-problem-detector-draino usando a barra de procura.
4. Selecione o gráfico node-problem-detector-draino. Um arquivo leia-me exibe informações sobre a instalação, desinstalação, configuração e outros detalhes do gráfico para o node-problem-detector-draino.
5. Para configurar o gráfico, clique em **Configurar** .
6. Nomeie a liberação do Helm e selecione o namespace kube-system no menu. O nome deve consistir em caracteres alfanuméricos minúsculos ou caracteres de traço (-) e deve iniciar e terminar com um caractere alfanumérico.
7. Assegure-se de ler e concordar com o contrato de licença.
8. Opcional: customize os campos `Todos os parâmetros` para sua preferência.
9. Para implementar o gráfico node-problem-detector-draino e criar uma liberação do node-problem-detector-draino, clique em **Instalar**.

Verificando a instalação

Após a conclusão da instalação, verifique se o node-problem-detector-draino que você ativou foi criado e está em execução:

Assegure-se de que os pods do Kubernetes correspondentes estejam implementados e que todos os contêineres estejam ativos. Execute o comando a seguir:

```
kubectl -n kube-system get pods | grep -E "node-problem-detector|draino"

The output might resemble the following content:
npm-draino-57df88dc45-cls7r          1/1     Running     0       2h      10.1.96.125       <none>
npm-node-problem-detector-68x5s      1/1     Running     0       2h      10.1.16.147       <none>
npm-node-problem-detector-8rzkq      1/1     Running     0       2h      10.1.62.146       <none>
npm-node-problem-detector-b2vzb      1/1     Running     0       2h      10.1.75.82        <none>
npm-node-problem-detector-bgbs4      1/1     Running     0       2h      10.1.249.116      <none>
npm-node-problem-detector-ltvjn      1/1     Running     0       2h      10.1.96.126       <none>
npm-node-problem-detector-r2drx      1/1     Running     0       2h      10.1.93.218       <none>
npm-node-problem-detector-t99f2      1/1     Running     0       2h      10.1.32.179       <none>
```

Agora você está pronto para monitorar os nós do cluster.

Gerenciar kube-proxy usando IPVS

O IPVS (IP Virtual Server) é um recurso beta no Kubernetes 1.9.1. O modo `kube-proxy ipvs` fornece benefícios como o aprimoramento de desempenho para kube-proxy, quando comparado com métodos tradicionais de uso do modo iptables e userspace.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

IPVS em execução em um host atua como um balanceador de carga na frente de um cluster de servidores reais. O IPVS pode direcionar solicitações para serviços baseados em TCP/UDP para os servidores reais. O IPVS também pode fazer serviços que estão em execução nos servidores reais aparecem como um serviço virtual em um único endereço IP.

Para ativar o modo `kube-proxy ipvs`, deve-se configurar o parâmetro `kube_proxy_extra_args` no arquivo de configuração de cluster.

O arquivo de configuração para um cluster com `kube-proxy ipvs` ativado pode ser semelhante à saída a seguir:

```
## Kubernetes Settings
# kube_apiserver_extra_args: []
# kube_controller_manager_extra_args: []
kube_proxy_extra_args: ["--feature-gates=SupportIPVSProxyMode=true","--proxy-mode=ipvs"]
```

Após a instalação ser concluída, verifique se as regras IPVS foram criadas concluindo as etapas a seguir:

1. Revise o log do contêiner kube-proxy.

```
[root@testnode ~]# docker ps |grep proxy

6e8b9b058bfc
ibmcom/kubernetes@sha256:0a186c019bd7d3a078799a387663da93c162b290b0665d16b229dba7d8f060b7
"/hyperkube proxy ..."   11 minutes ago      Up 11 minutes
k8s_proxy_k8s-proxy-9.21.53.16_kube-system_97991d33fbaf5606a3a6113337710e27_0

docker logs 6e8b9b058bfc -f
...
I0131 13:24:32.282248       1 feature_gate.go:184] feature gates:
map[SupportIPVSProxyMode:true]
I0131 13:24:32.283828       1 server_others.go:180] Using ipvs Proxier.
I0131 13:24:32.284643       1 server_others.go:205] Tearing down inactive rules.
I0201 05:31:25.008541       1 server.go:426] Version: v1.11.0+icp-ee
...
```

2. Verifique se o modo de IPVS é detectado em cada nó do cluster.

```
ipvsadm -Ln
```

A saída se assemelha ao código a seguir:

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.17.0.1:31443 rr
  -> 10.1.20.84:9443              Masq    1      0          0
TCP  172.17.0.1:32292 rr
  -> 10.1.20.94:3000              Masq    1      0          0
TCP  9.21.53.16:30090 rr
  -> 10.1.20.80:9090              Masq    1      0          0
TCP  9.21.53.16:30296 rr
  -> 10.1.20.93:3001              Masq    1      0          0
TCP  9.21.53.16:31443 rr
  -> 10.1.20.84:9443              Masq    1      0          0
TCP  9.21.53.16:32292 rr
...
```

Ajuste automático de escala de pod horizontal usando métricas customizadas

O Horizontal Pod Autoscaler (HPA) no IBM Cloud Private permite que seu sistema escale automaticamente as cargas de trabalho para cima ou para baixo com base no uso de recurso. Esse ajuste automático de escala ajuda a garantir acordos de nível de serviço (SLAs) para as suas cargas de trabalho.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Por padrão, a política HPA escala automaticamente o número de pods com base na utilização da CPU observada. No entanto, em muitas situações, você pode desejar escalar o aplicativo com base em outras métricas monitoradas, como o número de solicitações recebidas ou o consumo de memória. Iniciando com o IBM Cloud Private Versão 3.2.0, você tem a capacidade de automatizar o ajuste de escala alavancando o Prometheus e o adaptador do Prometheus.

Prometheus

O Prometheus é amplamente usado para monitorar todos os componentes de um cluster do Kubernetes. Esses componentes incluem o plano de controle, os nós do trabalhador e os aplicativos que estão em execução no cluster.

Adaptador do Prometheus

O adaptador do Prometheus é a camada de agregador do do Kubernetes que instala APIs extras no estilo do Kubernetes e registra servidores de API customizados no cluster do Kubernetes. O adaptador reúne os nomes de métricas disponíveis do Prometheus em intervalos regulares e depois expõe as métricas para o HPA para ajuste automático de escala.

Preparando para a instalação

Por padrão, no IBM Cloud Private, o HPA é ativado para escala automática com base na utilização da CPU. Para ativar o ajuste de escala automático com base em métricas customizadas, deve-se remover a opção `custom-metrics-adapter` do parâmetro `disabled_management_services` no arquivo `/<installation_directory>/cluster/config.yaml`.

Seu arquivo de configuração pode ser semelhante ao código a seguir:

```
## Management Services Settings
## You can disable following services: custom-metrics-adapter, istio, metering, monitoring, service-catalog, storage-glusterfs, vulnerability-advisor
management_services:
  istio: disabled
  vulnerability-advisor: disabled
  storage-glusterfs: disabled
  storage-minio: disabled
```

Verificando a instalação

Após a instalação ser concluída, verifique se o `custom-metrics-adapter` está ativado.

1. Assegure-se de que o grupo de APIs `autoscaling/v2beta1` seja exibido.

   ```
   Kubectl api-versões |grep "autoscaling/v2beta1"
   ```

   A saída se assemelha ao código a seguir:

   ```
   Autoscaling/v2beta1
   ```

2. Assegure-se de que o pod `custom-metrics-adapter` seja implementado e esteja em um estado `running`.

   ```
   kubectl get po -n kube-system |grep custom-metrics-adapter
   ```

   A saída se assemelha ao código a seguir:

   ```
   custom-metrics-adapter-76d7bb8dcd-2pj4k                          1/1       Running     0
   18m
   ```

3. Liste as métricas customizadas padrão que são fornecidas pelo adaptador do Prometheus no pod.

   ```
   kubectl get --raw "/apis/custom.metrics.k8s.io/v1beta1" | jq .  |grep "pods/"
   ```

   A saída se assemelha ao código a seguir:

   ```
         "name": "pods/kube_pod_container_status_waiting_reason",
         "name": "pods/fs_read",
         "name": "pods/memory_failures",
         "name": "pods/kube_pod_status_phase",
         "name": "pods/kube_pod_container_resource_limits_memory_bytes",
         "name": "pods/cpu_user",
         "name": "pods/fs_usage_bytes",
         "name": "pods/tasks_state",
         "name": "pods/kube_pod_container_info",
         "name": "pods/cpu_cfs_throttled",
         "name": "pods/fs_sector_writes",
         "name": "pods/kube_pod_created",
         "name": "pods/network_tcp_usage",
         "name": "pods/spec_memory_limit_bytes",
         "name": "pods/network_udp_usage",
         "name": "pods/memory_max_usage_bytes",
         "name": "pods/spec_cpu_quota",
         "name": "pods/kube_pod_container_status_terminated_reason",
         "name": "pods/cpu_system",
         "name": "pods/kube_pod_container_status_running",
         "name": "pods/kube_pod_status_ready",
         "name": "pods/fs_io_time_weighted",
         "name": "pods/fs_reads_bytes",
         "name": "pods/kube_pod_info",
         "name": "pods/fs_reads_merged",
         "name": "pods/kube_pod_container_resource_requests_cpu_cores",
         "name": "pods/fs_io_time",
         "name": "pods/kube_pod_container_resource_limits_cpu_cores",
         "name": "pods/fs_inodes",
         "name": "pods/start_time_seconds",
         "name": "pods/kube_pod_container_status_terminated",
   ```

```
        "name": "pods/kube_pod_container_status_waiting",
        "name": "pods/cpu_usage",
        "name": "pods/spec_cpu_shares",
        "name": "pods/spec_memory_reservation_limit_bytes",
        "name": "pods/kube_pod_container_status_ready",
        "name": "pods/fs_writes_merged",
        "name": "pods/fs_inodes_free",
        "name": "pods/cpu_cfs_throttled_periods",
        "name": "pods/kube_pod_labels",
        "name": "pods/cpu_load_average_10s",
        "name": "pods/fs_io_current",
        "name": "pods/memory_working_set_bytes",
        "name": "pods/spec_memory_swap_limit_bytes",
        "name": "pods/fs_reads",
        "name": "pods/kube_pod_container_resource_requests_memory_bytes",
        "name": "pods/memory_rss",
        "name": "pods/cpu_cfs_periods",
        "name": "pods/fs_writes_bytes",
        "name": "pods/fs_writes",
        "name": "pods/last_seen",
        "name": "pods/spec_cpu_period",
        "name": "pods/kube_pod_start_time",
        "name": "pods/fs_write",
        "name": "pods/memory_failcnt",
        "name": "pods/kube_pod_container_status_restarts",
        "name": "pods/fs_sector_reads",
        "name": "pods/kube_pod_status_scheduled",
        "name": "pods/memory_cache",
        "name": "pods/memory_usage_bytes",
        "name": "pods/memory_swap",
        "name": "pods/fs_limit_bytes",
        "name": "pods/kube_pod_owner",
```

Exemplo: implementando um aplicativo com uma política HPA

Este exemplo mostra como escalar automaticamente um aplicativo da web nginx com base no uso de memória usando uma política HPA. Quando o `memory_usage_bytes` de um pod nginx é maior que 10 M, a política escalará o aplicativo da web nginx. Escalar para cima um aplicativo aumenta o número de pods disponíveis para uma implementação. Se o `memory_usage_bytes` de um pod nginx for menor que 10 M, o aplicativo passará por scale down, mas não será ajustado para menos que o número mínimo de réplicas especificadas para a implementação.

1. Crie o arquivo `podinfo-svc.yaml` usando o código a seguir:

```
---
apiVersion: v1
kind: Service
metadata:
  name: podinfo
  labels:
    app: podinfo
  annotations:
    prometheus.io/scrape: "true"
spec:
  type: NodePort
  ports:
   - port: 80
      targetPort: 80
      nodePort: 31198
      protocol: TCP
  selector:
    app: podinfo
```

2. Crie um serviço `podinfo` executando o comando a seguir:

```
kubectl create -f podinfo-svc.yaml
```

   A resposta é semelhante ao exemplo a seguir:

```
Serviço "podinfo" criado
```

3. Crie o arquivo `podinfo-dep.yaml` usando o código a seguir:

```
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: podinfo
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: podinfo
      annotations:
        prometheus.io/scrape: 'true'
    spec:
      containers:
      - name: podinfod
        image: nginx:1.8.1
        imagePullPolicy: Always
        ports:
        - containerPort: 80
          protocol: TCP
        resources:
          requests:
            memory: "32Mi"
            cpu: "1m"
          limits:
            memory: "256Mi"
            cpu: "100m"
```

4. Crie uma implementação `podinfo` executando o comando a seguir:

   ```
   kubectl create -f podinfo-dep.yaml
   ```

   A resposta é semelhante ao exemplo a seguir:

   ```
   implementação "podinfo" criada
   ```

5. Crie o arquivo `podinfo-hpa-custom.yaml` usando o código a seguir:

   ```
   ---
    apiVersion: autoscaling/v2beta1
    kind: HorizontalPodAutoscaler
    metadata:
      name: podinfo
    spec:
      scaleTargetRef:
        apiVersion: extensions/v1beta1
        kind: Deployment
        name: podinfo
      minReplicas: 2
      maxReplicas: 10
      metrics:
      - type: Pods
        pods:
          metricName: memory_usage_bytes
          targetAverageValue: 10485760
   ```

6. Crie uma política de HPA `podinfo` baseada no uso de memória de pod (memory_usage_bytes, `10485760 = 10M`) executando o comando a seguir:

   ```
   kubectl create -f podinfo-hpa-custom.yaml
   ```

   A resposta é semelhante ao exemplo a seguir:

   ```
    Horizontalpodautoscaler.autoscaling "podinfo" criado
   ```

7. Simule o carregamento usando o aplicativo `ab` Apache. Este aplicativo aciona uma carga de ajuste automático de escala.

   ```
   for a in `seq 1 50`; do ab -rSqd -c 200 -n 20000 <node_ip>:31198/;done
   ```

   `<node_ip>` é o endereço IP de um nó em seu cluster do IBM Cloud Private.

Instalando o IBM Cloud Private usando o containerd

`cri` é uma implementação de plug-in containerd que é usada pelo container runtime interface (CRI) do Kubernetes. Com `cri`, é possível executar o Kubernetes usando containerd como o tempo de execução do contêiner.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Antes de iniciar

- A instalação do IBM Cloud Private usando containerd é suportada somente nos sistemas operacionais Ubuntu 16.04 LTS que estão em execução no Linux®.
- Antes de concluir este procedimento, deve-se concluir as seções *Instalar o Docker somente para o seu nó de inicialização*, *Configurar o ambiente de instalação* e *(Opcional) Customizar o seu cluster* do documento de instalação para sua edição do IBM Cloud Private.
    - Instalando as edições Cloud Native, Enterprise e Community do IBM® Cloud Private

Configure o IBM Cloud Private para usar o tempo de execução do contêiner containerd

1. Atualize o arquivo `<installation_directory>/cluster/config.yaml`. Altere o parâmetro `container_runtime` para `containerd`.

   ```
   container_runtime: containerd
   ```

2. Continue com a seção *Implementar o ambiente* do documento de instalação para sua edição do IBM Cloud Private.

    - Instalando as edições Cloud Native, Enterprise e Community do IBM® Cloud Private

Restringindo o acesso aos serviços de plataforma

Configure políticas de rede de segurança de plataforma para restringir o acesso aos serviços de plataforma.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

Os serviços de plataforma IBM® Cloud Private são hospedados em namespaces, como `kube-system` e `cert-manager`. Todos os serviços hospedados pela plataforma podem ser acessados de todos os outros namespaces. É possível usar as políticas de rede de segurança de plataforma para restringir o acesso de outros namespaces. As políticas de rede de segurança não se aplicam a serviços que são hospedados na rede do host.

Ao configurar políticas de rede de segurança de plataforma, os seguintes serviços têm permissão para serem acessados por outros namespaces:

- Helm / Tiller
- Kube-api, etcd
- IBM Cloud Private ingresso de gerenciamento
- Medição
- Minio
- Armazenamento Persistente: GlusterFS
- Agrupamento de cluster

Importante: em um ambiente de alta disponibilidade, se você deseja ativar as políticas de rede de segurança de plataforma, deve-se configurar um endereço IP virtual (VIP). As políticas de rede de segurança da plataforma não podem ser ativadas em ambientes nos quais apenas o balanceador de carga está configurado e nenhum VIP está configurado.

Ative políticas de rede de segurança de plataforma no IBM Cloud Private

É possível ativar as políticas de rede de segurança de plataforma durante a instalação do IBM Cloud Private ou depois de instalar seu cluster do IBM Cloud Private.

Ativando políticas de rede de segurança de plataforma durante a instalação do

IBM Cloud Private

Ative `platform-security-netpols` na seção `management_services` no arquivo `/<installation_directory>/cluster/config.yaml`.

```
management_services:
  platform-security-netpols: enabled
```

Em seguida, continue com a instalação do IBM Cloud Private.

Ativando políticas de rede de segurança de plataforma após a instalação do

IBM Cloud Private

Para ativar as políticas de rede de segurança de plataforma depois de instalar seu cluster, deve-se executar o comando de complemento.

1. Ative `platform-security-netpols` na seção `management_services` no arquivo `/<installation_directory>/cluster/config.yaml`.

   ```
   management_services:
    platform-security-netpols: enabled
   ```

2. Execute o comando de complementos a seguir:

   ```
   docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
   ```

Verificando as políticas de rede de segurança de plataforma em seu cluster

Depois de ativar as políticas de rede de segurança de plataforma, use o comando a seguir para verificar se as políticas estão instaladas. Deve-se instalar o `kubectl` para executar o comando. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

```
kubectl get networkpolicy
```

A saída inclui todas as políticas de rede que estão configuradas em seu cluster.

Desative as políticas de rede de segurança de plataforma no IBM Cloud Private

Para desativar as políticas de rede de segurança de plataforma, execute o comando de complemento.

1. Desative `platform-security-netpols` na seção `management_services` no arquivo `/<installation_directory>/cluster/config.yaml`.

   ```
   management_services:
    platform-security-netpols: disabled
   ```

2. Execute o comando de complementos a seguir:

   ```
   docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
   ```

IBM Cloud Private efetuando login com o IBM Z

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

O componente de criação de log do IBM® Cloud Private nos nós de gerenciamento do IBM® Z é suportado apenas como uma visualização de tecnologia na liberação do 3.2.0. É possível instalar com êxito o componente de criação de log, mas os logs não são coletados e disponibilizados, a menos que você faça mudanças manuais na configuração.

Fazer essas mudanças na configuração desativa uma parte da segurança que protege o acesso à pilha de criação de log. Qualquer carga de trabalho que está instalada no IBM Cloud Private pode enviar dados para o Logstash em vez de restringir o consumo apenas aos logs reais do aplicativo. Uma carga de trabalho maliciosa pode enviar o tráfego que aumenta o carregamento na pilha de criação de log ou, arbitrariamente, introduz logs confusos, inválidos ou maliciosos. Por esses motivos, a criação de log no IBM Z é fornecida apenas como uma visualização de tecnologia. Não use-o em um ambiente de produção.

Ativando a criação de log em nós de gerenciamento do IBM Z

Para ativar o processamento de log com os nós de gerenciamento do IBM Z, deve-se desativar a Segurança da Camada de Transporte (TLS) entre os pods do Filebeat que estão em execução em cada nó e os pods do Logstash que estão recebendo logs

em cada nó de gerenciamento. Você deve editar dois mapas de configuração no IBM Cloud Private para fazer essas mudanças. Atualize os mapas de configuração de uma das maneiras a seguir.

- Insira o comando a seguir a partir da CLI `kubectl`.

```
kubectl -n kube-system edit configmap <name>
```

- No IBM Cloud Private do console de gerenciamento:

  1. Clique em **Configuração** > **ConfigMaps** e selecione **Editar** no menu **mais opções** para o `configmap` desejado.
  2. Modifique o configmap denominado `logging-elk-filebeat-ds-config`.
     - No atributo `data` > `filebeat.yml`, comente os atributos `ssl.*` a seguir, prefixando-os com o caractere `#`:

       ```
       ssl.certificate_authorities
       ssl.certificate
       ssl.key
       ssl.key_passphrase
       ```

  3. Modifique o configmap denominado `logging-elk-logstash-pipeline-config`.
     - No atributo `data` > `k8s.conf`, altere `ssl => true` para `ssl => false`
  4. Exclua os pods de carga de trabalho `logging-elk-logstash` e os pods do conjunto de daemons `logging-elk-filebeat-ds` para forçá-los a reiniciar com a nova configuração.

Esse procedimento também pode ser usado para inverter as mudanças e restaurar os valores originais para os mapas de configuração.

Controlador de política de mutação

O controlador de política de mutação pode ser usado para relatar pods mudados de imagens originalmente digitalizadas. É possível cumprir políticas de mutação com o controlador de política de mutação.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

O controlador de política de mutação se comunica com vários componentes para detectar e corrigir mutações. Há duas interfaces de componente para o controlador de política de mutação:

- Uma interface de componente se comunica com o servidor da API do Kubernetes local para obter a lista de pods mudados e reiniciar os pods mudados.
- A segunda interface do componente recebe o status de mutação de um pod comunicando-se com a API do Mutation Advisor.

Política de mutação

Uma política de mutação é uma instância CustomResourceDefinition (CRD) que contém as especificações das quais os pods devem ser monitorados e qual ação tomar se uma mutação for detectada. Para obter informações adicionais sobre CRDs, consulte Estender a API do Kubernetes com CustomResourceDefinitions.

Visualize a seguinte descrição de exemplo de uma política de mutação e revise as descrições de elementos da política de mutação:

```
Name:         mutation-policy-example
Namespace:    default
Labels:       category=system-and-information-integrity
APIVersion:  mcm.ibm.com/v1alpha1
Kind:         MutationPolicy
Metadata:
  Finalizers:
    finalizer.mcm.ibm.com
Spec:
  Conditions:
    Ownership:
      ReplicaSet
      Deployment
      DeamonSet
      ReplicationController
  NamespaceSelector:
```

```
      Exclude:
        kube-system
      Include:
        default
        kube-*
    labelsSelector:
      env: "production"
    RemediationAction:  enforce
 Status:
   CompliancyDetails:
     Mutation-Policy-Example:
       Default:
         0 mutated pods detected in namespace `default`
       Kube - Public:
         0 mutated pods detected in namespace `kube-public`
   Compliant:  Compliant
  Events:
    Type    Reason         Age                            From                         Message
    ----    ------         ----                           ----                         -------
    Normal  Pod-Restarted  37m                            mutationpolicy-controller  Restarted Pod
default/nginx-7cdbd8cdc9-j8fh9
    Normal  Pod-Restarted  37m                            mutationpolicy-controller  Restarted Pod
kube-public/nginx-7cdbd8cdc9-5k2j4
```

Nota: Em uma política de mutação, o rótulo `category=system-and-information-integrity` categoriza a política e facilita a consulta das políticas de mutação. Se houver um valor diferente para a categoria `key` em sua política de mutação, o valor será substituído pelo controlador de mutação.

Elementos de política de mutação

O `namespaceSelector` define quais namespaces estão sujeitos ao cumprimento da política de mutação. Uma única política de mutação pode ser aplicada a vários namespaces. Por exemplo, se houver duas políticas de mutação com o mesmo valor para o parâmetro `namespaceSelector`, somente a primeira política será aplicada ao namespace especificado.

Visualize o seguinte exemplo de YAML do parâmetro `namespaceSelector` em uma política de mutação que é aplicada a vários namespaces:

```
spec:
  namespaceSelector:
       include: ["default", "kube-*"]
       exclude: ["kube-system"]
```

O `labelsSelector` define qual pod é monitorado pelo controlador de política de mutação. É possível listar rótulos específicos para os seus pods a serem monitorados pelo controlador. Se o parâmetro `labelsSelector` não for especificado, todos os pods nos namespaces especificados serão monitorados pelo controlador de política de mutação.

Nota: O Mutation Advisor varre todos os contêineres no cluster Kubernetes; no entanto, o controlador de política de mutação verifica somente os pods que satisfazem o `namespaceSelector` e o `labelsSelector`.

Visualize o seguinte exemplo de YAML do parâmetro `labelsSelector` em uma política de mutação:

```
labelsSelector:
   env: "production"
```

O parâmetro `conditions` especifica as condições nas quais o controlador de política de mutação é capaz de reiniciar um pod mudado. Um pod deve ser de propriedade de um dos seguintes controladores: ReplicaSet, Deployment, DaemonSet, ReplicationController.

Para reiniciar um pod, deve-se atualizar e editar o valor para o parâmetro `remediationAction` para `enforce`. Visualize o seguinte exemplo de YAML do parâmetro `conditions` em uma política de mutação:

```
remediationAction: enforce
conditions:
    ownership: ["ReplicaSet", "Deployment", "DaemonSet", "ReplicationController"]
```

Nota: Ative o controlador para excluir pods que não são de propriedade de um controlador, atualizando o valor `ownership` para `none`.

Política de pai para a política de mutação

Uma política de mutação pode ser criada diretamente no cluster gerenciado ou no cluster de hub, se ele gerenciar seu cluster gerenciado.

Se o IBM Multicloud Manager não estiver instalado, é possível criar uma política de mutação diretamente em seu cluster gerenciado executando o seguinte comando: `kubectl create -f mutationPolicyFile.yaml`.

Se o IBM Multicloud Manager estiver instalado, crie uma política pai que inclua a política de mutação a ser propagada para o cluster gerenciado.

Visualize a descrição de exemplo da política pai:

```
API Version:  policy.mcm.ibm.com/v1alpha1
Kind:         Policy
Name:         policy-objects
Namespace:    default
Labels:       dev=true
Spec:
  Compliance Type:  musthave
  Namespaces:
    Exclude:
      kube*
    Include:
      default
  Policy-Templates:
    Compliance Type:         musthave
    ObjectDefinition:
    - apiVersion: policies.ibm.com/v1alpha1
      kind: MutationPolicy
      metadata:
        name: mutation-policy-example
        label:
          category: "System-Integrity"
      spec:
        namespaceSelector:
          include: ["default","kube-*"]
          exclude: ["kube-system"]
        labelsSelector:
          env: "production"
        remediationAction: enforce # enforce or inform
        conditions:
          ownership: ["ReplicaSet", "Deployment", "DeamonSet", "ReplicationController"]
        RemediationAction:  enforce
      - apiVersion: policies.ibm.com/v1alpha1
        kind: MutationPolicy
        metadata:
          name: mutation-policy-example2
          ...
Events:
  Type     Reason                                                  Age                 From
Message
  ----     ------                                                  ----                ----
-------
  Normal   policy: default/mutation-policy-example2               2m46s
mutationpolicy-controller  NonCompliant ; 1 mutated pods detected in namespace `system`
  Normal   policy: default/mutation-policy-example                2m5s
mutationpolicy-controller  NonCompliant ; 3 mutated pods detected in namespace `kube-public`; 2
mutated pods detected in namespace `default`
```

O parâmetro `policy-templates` define um conjunto de políticas de mutação a serem propagadas para os clusters gerenciados. As políticas de mutação são criadas pelo controlador de política no cluster gerenciado.

Sinalizações do controlador de mutação

Conforme você instala o controlador de mutação, deve configurar sinalizações do controlador de mutação. Visualize as descrições das sinalizações do controlador de mutação:

- `update-frequency`: Define a frequência de atualizações de consulta do controlador de política de mutação para o Mutation Advisor para novas mutações.
- `watch-ns`: Deve corresponder ao namespace no qual as políticas pai são criadas.

- `parent-event`: Define se os eventos são enviados para as políticas pai no namespace sobre o status das políticas de mutação.

Criando uma política de mutação

Conclua as seguintes etapas para criar uma política de mutação:

1. Crie uma política de mutação. Certifique-se de que o CRD da política de mutação exista.

   O arquivo YAML da política de mutação pode ser semelhante ao seguinte conteúdo:

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
    creationTimestamp: null
    labels:
        controller-tools.k8s.io: "1.0"
    name: mutationpolicies.mcm.ibm.com
spec:
    group: mcm.ibm.com
    names:
    kind: MutationPolicy
    plural: mutationpolicies
    scope: Namespaced
    validation:
    openAPIV3Schema:
        properties:
        apiVersion:
            description: 'APIVersion defines the versioned schema of this representation
                of an object. Servers should convert recognized schemas to the latest
                internal value, and can reject unrecognized values. More info:
https://git.k8s.io/community/contributors/devel/api-conventions.md#resources'
            type: string
        kind:
            description: 'Kind is a string value representing the REST resource this
                object represents. Servers might infer this from the endpoint the client
                submits requests to. Não é possível ser atualizado. In CamelCase. More info:
https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds'
                type: string
        metadata:
            type: object
```

2. Crie uma política de mutação em sua política pai.

   Sua política de mutação pode ser semelhante ao seguinte arquivo YAML:

```
apiVersion: policies.ibm.com/v1alpha1
kind: MutationPolicy
metadata:
  name: mutation-policy-example
  label:
    category: "System-Integrity"
spec:
  namespaceSelector:
    include: ["default","kube-*"]
    exclude: ["kube-system"]
  #labelsSelector:
    #env: "production"
  remediationAction: enforce # enforce or inform
  conditions:
    ownership: ["ReplicaSet", "Deployment", "DaemonSet", "ReplicationController"]
```

3. Para obter uma descrição de sua política de mutação, execute o seguinte comando:

```
kubectl describe mutationpolicies.mcm.ibm.com <myPolicyName>
```

Sua política de mutação é criada.

Serviço de funcionamento do sistema IBM Cloud Private

Serviço de funcionamento do sistema IBM Cloud Private é uma API de REST que fornece o status de seus nós, o servidor de API do Kubernetes, pods com mau funcionamento e os serviços de gerenciamento do IBM Cloud Private e suas dependências.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

O serviço de funcionamento do sistema fornece o status de funcionamento de seu sistema IBM Cloud Private. Visualize a tabela para obter uma descrição dos detalhes do status de funcionamento que são fornecidos com o serviço de funcionamento do sistema:

Status	Descrição de Saída
Nó do cluster IBM Cloud Private	• Fornece detalhes do status de funcionamento do nó do cluster. • Fornece detalhes de falha e eventos do pod para quaisquer nós com mau funcionamento.
Servidor API do Kubernetes	• Fornece detalhes do status de funcionamento do servidor de API do Kubernetes e de outro serviço de gerenciamento.
Pods com mau funcionamento	• Fornece detalhes de falha e eventos do pod no namespace `kube-system`.
Serviço de gerenciamento de cluster do IBM Cloud Private	• Fornece o status de funcionamento de todos os serviços de gerenciamento do IBM Cloud Private. • Fornece a dependência de cada serviço de gerenciamento.

Consulte Componentes do IBM Cloud Private para obter uma descrição dos componentes e suas dependências.

Pré-requisitos

Instale o IBM Cloud Private com o Kubernetes. Para obter mais informações, consulte a seção *Configurações do Kubernetes* na página Customizando o cluster com o arquivo config.yaml.

Ativando o serviço de funcionamento do sistema durante a instalação

É possível customizar o serviço de funcionamento do sistema antes de instalar o IBM Cloud Private.

Edite seu `config.yaml` que está localizado na pasta `/<installation_directory>/cluster` incluindo o conteúdo a seguir. Em seguida, salve e saia do arquivo.

```
management_services:
 system-healthcheck-service: enabled
```

O serviço de funcionamento do sistema é ativado durante a instalação.

Ativando o serviço de funcionamento do sistema após a instalação a partir da

console de gerenciamento

Como um administrador de cluster, conclua as etapas a seguir para ativar o serviço de funcionamento:

1. Efetue login em seu cluster do IBM Cloud Private console de gerenciamento.
2. Instale o gráfico `system-healthcheck-service` clicando em **Catálogo**. Selecione o gráfico `system-healthcheck-service`.
3. Insira um valor para o *Nome da liberação do Helm*.
4. Selecione um namespace a partir do *Menu do namespace de destino*.
5. Clique em **Configurar**
6. Desinstale o `system-healthcheck-service`:

1. No menu de navegação, clique em **Gerenciar** > **Repositórios de Helm**.
2. Clique no ícone **Abrir e fechar opções** para o gráfico `system-healthcheck-service`.
3. Clique em **Excluir**.

Ativando o serviço de funcionamento do sistema após a instalação a partir da interface da linha de

comandos (CLI)

1. Implemente o gráfico `system-healthcheck-service` executando o comando a seguir:

   ```
   helm install system-healthcheck-chart
   ```

2. Instale o gráfico `system-healthcheck-service` executando o comando a seguir:

   ```
   helm install system-healthcheck-service --name <release-name> --namespace kube-system
   ```

3. Para desinstalar o gráfico `system-healthcheck-service`, execute o comando a seguir:

   ```
   helm delete <release-name> --purge --tls
   ```

Detalhes da API do serviço de funcionamento do sistema

É possível acessar as APIs do serviço de funcionamento do sistema.

Obter a API do serviço de funcionamento do sistema

Versão da API
v1alpha1
Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/cluster-health/v1alpha1/health/`

Comando

`GET`

Formato de saída de comando

`application/json`

Execute o comando a seguir para acessar a prontidão da API do serviço de funcionamento do sistema:

```
curl -X GET https://<Cluster Master Host>:<Cluster Master API Port>/cluster-
health/v1alpha1/health -H 'authorization: Bearer 'k8sTokenValue
```

Sua saída pode ser semelhante à saída a seguir:

```
"Cluster health is Healthy"
```

Obter o status do cluster

Versão da API
v1alpha1
Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/cluster-health/v1alpha1/clusterstatus/

Comando

GET

Formato de saída de comando

application/json

Execute o comando a seguir para acessar o status do cluster:

```
   curl -X GET https://<Cluster Master Host>:<Cluster Master API Port>/cluster-
health/v1alpha1/clusterstatus -H 'authorization: Bearer 'k8sTokenValue
```

A saída pode ser semelhante ao conteúdo a seguir:

```
{
     "serviceStatus": {
             "audit-logging": {
                    "status": "Running",
                     "depends": ["k8s"]
             },
             "auth-apikeys": {
                        "status": "NotInstalled",
                        "depends": ["k8s"]
           },
              "auth-idp": {
                        "status": "Running",
                        "depends": ["k8s"]
              },
              "auth-pap": {
                        "status": "Running",
                        "depends": ["k8s"]
           }
      "podFailureStatus": { ...}
      "nodeStatus": { ... }
  }
```

Obter o status do nó

Versão da API

v1alpha1

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/cluster-health/v1alpha1/nodestatus/

Comando

```
GET
```

Formato de saída de comando

```
application/json
```

Execute o comando a seguir para obter o status do nó:

```
curl -X GET https://<Cluster Master Host>:<Cluster Master API Port>/cluster-
health/v1alpha1/nodestatus -H 'authorization: Bearer 'k8sTokenValue
```

A saída pode ser semelhante ao conteúdo a seguir:

```
{
   "nodeStatus": {
           "10.21.3.79": [{
                    "type": "MemoryPressure",
                    "status": "False",
                    "lastHeartbeatTime": "2019-05-03T20:18:45Z",
                    "lastTransitionTime": "2019-04-24T22:04:44Z",
                    "reason": "KubeletHasSufficientMemory",
                    "message": "kubelet has sufficient memory available"
}
```

Problemas Conhecidos

O serviço de funcionamento do sistema não fornece o status de funcionamento dos serviços de gerenciamento do IBM Cloud Private a seguir:

- cert-manager
- platform-pod-security
- unified-router
- platform-security-netpols
- monitoring-crd
- istio

Instalando o IBM Cloud Private com o IBM Cloud Kubernetes Service

Como uma visualização técnica, é possível instalar o IBM® Cloud Private com o IBM Cloud Kubernetes Service.

Importante: Este conteúdo é uma visualização técnica e não deve ser considerado em um ambiente de produção.

- Visão geral
- Pôster de
- Autenticação e Autorização
- Limitações e diferenças de integração

Visão geral

O IBM Cloud Kubernetes Service é um serviço Kubernetes gerenciado. É possível implementar os IBM Certified Containers remotamente em um cluster do IBM Cloud Kubernetes Service usando o IBM Multicloud Manager. Como alguns IBM Certified Containers dependem de serviços específicos do IBM Cloud Private, por exemplo, o IAM e a criação de log, é possível implementar o IBM Cloud Private no IBM Cloud Kubernetes Service para fornecer essas dependências de serviço.

O IBM Cloud Kubernetes Service é um serviço de contêiner gerenciado para a entrega rápida de aplicativos que podem ser ligados a serviços avançados, como o IBM Watson e o IBM Blockchain.

Como um provedor Kubernetes certificado, o IBM Cloud Kubernetes Service fornece os benefícios a seguir:

- planejamento inteligente
- recuperação automática
- Escala horizontal
- descoberta de serviço
- balanceamento de carga

- lançamentos e retrocessos automatizados
- gerenciamento de segredo e configuração
- principal do Kubernetes gerenciado pelo {{site.data.keyword.ibm_notm}}
- atualizações de segurança são aplicadas automaticamente pelo {{site.data.keyword.ibm_notm}}
- nós do trabalhador do Kubernetes estão na conta do cliente
- Atualizações e correções de nó são fornecidas por {site.data.keyword.ibm_notm}}

O serviço Kubernetes também possui recursos avançados em torno de gerenciamento de cluster simplificado, políticas de segurança e de isolamento de contêiner, capacidade de projetar seu próprio cluster e ferramentas operacionais integradas para a consistência na implementação.

Instalação

Instalação

1. Instale o Docker para seu nó de inicialização. O nó de inicialização é o nó que é usado para a instalação de seu cluster. Para obter mais informações, consulte Nó de inicialização.

   É necessária uma versão do Docker que seja suportada por IBM Cloud Private instalada em seu nó de inicialização. Consulte Versões do Docker suportadas. Para instalar o Docker, consulte Instalando manualmente o Docker.

2. Crie um cluster padrão do IBM Cloud Kubernetes Service com um mínimo de três nós do trabalhador. Para um ambiente de produção, 5 nós do trabalhador são recomendados. Todos os nós do trabalhador devem ter um valor de tipo de máquina de pelo menos `b2c.8x32` e uma versão do Kubernetes que corresponda à versão do Kubernetes que é necessária para o IBM Cloud Private. Para obter mais informações, consulte Componentes e a Documentação do IBM Cloud Kubernetes Service sobre a configuração de clusters e de nós do trabalhador.

3. Configure os valores `iks.config.yaml` do instalador do IBM Cloud Private:

   1. Obtenha os detalhes do cluster e o trabalhador para o cluster. Anote o nome do cluster, o valor de Subdomínio do Ingress e os endereços IP Público e Privado do trabalhador.

   2. Configure os valores de cluster do IBM Cloud Kubernetes Service no arquivo `iks.config.yaml` do IBM Cloud Private.

      Notas:

      - Configure `cluster_name` para o nome do cluster do IBM Cloud Kubernetes Service.
      - Configure os valores `cluster_nodes`, `master`, `management` e `proxy` para os valores de endereço IP privado de três dos trabalhadores de cluster do IBM Cloud Kubernetes Service.
      - Configure `storage_class` para `ibmc-file-gold`.
      - Configure `enable_impersonation` como `true`.
      - Configure o valor `iks console host` para o endereço IP público do nó principal.
      - Configure o valor `iks.router.cluster_host` para o valor de Subdomínio do Ingress de clusters do IBM Cloud Kubernetes Service.
      - Configure o valor `iks.router.proxy_host` para o endereço IP público do nó do trabalhador do proxy.

      O código a seguir é um arquivo `iks.config.yaml` de amostra:

```
cluster_name: iks-cluster-name
cluster_nodes:
master:
  - 10.167.6.160
management:
  - 10.167.6.166
proxy:
  - 10.167.6.179

storage_class: "ibmc-file-gold"
enable_impersonation: true

# IKS host and port assignments:
#   console host is the master node's public IP address
#   console port is used to access the ICP console
#   router cluster_host is the IKS cluster Ingress Subdomain
#   router proxy_host is the proxy node's public IP address
#   router kubernetes_ingress_port is the port to used to communicate with Kubernetes
iks:
```

```
console:
  host: x.x.x.x
  port: 8443
router:
  cluster_host: iks.cluster.ingress.subdomain
  proxy_host: y.y.y.y
  kubernetes_ingress_port: 8001

default_admin_password: admin
password_rules:
- '.*'

kubernetes_cluster_type: iks
```

4. Copie a configuração de cluster necessária do IBM Cloud Kubernetes Service para o instalador do IBM Cloud Private:

    1. Obtenha a configuração de cluster a partir do cluster do IBM Cloud Kubernetes Service e configure a variável de ambiente KUBECONFIG. Para obter mais informações, consulte Etapa 6 da documentação do IBM Cloud Kubernetes Service. Configurar o cluster que você criou como o contexto para esta sessão.
    2. Certifique-se de que a variável de ambiente KUBECONFIG esteja configurada e, em seguida, execute os comandos a seguir em uma janela do terminal a partir da pasta do instalador do IBM Cloud Private para copiar os valores de configuração de cluster necessários para o diretório do cluster do instalador:

    ```
    cp  ${KUBECONFIG%/*}/*.yml cluster/kubeconfig
    cp  ${KUBECONFIG%/*}/*.pem cluster/
    ```

5. Execute o instalador do IBM Cloud Private para instalar o IBM Cloud Private no IBM Cloud Kubernetes Service:

    1. Anexe os valores em `iks.config.yaml` ao término do arquivo `config.yaml` do instalador do IBM Cloud Private.

    2. Em uma janela do terminal, execute o instalador do IBM Cloud Private, especificando para usar a lista de execução `install-with-iks`.

        Por exemplo:

        ```
        docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer ibmcom/icp-
        inception:3.2.0 install-with-iks -v | tee intall.log
        ```

        Nota: a imagem do Docker de incepção do IBM Cloud Private é diferente.

Desinstalação

Para desinstalar o IBM Cloud Private com o IBM Cloud Kubernetes Service, execute o comando a seguir:

```
docker run --net=host -t -e LICENSE=accept \
-v /path/to/cluster:/installer/cluster ibmcom/icp-inception:3.2.0 uninstall-with-iks -v
```

Autenticação e Autorização

A autenticação é a mesma que uma instalação padrão do IBM Cloud Private. A experiência de login para a console de gerenciamento e a CLI (`cloudctl`) é a mesma. O LDAP é o mesmo e usuários e grupos de usuários são importados em uma equipe do IBM Cloud Private. Os usuários são designados a uma função durante a importação e, quando os usuários efetuam login na console de gerenciamento e no `cloudctl`, um `accesstoken` e um `id_token` são gerados.

A autorização é baseada nas equipes e nos recursos que você incluiu. Durante a configuração, é possível incluir um recurso de namespace em uma equipe. A autorização é feita com base na função do usuário que está agindo em um namespace.

Limitações e diferenças de integração

As seguintes limitações e diferenças existem para o IBM Cloud Private com o IBM Cloud Kubernetes Service:

Docker

O IBM Cloud Kubernetes Service usa containerd como seu tempo de execução de contêiner. O IBM Cloud Private usa o Docker, no entanto, ele suporta o containerd como uma visualização de tecnologia.

Serviços

Visualize a tabela a seguir dos serviços de gerenciamento do IBM Cloud Private e suas dependências que são suportados no IBM Cloud Private com o Kubernetes Service (IKS):

Tabela 1. IBM Cloud Private com as dependências do serviço de gerenciamento do IBM Cloud Kubernetes Service

Serviço de gerenciamento	Dependências
image-manager	
cert-manager	
mongodb	
monitoring-crd	
auth-idp	mongodb
auth-apikeys	mongodb
auth-pap	mongodb
auth-pdp	mongodb, auth-idp, auth-pap, auth-apikeys
catalog-ui	auth-idp, platform-api, helm-api, helm-repo, multicluster-hub
custom-metrics-adapter	monitoring
helm-api	mongodb, platform-api, icp-management-ingress, helm-repo, mgmt-repo
helm-repo	mongodb
icp-management-ingress	
logging	IAM
metering	mongodb, IAM
metrics-server	
nginx-ingress	
mgmt-repo	mongodb
monitoring	IAM
platform-api	IAM
platform-ui	auth-idp, platform-api, catalog-ui, image-manager
secret-watcher	
security-onboarding	IAM
web-terminal	platform-api, IAM

Nota: o Identity and Access Management (IAM) inclui os serviços a seguir: `auth-idp`, `auth-pap`, `auth-pdp`, `auth-apikeys` e `secret-watcher`.

Os serviços a seguir não são suportados no IBM Cloud Private com o IKS: Vulnerability Advisor, Mutation Advisor e registro de imagem local. Consulte Ativando e desativando os serviços de gerenciamento do IBM Cloud Private para obter mais informações sobre as dependências para os serviços de gerenciamento.

Nota: é possível usar o IBM Cloud Container Registry, se necessário. Para obter mais informações, consulte o Tutorial de introdução do IBM Cloud Container Registry.

Aumente e reduza os nós do trabalhador do IBM Cloud Private:

Siga as instruções do IBM Public Cloud Kubernetes Service para incluir e remover nós do trabalhador e zonas. É possível incluir ou remover manualmente os nós do trabalhador usando a console de gerenciamento ou a CLI. É possível incluir ou remover nós do trabalhador em uma ou múltiplas zonas existentes. É possível redimensionar um conjunto de trabalhadores existente ou criar um novo conjunto de trabalhadores.

Nota: deve-se ter as permissões do IBM Cloud Kubernetes Service necessárias.

Para obter mais informações, consulte a Documentação do IBM Cloud Kubernetes Service sobre como incluir nós do trabalhador manualmente.

Memória

As opções de armazenamento do IBM Public Cloud Kubernetes Service são suportadas. O armazenamento de arquivo está sempre disponível para seu cluster do IBM Cloud Private. É possível usar gráficos do Helm para implementar opcionalmente o armazenamento de Bloco e de objeto ou ambos.

Para obter mais informações, consulte os tópicos do IBM Cloud Kubernetes Service a seguir:

- Armazenando dados no IBM File Storage for IBM Cloud

- Armazenando dados no IBM Block Storage for IBM Cloud
- Armazenando dados no IBM Cloud Object Storage

Ingresso

Por padrão, o cluster do IBM Cloud Private obtém o controlador de ingresso do IBM Public Cloud Kubernetes Service (balanceador de carga avançado). Siga estas instruções se você deseja substituir o balanceador de carga avançado pelo controlador de ingresso nginx do IBM Cloud Private para suas cargas de trabalho: Documentação do IBM Cloud Kubernetes Service sobre como trazer seu próprio controlador do Ingress.

Nós do cluster

Os nós do IBM Cloud Private são configurados como nós do trabalhador de cluster do IBM Cloud Kubernetes Service na infraestrutura do IBM Public Cloud. É possível encontrar algumas diferenças devido à segurança extra. Por exemplo, `ssh` é desativado em todos os nós. Para obter mais informações, consulte Documentação do IBM Cloud Kubernetes Service sobre nós do trabalhador.

Instalando o Knative no IBM Cloud Private

O Knative fornece um conjunto de componentes de middleware que são essenciais para construir aplicativos modernos, centrados na origem e baseados em contêineres que podem ser executados em qualquer lugar no local, na nuvem ou mesmo em um data center de terceiros.

Limitação: o Knative não suporta a especificação de um namespace para instalação. Antes de instalar o Knative, certifique-se de instalar o istio em seu cluster do IBM Cloud Private.

- Instalando o Knative para um cluster existente
- Instalando o knative usando a CLI
- Verificando a instalação
- Suporte para knative para Linux® on Power® (ppc64le)

Instalando o Knative em um cluster existente

Nota: um cluster do IBM Cloud Private 3.2.0 suporta o gráfico `knative` versão 0.2.x e 0.1.x. Os gráficos `knative` estão disponíveis no repositório `ibm-charts`: `https://github.com/IBM/charts/tree/master/community/knative`.

Será possível implementar o Knative se você já tiver um cluster do IBM Cloud Private 3.2.0 instalado. O gráfico Knative atual deve usar a CLI para instalar os crds knative primeiro. Use o comando a seguir:

```
kubectl apply -f https://raw.githubusercontent.com/IBM/charts/master/community/knative/all-crds.yaml
```

Para instalar um gráfico Knative a partir da console de gerenciamento do IBM Cloud Private, clique em **Catálogo** e procure pelo gráfico `knative`. Escolha o gráfico Knative que você deseja instalar.

Instalando o Knative usando a CLI

Instale os crds de knative com o seguinte comando:

```
kubectl apply -f https://raw.githubusercontent.com/IBM/charts/master/community/knative/all-crds.yaml
```

Instale o gráfico usando a CLI do Helm:

```
helm repo add ibm-community-charts
https://raw.githubusercontent.com/IBM/charts/master/repo/community
helm install ibm-community-charts/knative --name knative [--tls]
```

O comando implementa o knative em um cluster do IBM Cloud Private 3.2.0 na configuração padrão. A seção configuração lista os parâmetros que podem ser configurados durante a instalação.

Verificando a instalação

Após a conclusão da instalação, verifique se todos os componentes que você ativou para Knative estão criados e em execução. Monitore os componentes Knative até que todos os componentes mostrem um status de `Running`. Por exemplo:

```
kubectl get pod -n knative-serving
NAME                          READY     STATUS    RESTARTS   AGE
activator-74bc454c4b-tcpqs    2/2       Running   0          4m
autoscaler-8bd664478-pqhph    2/2       Running   0          4m
controller-7cbd5bdc88-9z6dq   1/1       Running   0          4m
webhook-7bcff85bf9-vz9hm      1/1       Running   0          3m

kubectl get pod -n knative-build
NAME                                READY     STATUS    RESTARTS   AGE
build-controller-d9584dcd6-hpb7b    1/1       Running   0          10m
build-webhook-5bfdbd4fb7-nn79w      1/1       Running   0          10m
```

Se você ativar os parâmetros `eventing`, `monitoring` e `eventingSources`, é possível visualizar as informações de status. Por exemplo:

```
kubectl get pod -n knative-eventing
NAME                                          READY     STATUS    RESTARTS   AGE
eventing-controller-6f8f5698ff-mrvbq          2/2       Running   0          12m
in-memory-channel-controller-787865b86d-w781v 2/2       Running   1          12m
in-memory-channel-dispatcher-78bfc7d88f-864wc 2/2       Running   1          12m
webhook-75dcb58956-5smqc                      1/1       Running   0          12m

kubectl get pod -n knative-monitoring
NAME                                 READY     STATUS    RESTARTS   AGE
elasticsearch-logging-0              1/1       Running   0          12m
elasticsearch-logging-1              1/1       Running   0          7m
grafana-744b8d4ccb-vsskd             1/1       Running   0          12m
kibana-logging-7d8bd66996-mx7rm      1/1       Running   0          13m
kube-state-metrics-68cd885bf7-sc7zs  4/4       Running   0          9m
node-exporter-pgv8h                  2/2       Running   0          13m
node-exporter-q8txs                  2/2       Running   0          13m
prometheus-system-0                  1/1       Running   0          12m
prometheus-system-1                  1/1       Running   0          12m

kubectl get pod -n knative-sources
NAME                  READY     STATUS    RESTARTS   AGE
controller-manager-0  1/1       Running   0          13m
```

Suporte Knative para Linux on Power (ppc64le)

Uma instalação knative em um cluster do IBM Cloud Private 3.2.0 que está em execução em uma máquina do Linux on Power (ppc64le) requer as seguintes configurações:

```
build:
  buildController:
    image: ibmcom/knative-build-cmd-controller:0.5
  buildWebhook:
    image: ibmcom/knative-build-cmd-webhook:0.5
  credsInit:
    image: ibmcom/knative-build-cmd-creds-init:0.5
  gcsFetcher:
    image: ibmcom/gcs-fetche:0.5
  gitInit:
    image: ibmcom/knative-build-cmd-git-init:0.5
  nop:
    image: ibmcom/knative-build-cmd-nop:0.5
eventing:
  enabled: true
  eventingController:
    image: ibmcom/knative-eventing-cmd-controller:0.5
  inMemoryProvisioner:
    enabled: true
    inMemoryChannelController:
      controller:
        image: ibmcom/knative-eventing-pkg-provisioners-inmemory-controller:0.5
    inMemoryChannelDispatcher:
      dispatcher:
        image: ibmcom/knative-eventing-cmd-fanoutsidecar:0.5
  webhook:
    image: ibmcom/knative-eventing-cmd-webhook:0.5
eventingSources:
  enabled: true
  controllerManager:
    manager:
```

```
      image: ibmcom/knative-eventing-sources-cmd-manager:0.5
serving:
  activator:
    image: ibmcom/knative-serving-cmd-activator:0.5.2
  autoscaler:
    image: ibmcom/knative-serving-cmd-autoscaler:0.5.2
  controller:
    image: ibmcom/knative-serving-cmd-controller:0.5.2
  queueProxy:
    image: ibmcom/knative-serving-cmd-queue:0.5.2
  webhook:
    image: ibmcom/knative-serving-cmd-webhook:0.5.2
```

Salve o conteúdo de configuração em `power-values.yaml`. Instale os crds Knative com o comando a seguir:

```
kubectl apply -f https://raw.githubusercontent.com/IBM/charts/master/community/knative/all-crds.yaml
```

Instale o gráfico usando a CLI do Helm com a opção `-f power-values.yaml`

```
helm repo add ibm-community-charts
https://raw.githubusercontent.com/IBM/charts/master/repo/community
helm install ibm-community-charts/knative --name knative -f power-values.yaml [--tls]
```

Limitação: apenas a imagem em `power-values.yaml` suporta uma máquina Linux on Power (ppc64le).

Resolução de Problemas e Suporte

Saiba como isolar e resolver problemas com o IBM® Cloud Private.

Verifique se seus problemas não estão relacionados a requisitos do sistema operacional, como disco, memória e capacidades de CPU. Para obter mais informações sobre os requisitos do sistema, consulte Requisitos do sistema.

- Suporte
- Problemas relatados corrigidos
- Instalação e upgrade
- Login
- Segurança
- LDAP
- console de Gerenciamento
- Redes
- Armazenamento
- Eventos, logs e códigos de erro

Suporte

Caso precise de ajuda para usar o IBM® Cloud Private e o IBM Multicloud Manager, visite a Página de suporte. Além disso, é possível ver se o problema está documentado em Problemas conhecidos e limitações ou em Resolução de problemas.

Para obter informações sobre o IBM Multicloud Manager, consulte Resolução de problemas do IBM Multicloud Manager e Limitações e problemas conhecidos do IBM Multicloud Manager.

Saiba mais sobre o suporte a partir dos tópicos a seguir:

- Tipos de suporte do IBM Cloud Private
- Suporte de software livre no IBM Cloud Private
- MustGather para coletar logs e obter suporte

Tipos de suporte do IBM Cloud Private

A IBM oferece diferentes níveis de suporte para as diferentes ofertas do IBM Cloud Private e, se mais suporte é necessário, é possível comprar planos de suporte.

A IBM oferece planos de suporte de alta qualidade para os diferentes pacotes configuráveis do produto corporativo. O acesso à grande equipe de suporte da IBM está incluído com a compra do IBM Cloud Private (S &S do Passport Advantage). É possível

acessar a equipe de Suporte IBM por meio da página de suporte

Se você comprou um pacote configurável do produto corporativo, também será possível comprar o Suporte Premium para o IBM Cloud Private. Esse plano oferece responsividade mais rápida, maior prioridade para casos de suporte e um gerente de sucesso do cliente nomeado para ajudá-lo a se tornar mais bem-sucedido. Entre em contato com o departamento de vendas do IBM Cloud Private para saber mais sobre o Suporte Premium do IBM Cloud Private.

A IBM também pode fornecer opções adicionais de suporte ao software livre para clientes que precisam de mais assistência. Entre em contato com Vendas IBM ou com o fornecedor de software livre para obter mais informações sobre esses serviços.

O IBM Cloud Private-CE (Community Edition) oferece um plano de suporte digital que inclui:

- A Comunidade Slack pública do IBM Cloud Private-CE (Community Edition). Inscreva-se ou efetue logon.
- Stack Overflow. Visualize perguntas do Stack Overflow com a tag ibm-cloud-private.
- O Watson Chatbot na página de suporte do IBM Cloud Private. Tente o robô de bate-papo sem precisar efetuar login.

Suporte de software livre no IBM Cloud Private

Como o IBM® Cloud Private contém software livre, a IBM oferece algum suporte para o software livre que ele utiliza.

A IBM usa software livre de três maneiras:

- Para ajudar seus clientes avaliados a evitar o bloqueio de fornecedores
- Para assegurar a compatibilidade com o maior ecossistema de software
- Para acelerar o desenvolvimento de seus produtos

Os componentes de software livre de nível corporativo maduros que são usados no IBM Cloud Private foram cuidadosamente selecionados, integrados e testados para funcionar conforme descrito na documentação e nas licenças do produto. Os componentes incluem componentes de software livre integrantes como Kubernetes, Docker Engine, Helm, pilha ELK e Cloud Foundry.

O IBM® Software Support Handbook é a referência definitiva para instrução de suporte de software livre da IBM. Para revisar a instrução de suporte de software livre da IBM, consulte Software de terceiros e software Open Source no IBM Software Support Handbook. A equipe de suporte para IBM Cloud Private usa as informações nesta instrução para definir o escopo de suas interações com o cliente.

O principal objetivo da equipe de suporte do IBM Cloud Private é ajudar os clientes pagantes a ter seus produtos suportados funcionando novamente o mais rápido possível. Embora as opções de suporte à comunidade estejam disponíveis para todos os usuários do IBM Cloud Private, entrar em contato com o Suporte IBM é a melhor opção de suporte para usuários das diferentes edições corporativas de pacote configurável do IBM Cloud Private. Os canais de suporte à comunidade são as únicas opções de suporte para os usuários do IBM Cloud Private-CE (Community Edition). Consulte tipos de suporte do produto para obter mais informações.

Ao entrar em contato com o suporte IBM, existem duas fases de suporte livre:

1. A fase de determinação de problema. Durante essa fase, os engenheiros de suporte determinam se a causa raiz de um problema é devido ao código de software livre ou código IBM.
2. A fase de suporte ao software livre. Se a causa raiz é devido ao código de software livre, engenheiros de suporte identificam qual projeto de software livre contém o problema e trabalham com o cliente para identificar as próximas etapas.

Se um defeito crítico é localizado no software livre no IBM Cloud Private, engenheiros da IBM utilizam recursos da comunidade de software livre para entregar uma correção para o problema. A correção resultante é integrada, testada e liberada para os clientes como uma atualização de emergência e na próxima liberação do produto. Se nenhuma correção estiver disponível na comunidade para um defeito crítico, a IBM poderá usar esforços comerciais razoáveis para fornecer uma correção de teste para os clientes e, então, trabalhar com a comunidade de software livre específica para criar uma correção oficial. O árbitro final sobre se uma correção suportada pode ser fornecida pertence à comunidade de software livre.

Observe que, se qualquer projeto de software livre de nível corporativo tiver um defeito crítico, muitos clientes serão afetados e a comunidade trabalhará o mais rápido possível para resolver isso. É incomum que apenas um cliente encontre um defeito crítico de software livre.

A IBM não pode suportar nenhum software livre que não seja fornecido com o IBM Cloud Private. Além disso, se um cliente optar por fazer qualquer adição, subtração ou upgrade no código de software livre do IBM Cloud Private fora das atualizações do produto, esse código não será suportado. Para o Cloud Foundry, a IBM pode ajudá-lo a relatar defeitos nos buildpacks e tempos de

execução da comunidade para a comunidade de software livre apropriada. A IBM não pode fornecer correções para esses buildpacks e tempos de execução da comunidade do Cloud Foundry.

Nota: o IBM® Cloud Private suportará novas versões dos sistemas operacionais suportados, o Kubernetes, o Docker e outras infraestruturas dependentes quando novas liberações acontecerem e quando estiverem totalmente testadas pela equipe do IBM® Cloud Private.

Se você precisar de mais suporte, a IBM poderá ajudar. Consulte tipos de suporte do produto na documentação do produto para obter mais informações sobre o Suporte Premium do IBM Cloud Private.

MustGather para coletar logs e obter suporte

A intenção desse documento é orientar a reunião de informações de acordo com o problema que está ocorrendo, antes de abrir um caso com o Suporte IBM®. Conclua as seguintes tarefas para ajudá-lo a familiarizar-se com o processo de resolução de problemas.

Nota: Uma ferramenta `healthcheck` coleta os dados necessários para abrir o caso de suporte no portal de suporte IBM.

Auto-diagnóstico

Antes de entrar em contato com o Suporte IBM, consulte Estouro de pilha ou Resolução de problemas e suporte do IBM® Cloud Private para verificar se seu problema foi relatado.

Para visualizar exemplos e tutoriais específicos para tópicos do IBM Cloud Private, consulte os seguintes links:

- Explore outros tópicos na Seção de resolução de problemas do Knowledge Center do IBM.
- Explore a Comunidade de Desenvolvedores do IBM Cloud Private.

Informações gerais sobre resolução de problemas

Ao abrir um chamado para suporte, siga as instruções do Suporte IBM. À medida que você reúne informações para abrir um chamado para suporte, é necessário incluir as seguintes informações:

- Title
- Versão de produto
- Plataforma (Linux® on Power® (ppc64le), Linux® ou {{site.data.keyword.s390}})
- Sistema operacional *S.O.* ({{site.data.keyword.rhel_short}}, Ubuntu, SUSE Linux Enterprise Server)
- Plataforma de virtualização (VMWARE, Azure, AWS, IBM Cloud)
- Alta Disponibilidade *HA*
- Área do problema
- Severidade
- Descrição detalhada do erro
- Arquivo `hosts`
- Arquivo `config.yaml`

- resultados de verificação de funcionamento

  Nota: Os arquivos `hosts` e `config.yaml` estão localizados no diretório `/<installation-directory>/cluster`. O arquivo `hosts` fornece detalhes da topologia do servidor para o IBM. O arquivo `config.yaml` fornece detalhes de customização, que também incluem detalhes do balanceador de carga.

Tipos de problemas

- Instalação
- Upgrade
- Desinstalação
- Reinstalação
- Segurança (LDAP, Alta Disponibilidade (HA), URL de Acesso)
- Uso da interface da linha de comandos (CLI) (Helm, Kubernetes, IBM Cloud Private, Docker)
- Implementação de pods
- Travamento de pods
- Interrupção de servidor
- Servidor não inicia
- Problema de memória

- Desempenho

Instalação

Conforme você instala o IBM Cloud Private, assegure-se de que todos os requisitos do sistema tenham sido atendidos e conclua todos os pré-requisitos. Para obter mais informações, consulte Requisitos do sistema IBM Cloud Private e Instalação.

Nota: verifique se o Docker está instalado com os binários do IBM ou diretamente do Docker.

Se você tiver problemas de instalação, visualize seus logs de instalação do cluster para visualizar o resumo. Execute o comando a seguir:

```
  docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/<installation-directory/cluster/logs
ibmcom/icp-inception-<architecture>:<version> -vvv
```

Upgrade

Se você tiver problemas de upgrade, conclua as etapas a seguir:

1 . Se você fizer o upgrade de seu cluster a partir de uma versão anterior, execute o comando a seguir:

```
docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/<installation-directory>/cluster/upgrade-
version
```

1. À medida que você atualiza seu cluster, verifique o número da versão em seu arquivo `tar.gz`. Execute o comando a seguir:

   ```
   docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/<installation-
   directory>/cluster/images ibmcom/icp-inception-<architecture>:<version> -vvv
   ```

2. Depois de fazer upgrade da versão, visualize os logs de instalação do cluster para verificar se FixPacks estão instalados. Execute o comando a seguir:

   ```
   docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/<installation-directory>/cluster/logs
   -vvv
   ```

   Nota: é possível verificar se quaisquer FixPacks estão instalados a partir do console de gerenciamento. Efetue login no cluster do IBM Cloud Private e, em seguida, clique em **Sobre**.

Desinstalação

Se você tiver problemas com a desinstalação do IBM Cloud Private, conclua as etapas a seguir:

1 . Visualize o diretório no qual o IBM Cloud Private está instalado. Execute o comando a seguir:

```
docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/<installation-directory/cluster/logs
ibmcom/icp-inception-<architecture>:<version> -vvv
```

1. Execute novamente o comando de desinstalação:

   ```
   sudo docker run -e LICENSE=accept --net=host \
   -t -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):
   <version>-ee uninstall -vvv
   ```

2. Crie uma listagem de arquivos do diretório no qual IBM Cloud Private está instalado e execute o comando a seguir:

   ```
   ls -ltR > filelist.txt
   ```

Reinstalação

Se você tiver problemas com a reinstalação de seu cluster do IBM Cloud Private, conclua as etapas a seguir para reinstalar seu cluster:

1 . Desinstale seu cluster executando o comando a seguir:

```
sudo docker run -e LICENSE=accept --net=host \
-t -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):
```

```
<version>-ee uninstall -vvv
```

1. Crie uma listagem de arquivos do diretório no qual IBM Cloud Private está instalado e execute o comando a seguir:

   ```
   ls -ltR > filelist.txt
   ```

2. Remova os dados do Docker de seu diretório executando o comando a seguir:

   ```
   rm -rf /var/lib/docker
   ```

3. Reinstale seu cluster executando o comando a seguir:

   ```
   docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/<installation-directory/cluster/logs
   ibmcom/icp-inception-<architecture>:<version> -vvv
   ```

Segurança (LDAP, HA, URL de Acesso)

Importante: Verifique seu acesso de função. Para obter mais informações, consulte Controle de acesso baseado na função.

Visualize as perguntas a seguir para ajudá-lo a reunir as informações a seguir:

- Qual é a função do usuário que está com login efetuado?
- Qual diretório LDAP você está usando?
- Você configurou o LDAP sobre SSL?
- Qual é sua URL LDAP?
- O LDAPSearch está instalado?

Se você tiver problemas de segurança, conclua as etapas a seguir:

1 . Verifique se a procura LDAP está instalada executando o comando a seguir:

```
```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub
```

Se a procura LDAP não estiver instalada, execute o comando a seguir para instalar a procura LDAP:

* Para Ubuntu:
  ```
  apt install ldap-utils
  ```

* Para Red Hat Enterprise Linux (RHEL):
  ```
  yum install openldap-clients
  ```

1. Visualize o status do pod `auth` executando o comando a seguir:

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

2. Para visualizar os logs a partir de seu pod `auth`, execute o comando a seguir:

   ```
   kubectl -n kube-system logs auth-yours
   ```

3. Visualize os detalhes sobre a procura LDAP executando o comando a seguir:

   ```
   ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -
   s sub
   ```

Uso da linha de comandos (CLI) (Helm, Kubernetes, IBM Cloud Private, Docker)

Visualize as perguntas a seguir para ajudar a reunir informações para quaisquer problemas de CLI:

- Qual linha de comando você está usando?
- Qual é a função do usuário que está executando o comando?
- O problema ocorreu como um usuário autenticado LDAP ou administrador de cluster? Se for um usuário autenticado LDAP, ele é um administrador no namespace?
- Em qual conta ocorreu o problema com o uso da CLI?
- Qual é a saída do comando?
   ```

- Esse comando estava funcionando e foi interrompido ou é a primeira vez que você está tentando executá-lo?

Conclua as etapas a seguir para coletar os logs para problemas da CLI do Helm, Kubernetes, IBM Cloud Privatee do Docker:

- Para Helm:
  - Verifique a versão da CLI do Helm executando o comando a seguir:

    ```
    helm version --tls
    ```

  - Depure o `helm` executando o comando a seguir:

    ```
    helm <command> --tls --debug
    ```

- Para Kubernetes:
  - Verifique a versão da CLI do Kubernetes executando o comando a seguir:

    ```
    kubectl version
    ```

  - Depure o `kubectl` executando o comando a seguir:

    ```
    kubectl version --v=<version_number>
    ```

- Para a CLI do IBM Cloud Private:
  - Verifique a CLI do IBM Cloud Private executando o comando a seguir:

    ```
    cloudctl version
    ```

- Para a CLI do Docker:
  - Verifique o número da versão para o Docker executando o comando a seguir:

    ```
    docker info
    ```

Implementando pods

Conclua as etapas a seguir para reunir logs para problemas de implementação do pod:

1 . Para obter uma lista de seus pods, execute o comando a seguir:

```
kubectl get pods --all-namespaces -o wide
```

1. Obtenha uma descrição de seus pods executando o comando a seguir:

   ```
   kubectl describe pods --all-namespaces
   ```

Travamento de pods

Se seus pods estiverem travando, identifique o pod executando o comando a seguir:

```
kubectl get pods --all-namespaces
```

Interrupção de servidor

Se estiver ocorrendo interrupção do servidor, colete os logs para descrever o problema e o local do servidor interrompido. Certifique-se de incluir quais etapas foram seguidas e a frequência do erro.

Por exemplo, seu log pode ter as informações a seguir:

- Uma ou mais mensagens de erro
- Dump de memória de travamento do sistema
- Captura de tela da saída a partir do comando `top` (se possível)

Servidor não inicia

Se seu ambiente do IBM Cloud Private não for iniciado, conclua as seguintes etapas para coletar os logs:

1 . Identifique qual servidor não está iniciando (principal, proxy, gerenciamento, trabalhador).

1. Verifique o status de seu servidor `kubelet` executando o comando a seguir:

   ```
   systemctl status kubelet
   ```

2. Para verificar o status de seu servidor `docker`, execute o comando a seguir:

   ```
   systemctl status docker
   ```

3. Para obter os logs do `kubelet`, execute o comando a seguir:

   ```
   journalctl -u kubelet
   ```

4. Para obter os logs do `docker`, execute o comando a seguir:

   ```
   journalctl -u docker
   ```

5. Verifique os serviços do Kubernetes a partir do nó principal executando o seguinte comando:

   ```
   docker ps -a | grep hyper
   ```

6. Verifique os contêineres em execução no Docker executando o seguinte comando:

   ```
   docker ps | grep ibmcom | grep k8s
   ```

Problema de memória

Se você ficar sem memória, use a ferramenta `healthcheck`. Execute o comando a seguir:

```
  sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster ibmcom/icp-
inception-<architecture>:<version> healthcheck -v
```

Desempenho

Se o ambiente do IBM Cloud Private estiver lento, reúna as seguintes informações para cada nó para abrir uma solicitação de serviço para o Suporte IBM:

- Uso de Memória
- Memória Disponível
- uso de CPU
- Saída de `top`
- Synthetic Apeture Radar (SAR) (se disponível)
- nmon (se disponível) (monitor que obtém o usuário do recurso do servidor)

Coletando logs

Na pasta `/<installation-directory>/cluster/`, use a seguinte ferramenta `healthcheck` para coletar logs. Ao usar a ferramenta `healthcheck`, é criado um arquivo para coletar as informações necessárias para abrir um caso com o Suporte IBM®.

Crie uma pasta `healthcheck` para seus logs, execute o seguinte comando:

```
  sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster ibmcom/icp-
inception-<architecture>:<version> healthcheck -v
```

É possível verificar o nome da imagem executando o seguinte comando:

```
  docker images | grep inception
```

Nota: Deve-se arquivar toda a pasta healthcheck. Antes de arquivar a pasta, verifique os pods e as pastas do sistema e se cada pasta está preenchida com arquivos de log na pasta `healthcheck`.

Problemas relatados corrigidos

Revise a lista de problemas corrigidos para ver se o problema relatado foi corrigido na liberação.

Problema	Descrição
21259	Como implementar uma liberação do Helm sem mudar manualmente o repositório de imagem

Problema	Descrição
21733	O terminal da web não funciona
25482	IBM Cloud Private - problema do terminal da web
21187	O Installer não faz upload da regra de senha do administrador padrão para o serviço de API do ICP
23733	Nós do trabalhador ainda são exibidos por meio do comando cloudctl depois de removê-los
21703	Falha de instalação off-line do CF 3.1.2: não é possível localizar a imagem cfp-config-manager-3.1.2-024
21044	O cliente precisa de uma correção ou etapas para atualizar o TLS 1.2 para a porta 443 (ingresso)
19766	Vulnerabilidades baixas de SSL ainda são mostradaqqas após o upgrade da 2.1.0.3 para a 3.1
23949	A versão do servidor, openresty/1.13.6.2, foi divulgada no cabeçalho de resposta do servidor HTTP.
19088	A vulnerabilidade é 42873 - Conjuntos de cifras de intensidade média SSL suportados
17024	O serviço Kibana está em status vermelho: configuração: Erro 503 Serviço Indisponível
24087	"Erro Interno do Servidor" ao tentar visualizar o log de auditoria no Kibana utilizando

um usuário que possui a função de Auditor | |23975 | O usuário Auditor pode ver os logs do aplicativo na descoberta do Kibana | |20773 | o nome de domínio do cluster que começa com "svc" está quebrando a instalação do mongodb | |24305 | Renderização direta do Grafana: erro: "A inicialização de modelagem falhou: Não autorizado" | |22673 | ICP Mongodb no estado PodInitializing | |20292 | O volume ou a taxa do Log de Auditoria está fazendo com que o ELK se torne instável - o cliente deseja que a alimentação de Logs de auditoria deseja desativada | |18073 | Instalando o serviço Principal: a correção do Mongodb para clusters do IBM Cloud Private versão 2.1.0.3 quebra o helm-api | |22130 | O monitoring-prometheus falha ao iniciar com um erro - "Falha ao abrir o diretório do BD de armazenamento de blocos: recurso temporariamente indisponível" | |23037 | Não há controle de autoridade na criação de log e no monitoramento quando alternado para eles a partir do console do ICP. | |18989 | O carregamento do gráfico cloudctl falha ocasionalmente | |19475 | EVRY: ICP 311: um usuário que está restrito a um determinado namespace não pode executar o comando |23061 | Direitos de recursos do gráfico/repositório do Helm | |25319 | Como restaurar o repositório local| |21408 | Gráfico do Helm ibm-mariadb-chart quebrado para a plataforma PPC na 3.1.1 | |20582 | Problemas para aplicar algumas correções do ICP 3.1.1 | |24890 | O `skip_pre_check` não ignora realmente a verificação do cluster_CA_domain | |21841 | 310->312 O endereço do balanceador de carga deve ser igual ao domínio de CA do cluster, | |21832 | pré-verificação do status do cluster antes do upgrade | |24067 | Faça upgrade para mandatos da 3.1.2 correspondentes ao cluster_CA_domain e ao cluster_lb_address | |22726 | O contêiner istio-proxy mostra `exec format error` no sistema Power | |23507 | A UI de conformidade mostra uma janela completamente em branco | |23266 | MCM 3.1.2. Consumo de memória do pod MongoDB | |24297 | O cliente precisa restringir os endereços IP de origem que podem acessar o ICP | |22811 | CVE-2019-1002100 | |18941 | Etapas de detalhe para backup/restauração no ICP CNE 3.1.x | |23586 | As mensagens de erro sobre o mariadb ocorreram repetidamente `Error: 105: Key already exists (/mariadb_lock)` | |19029 | EVRY:Alto uso de CPU nos Principais no ICP 311 de múltiplos principais | |21858 | ICP 2.1.0.3 - falha ao ativar correção temporária: icp-2.1.0.3-build502221 | |23721 | ICP 3.1.1 - coleta de Lixo com falha | |20719 | Uma correção icp-2.1.0.3-build510945 aplicável é necessária para a plataforma amd64 | |23672 | Autoridade de referência da imagem do Docker a partir do painel | |21368 | Unisys 2.1.0.3 Deployments Maxing out Workers, Nodes estão com mau funcionamento | |14141 | Atualize o ICP 2.1.0.3 para incluir uma correção crítica do Kubernetes disponível na v1.10.5 | |25394 | O /var/lib/calico/nodename deve ser removido ao remover um nó | |23438 | ICP4D: falha ao instalar o ICP for Data v1.2.1 no RHEL7.5 VM (Softlayer). | |23645 | Não é possível incluir recursos adicionais na equipe/os recursos incluídos anteriormente também são perdidos | |21856 | A página de visão geral do Container NÃO está disponível no ICP 3.1.2 | |23772 | Implementações - a coluna CREATED não é exata ou está totalmente errada | |21076 | EVRY:ICP 311 - os itens selecionados são desmarcados na Edição | |21722 | Nova instalação da 3.1.1 - os serviços são designados ao VIP principal, não ao VIP proxy | |20586 | Cluster HA: inconsistência no status do pod - em execução ou finalizando| |23253 | A Classificação de Implementações do Console da Web do ICP (Data de Criação) não funciona corretamente | |19933 | Senha LDAP em texto sem formatação na UI do navegador no ICP 2.1.0.2 | |19562 | A UI de procura do usuário LDAP não está em sincronização com a resposta de back-end | |14225 | A janela pop-up é muito pequena para mostrar a sequência LDAP ao criar uma equipe | |23763 | Problema de usabilidade na criação de uma página da equipe | |20867 | A inclusão da conexão LDAPS trava o contêiner platform-identity-mgmt | |24999 | O Login do Console falha com 400 Solicitação Inválida, MariaDB `ERROR 1210 (HY000) at line 1: WSREP (galera) not started` | |21567 | O pod auth-idp do ICP 3.1.1 continua reiniciando | |21396 | No Grupo, um Usuário aparece 2 vezes | |11994 | Comportamento inconsistente/errôneo ao configurar o LDAP para ICP | |23530 | Problema para correção Negado (usuário LDAP não reconhecido como administrador de cluster) | |20463 | LDAPS - código incorreto - código de erro 49 | |19930 | Efetuar login 10 a 20 vezes em uma linha com o login cloudctl foi bem-sucedido apenas 2 ou 3 vezes | |21331 | Login via bx pr não funciona de forma consistente a partir do pipeline Jenkins | |22261 | Erros OICD para produtos pós-instalados (TA/MC/CAM) quando o SAML está ativado | |21897 | Integração do OIDC para cargas de trabalho | |22980 | Solicite correção para mudar a porta 9443/TCP sobre SSL para TLSv1.2 | |21555 |Não é possível efetuar login com o LDAP, mas é possível incluir usuários sem problemas | |22583 | A interface da web não responde ao navegar para uma equipe | |24112 | MCM 3.1.2. O painel do Grafana não reflete as mudanças quando um componente do Aplicativo é movido para outro cluster | |23954 | A função `New Rule` em `Gerenciar Lista de Desbloqueio para o Mutation Advisor` é susceptível à vulnerabilidade de cross site scripting (XSS) armazenada. | |25439 | ICP 3.1.0 - Comportamento do VA no caso de imagens não suportadas | |18542 | Vulnerability Advisor - IP em vez do nome do cluster no console | |18940 | Desempenho e HA do CAM |

Instalação e Upgrade

A revisão encontrou erros de instalação e de upgrade com frequência.

A instalação é interrompida ou falha

A instalação é interrompida ou falha.

Sintomas

A instalação é paralisada ou falha sem explicação, aviso ou mensagem de erro clara.

Causas

As falhas ou interrupções de instalação podem ser atribuídas a vários motivos. Para identificar a causa raiz, obtenha um relatório detalhado durante a instalação.

Diagnosticando o problema

Execute o comando de implementação novamente com a opção detalhada (`-vvv`) especificada.

- Edições Standard:

```
docker run -e LICENSE=accept --net=host -v "$(pwd)":/installer/cluster ibmcom/icp-
inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee install -vvv
```

- Edição da comunidade:

```
docker run -e LICENSE=accept --net=host -v "$(pwd)":/installer/cluster ibmcom/icp-
inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0 install -vvv
```

Também é possível executar a implementação novamente com um cronômetro. Esse cronômetro controla o tempo que é gasto para cada tarefa e essas informações podem ajudá-lo a identificar a etapa que está causando problemas.

- Edições Standard:

```
docker run -e LICENSE=accept -e ANSIBLE_CALLBACK_WHITELIST=profile_tasks,timer --net=host -t -v
"$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee
install
```

- Edição da comunidade:

```
docker run -e LICENSE=accept -e ANSIBLE_CALLBACK_WHITELIST=profile_tasks,timer --net=host -t -v
"$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0
install
```

O componente etcd falha ao iniciar

Durante a instalação, o componente etcd falha ao iniciar.

Sintomas

A instalação é encerrada após 10 minutos esperando o componente etcd iniciar.

Nota: por padrão, a instalação espera por 10 minutos antes de encerrar. É possível mudar o tempo de espera atualizando o parâmetro `wait_for_timeout` no arquivo `/<installation_directory>/cluster/config.yaml`. Especifique o valor de parâmetro em segundos.

Causas

- O contêiner kubelet falha ao iniciar.
- A imagem de etcd não pode ser obtida.
- O contêiner etcd não pode iniciar.
- A porta etcd não está acessível. A porta etcd padrão é 4001.

Resolvendo o problema

- Verifique se kubelet está em execução:

  1. Efetue login em seu nó principal.
  2. Execute o comando a seguir para verificar o status de kubelet:

     ```
     systemctl status kubelet
     ```

     Se kubelet não estiver em execução, execute o comando a seguir para obter os logs:

     ```
     journalctl -u kubelet &>kubelet.log
     ```

- Em um ambiente IBM® Cloud Private-CE, certifique-se de que tenha acesso à Internet e seja possível extrair a imagem etcd.
- Nos ambientes do IBM® Cloud Private Enterprise e Cloud Native, assegure que a imagem etcd seja copiada no nó de inicialização. Use o comando a seguir para verificar se a imagem etcd é carregada:

  ```
  docker images | grep etcd
  ```

- Verifique se o contêiner etcd foi iniciado:

  1. Efetue login em seu nó principal como um usuário com permissão raiz.
  2. Execute o comando a seguir para verificar o status do contêiner etcd:

     ```
     docker ps | grep etcd
     ```

     Se o contêiner etcd não foi iniciado, execute os comandos a seguir para obter os logs:

     1. Obtenha o ID do contêiner etcd:

        ```
        docker ps -a | grep etcd
        ```

     2. Execute o comando para obter os logs:

        ```
        docker logs <etcd container ID> &>etcd.log
        ```

- Verifique se é possível se conectar ao etcd. Execute o comando a seguir:

  ```
  telnet <master node IP or cluster virtual IP address> 4001
  ```

- Assegure-se de que seu firewall não esteja bloqueando a porta 4001.

Falha ao conectar-se por meio de ssh

A instalação falha devido a um problema de conexão do SSH.

Sintomas

A mensagem de erro a seguir é exibida e a instalação falha.

```
fatal: [x.x.x.x] => Failed to connect to the host via ssh'
```

Causas

O SSH sem senha não é configurado na inicialização para outros nós no cluster.

Resolvendo o problema

- Para obter mais informações sobre como configurar o SSH sem senha, consulte Compartilhando chaves SSH entre nós do cluster.
- Assegure-se de que o SSH sem senha esteja configurado no nó de inicialização para todos os outros nós no cluster. Se um nó único for usado para inicialização e principal, ainda será necessário configurar o SSH sem senha nesse nó. Execute os comandos a seguir para verificar se o SSH sem senha está configurado no cluster:

  ```
  ssh -vvv -i cluster/ssh_key root@x.x.x.x
  ```

Falha ao criar contêineres

A instalação falha durante a criação do contêiner.

Sintomas

A instalação falha com a mensagem de erro a seguir:

```
Error creating container: UnixHTTPConnectionPool(host='localhost', port=None): Read timed out. (read
timeout=XXX)
```

Resolvendo o problema

Aumente o valor para o parâmetro `docker_api_timeout`. Para obter informações adicionais sobre o parâmetro `docker_api_timeout`, consulte Customizando o cluster com o arquivo config.yaml.

Kubelet falha ao iniciar

O kubelet falha ao iniciar. Aprenda sobre o cenário, o sintoma e a causa.

- Cenário 1: o kubelet falha ao iniciar devido a um certificado não autorizado
- Cenário 2: o kubelet falha ao iniciar devido a uma sinalização de kernel inválida

Cenário 1: o kubelet falha ao iniciar devido a certificados não autorizados

Sintomas

O kubelet falha ao iniciar e exibe uma mensagem semelhante à seguinte saída:

```
hyperkube[1554]: E0814 05:07:21.428053    1554 bootstrap.go:195] Part of the existing bootstrap
client certificate is expired:      2018-08-14 03:46:00 +0000 UTC
hyperkube[1554]: F0814 05:07:21.438534    1554 server.go:262] failed to run Kubelet: cannot create
certificate signing request: Unauthorized
```

Causa

O certificado de cliente que foi usado pelo kubelet expirou. Kubelet não conseguiu renovar o certificado. É necessário gerar um novo token de autoinicialização para associar novamente e obter um novo certificado.

Resolvendo o problema

Conclua a tarefa a seguir para resolver o problema:

1. Faça download do binário `kubeadm` e efetue login no nó principal com o comando a seguir:

   ```
   curl -L -o /usr/local/bin/kubeadm https://storage.googleapis.com/kubernetes-
   release/release/v1.11.1/bin/linux/amd64/kubeadm
   chmod +x /usr/local/bin/kubeadm
   ```

2. Execute o comando a seguir para gerar um novo token de autoinicialização:

   ```
   kubeadm --kubeconfig= /etc/cfc/conf/admin.kubeconfig token create --ttl 24h0m0s
   ```

   A saída é semelhante ao exemplo a seguir:

   ```
   k5ojt0.ko1wov52mdvnqbg6
   ```

   Nota: salve esta saída de comando para uso posterior.

3. Obtenha o novo certificate:

   - Efetue login no nó que falhou.
   - Edite a configuração kubelet. Em `/etc/cfc/kubelet/kubelet-bootstrap-config`, substitua `users.user.token` pelo valor que você salvou na etapa dois, `k5ojt0.ko1wov52mdvnqbg6`.
   - Reinicie o Kubelet.

Cenário 2: o kubelet falha ao iniciar devido a uma sinalização de kernel inválida

Sintomas

Ao fazer upgrade da 3.1.2 para a 3.2.0, o kubelet falha ao iniciar e exibe uma mensagem semelhante à saída a seguir:

```
hyperkube[804]: F1023 17:02:19.964867     804 kubelet.go:1333] Failed to start ContainerManager
[Invalid kernel flag: vm/overcommit_memory, expected value: 1, actual value: 0, Invalid kernel flag:
kernel/panic, expected value: 10, actual value: 0, Invalid kernel flag: kernel/panic_on_oops,
expected value: 1, actual value: 0]
```

Causa

No IBM Cloud Private 3.2.0, por padrão, o `protectKernelDefaults` é configurado como `true` no `/etc/cfc/kubelet/kubelet-service-config` de acordo com o requisito do CIS.

Resolvendo o problema

Para resolver esse problema ao fazer upgrade, configure `protectKernelDefaults` como `false` antes ou após a instalação.

- Antes da instalação

  Atualize o `cluster/config.yaml` para configurar `kubelet_extra_args: ["--protect-kernel-defaults = false"]`.

- Após a instalação

  1. Siga as Etapas de 1 a 5 em Reconfigurando o Kubelet em um cluster em tempo real para reconfigurar o kubelet.
  2. Na Etapa 2 em Editar o arquivo de configuração, configure protectKernelDefaults: false.

O controlador de ingresso nginx não é iniciado

O controlador de ingresso nginx falhou ao ser iniciado.

Sintomas

A mensagem de erro a seguir é exibida nos logs do Docker:

```
epoll_create() failed (24: Too many open files)
```

Causas

O número de arquivos abertos excedeu o limite do sistema. Por padrão, no máximo 1024 arquivos podem ser abertos. É possível verificar o limite de arquivos abertos executando o comando `ulimit -n`.

Resolvendo o problema

1. Abra o arquivo `docker.service` e inclua o código a seguir:

   ```
   LimitNOFILE=infinity
   LimitNPROC=infinity
   LimitCORE=infinity
   ```

2. Reiniciar Docker.

   ```
   docker restart
   ```

Pods falham ao inicializar

A inicialização dos pods falha porque eles não podem configurar o cookie `dm_task_set_cookie`.

Sintomas

O status para alguns pods é `rpc error: code = 2 desc = Error response from daemon: {"message":"devmapper: Error activating devmapper device for '68781a983adeef6156b303e9ffb18251a5fdd7267d0591226e11066dc9e9fe7d-init': devicemapper: Can't set cookie dm_task_set_cookie failed"}`

Causas

As versões do Docker anteriores à versão 17.06.2 não configuram adequadamente o cookie `dm_task_set_cookie`.

Resolvendo o problema

Reinicie o serviço do Docker executando o comando a seguir:

```
dmsetup udevcomplete_all
```

A instalação falha quando o firewalld está ativado

A instalação do IBM® Cloud Private no Red Hat Enterprise Linux 7 poderá falhar se firewalld estiver ativado.

Sintomas

A instalação falha com uma mensagem de erro que é semelhante à mensagem a seguir:

```
TASK [iptables : Creating firewalld rules for Etcd]
*********************************************************************************************
*************************************************************************
failed: [9.111.250.102] (item=4001) => {"failed": true, "item": 4001, "msg": "ERROR: Exception
caught: org.fedoraproject.FirewallD1.Exception: COMMAND_FAILED"}
failed: [9.111.250.102] (item=2380) => {"failed": true, "item": 2380, "msg": "ERROR: Exception
caught: org.fedoraproject.FirewallD1.Exception: COMMAND_FAILED"}


PLAY RECAP
*********************************************************************************************
*********************************************************************************************
************************
9.111.250.102              : ok=81   changed=16   unreachable=0    failed=1
localhost                  : ok=29   changed=0    unreachable=0    failed=0
```

Resolvendo o problema

1. Reinicie o serviço de daemon do firewall executando o comando a seguir:

   ```
   service firewalld restart
   ```

   (OR)

   ```
   systemctl restart firewalld.service
   ```

2. Tente novamente a instalação.

Controlador de ingresso relatado: `epoll_create() failed (24: Too many open files)`

O controlador de ingresso não funciona com uma máquina POWER, que possua 160 núcleos. O controlador de ingresso poderá falhar quando ele estiver em execução em um nó com muitos núcleos.

Causas

O controlador de ingresso pode estar em execução em um nó que possui muitos núcleos. O número máximo de descritores de arquivos abertos é calculado com a seguinte fórmula: `*RLIMIT_NOFILE/worker-processes) - 1024`. Para resolver, é possível diminuir o valor dos `processos do trabalhador` ou aumentar o valor do `RLIMIT_NOFILE` do contêiner.

Solução um: Edite o `configMap` de `nginx-ingress-controller` com um valor diminuído de `worker-processes`.

1. Para editar o `configmap` de `nginx-ingress-controller`, execute o seguinte comando:

   ```
   kubectl -n kube-system edit cm nginx-ingress-controller
   ```

2. Inclua `worker-processes: "2"` no `configMap`, tal como no exemplo a seguir. **Nota:** o valor pode não ser `2`, dependendo de sua configuração do `sysctl`.

   ```
   # Edite o objeto a seguir. As linhas que começam com '#' são ignoradas, # e um arquivo vazio
   abortarão a edição. If an error occurs while saving this file will be
   # reopened with the relevant failures.
   #
   apiVersion: v1
   data:
    body-size: "0"
    disable-access-log: "true"
    worker-processes: "2"
   ```

Solução dois: Configure o docker LimitNOFILE para um valor maior para aumentar o valor de RLIMIT_NOFILE do contêiner do controlador de ingresso.

Nota: O número máximo de descritores de arquivos abertos é calculado com a fórmula `RLIMIT_NOFILE / worker-processes) - 1024`. É possível aumentar `RLIMIT_NOFILE` mudando o Docker `LimitNOFILE` para um valor maior.

1. Execute o comando a seguir para listar seus pods que executam o ingresso:

   ```
   $ kubectl -n kube-system get pods -owide | grep nginx-ingress
   nginx-ingress-lb-amd64-sknw6                              1/1       Running   0          35m
   9.111.255.21    9.111.255.21
   ```

2. Efetue logon no nó em que o controlador de ingresso está em execução e edite o arquivo `/lib/systemd/system/docker.service` e atualize `LimitNOFILE` para um valor alto. Se `LimitNOFILE` estiver configurado como infinito, isso significa que o valor é `65536` ou `1048576 (2^20)`, com base em seu sistema operacional.

   ```
   [Service]
   LimitNOFILE=<your value>
   ```

   `LimitNOFILE` não pode ser maior que `fs.file-max`. Se precisar atualizar `fs.file-max` para um valor maior, execute o seguinte comando:

```
sysctl -w fs.file-max=<your value>
```

3. Reiniciar Docker.

```
systemctl daemon-reload systemctl restart docker
```

Pods falham com `CrashLoopBackOff`

Os pods permanecem no status `CrashLoopBackOff` e não são recuperados.

Sintomas

O pod falha com mensagem semelhante à seguinte:

```
unexpected error watching template /etc/nginx/template/nginx.tmpl: no space left on device
```

Causas

Nenhum espaço deixado no dispositivo.

Resolvendo o problema

1. Determine se o problema é um problema do sistema de arquivos.
   - Execute o comando a seguir para determinar se o uso de espaço em disco está cheio: `df -h`
   - Execute o comando a seguir para determinar se o uso de espaço `inode` está cheio: `df -i`
   - Execute o comando a seguir para ver se há um fd não liberado que está marcado como excluído: `lsof`
2. Use o comando, `lsof | grep inotify | wc -l` para verificar o uso de `inotify`. Use o comando `sysctl fs.inotify.max_user_watches` para verificar os valores atuais. Você pode ter atingido o limite no número total de observações de `inotify`. É possível aumentar o limite em `fs.inotify.max_user_watches` e reiniciar os pods.

```
# sysctl fs.inotify.max_user_watches=524288
fs.inotify.max_user_watches = 524288
# kubectl delete pod nginx-ingress-lb-amd64-6j9zm -n kube-system
pod "nginx-ingress-lb-amd64-6j9zm" deleted
```

Substituindo um nó principal

Quando ocorre uma falha de hardware, talvez você ache que o nó principal não está funcionando corretamente. Conclua estas etapas para substituir um nó principal.

1. Recrear o novo nó principal. Assegure-se de que o novo nó principal tenha o mesmo nome do host, endereço IP e nome da interface que o nó principal antigo.
2. Configure a autenticação SSH para o novo nó principal. Mantenha a mesma senha ou autenticação de chave SSH que o nó principal antigo.
3. Execute um comando para substituir o nó principal:
   - Para o Linux®, execute o seguinte comando:

     ```
     docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-inception-amd64:3.2.0-ee install -l master-node-ip
     ```

   - Para o Linux® on Power® (ppc64le), execute o seguinte comando:

     ```
     docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-inception-ppc64le:3.2.0-ee install -l master-node-ip
     ```

   - Para o Linux® on IBM® Z and LinuxONE, execute o seguinte comando:

     ```
     docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-inception-s390x:3.2.0-ee install -l master-node-ip
     ```

Atualizando os membros `etcd`

Se o pod `etcd` executar no novo nó principal, ele poderá falhar ao iniciar devido a dados inconsistentes. Para atualizar os membros `etcd`, consulte as etapas em O Pod do etcd falhou ao iniciar devido a dados inconsistentes.

comando manifest-tool não localizado

Comando `manifest-tool` não localizado ao construir imagens de multiarquitetura.

Sintomas

A construção da imagem de multiarquitetura falha, conforme indicado na mensagem de erro a seguir:

```
TAREFA [icp-registry-image-push : construindo imagens de multiarquitetura]
****************************************
COM FALHA - TENTANDO NOVAMENTE: construindo imagens de multiarquitetura (3 tentantivas restantes).
COM FALHA - TENTANDO NOVAMENTE: construindo imagens de multiarquitetura (2 tentantivas restantes).
COM FALHA - TENTANDO NOVAMENTE: construindo imagens de multiarquitetura (1 tentantiva restante).
fatal: [9.37.136.40-> 9.37.136.40 ]: FAILED! => changed=true
  attempts: 3
  cmd: |-
    export NO_PROXY=spectrumdiscover:8500
     rc=0
     for image in $(cat /tmp/image-registry/image-list.txt | sort -u); do
     manifest-tool push from-args --platforms linux/amd64,linux/ppc64le,linux/s390x --template
spectrumdiscover:8500/ibmcom/${image//:/-ARCH:} --target spectrumdiscover:8500/ibmcom/$image --
ignore-missing
     if [[ $? -ne 0 ] ] & [ [! "$image" =~ (icp-helm-api|icp-helm-rudder|icp-cert-gen) ]]; then
     echo "A construção de imagem de multiarquitetura falhou para $image"
     rc=1
     fi
     done
     exit $rc
  delta: '0:00:00.082852'
  end: '2018-09-11 13 :12:24.034750'
  msg: código de retorno não zero
  rc: 1
  start: '2018-09-11 13:12:23.951898'
  stderr: |-
    /bin/bash: linha 3: manifest-tool: comando não localizado
    /bin/bash: linha 3: manifest-tool: comando não localizado
    /bin/bash: linha 3: manifest-tool: comando não localizado
```

Causas

`manifest-tool` padroniza para o caminho do sistema, `/usr/local/bin`. No entanto, `/usr/local/bin` não está no PATH do ambiente do sistema. Por exemplo:

```
# env | grep PATH PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

Resolvendo o problema

1. No primeiro nó principal, copie `manifest-tool` do caminho `/usr/bin/manifest-tool` para o caminho `/usr/local/bin/manifest-tool`.
2. Reinstale o cluster de ICP inteiro.

Falha ao incluir o nó do Vulnerability Advisor (VA)

Ocorre uma falha quando você tenta incluir um nó do VA.

Sintomas

Ocorre uma falha quando você executa o comando a seguir para incluir um nó do VA:

```
docker run --rm -t -e LICENSE=accept --net=host -v \
  $(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee va -l
\
  ip_address_vanode1,ip_address_vanode2
```

O seguinte erro aparece:

```
ERRO! Não é possível recuperar os conteúdos do arquivo
Não foi possível localizar ou acessar '/installer/playbook/plays/storage-va.yaml' no Controlador
Ansible.
Se você estiver usando um módulo e esperar que o arquivo exista no remoto, consulte a opção
remote_src
```

Resolvendo o problema

Emita os comandos a seguir para uma solução alternativa para o problema:

```
sudo docker run -it --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-
inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee bash

sed -i '/storage/d' playbook/va.yaml

./installer.sh va -l <ip_address_vanode1,ip_address_vanode2>

sair
```

va-live-crawler causa alto uso de CPU e memória

O Live-crawlers em cada nó de seus clusters do IBM® Cloud Private pode causar alta ocupação de CPU e memória.

Resolvendo o problema

É possível incluir limites de recursos de CPU e memória como uma solução alternativa para o problema.

1. Localize os dois daemonsets a seguir.

   ```
   vulnerability-advisor-process-ma-crawler
   vulnerability-advisor-live-crawler
   ```

2. Edite os daemonsets para incluir limites de recurso.

   ```
   kubectl edit ds vulnerability-advisor-live-crawler -n kube-system
   kubectl edit ds vulnerability-advisor-process-ma-crawler -n kube-system
   ```

3. Inclua seus limites de CPU e de memória.

   ```
   resources:
           limits:
             cpu: 100m
             memory: 128Mi
           requests:
             cpu: 50m
             memory: 64Mi
   ```

Desativar o serviço `custom-metrics-adapter` ao desativar o serviço de monitoramento durante a instalação do IBM Cloud Private

Sintomas

Se você não desativar o `custom-metrics-adapter`, o status de implementação do adaptador aparecerá como `unhealthy` nas páginas do painel do ICP e da visualização de implementação.

Causas

O serviço `custom-metrics-adapter` tem uma dependência no serviço de monitoramento.

Resolvendo o problema

Reative o serviço de monitoramento. Implemente o gráfico Helm `ibm-icpmonitoring` mais recente no modo gerenciado por meio do catálogo do IBM Cloud Private no namespace kube-system.

Manutenção do etcd

Seu pod etcd pode falhar ao iniciar devido ao espaço do banco de dados excedido.

Para manter recursos de armazenamento que o keyspace etcd usa, consulte Gerenciando clusters etcd. Configure a cota de espaço e conclua a compactação do histórico e a desfragmentação.

Manutenção do diretório etcd write-ahead log (WAL)

Por padrão, o diretório etcd do WAL é configurado como `etcd_wal_dir: /var/lib/etcd-wal` no `config.yaml`. É possível configurar o diretório `/var/lib/etcd-wal` para um diretório de log remoto centralizado para criação de log persistente.

O valor de dimensionamento do log etcd do WAL é configurado em `/etc/cfc/pods/etcd.json` por `--max-wals`. Por exemplo, se `--max-wals=0`, o número máximo de arquivos WAL que são retidos será ilimitado. Se `--max-wals=5`, o número máximo de arquivos WAL que são retidos é 5. Se não houver nenhum número de arquivo designado a `--max-wals` em `etcd.json`, o valor padrão de dimensionamento de log do WAL será 5.

Para configurar manualmente o número do arquivo, por exemplo, para 5, siga estas etapas:

1. Efetue login em um nó principal de seu ambiente de alta disponibilidade (HA) ou efetue login em seu nó etcd, caso tenha separado o etcd do principal.
2. Pare o etcd executando o comando a seguir:

   ```
   mv /etc/cfc/pods/etcd.json /etc/cfc/etcd.json
   ```

   Importante: não crie nenhum arquivo de backup em `/etc/cfc/pods` e não execute o comando e a opção a seguir: `cp /etc/cfc/pods/etcd.json /etc/cfc/pods/etcd.json.orig`.
3. Execute o comando a seguir para verificar se o etcd foi interrompido. Se não houver saída, o etcd foi interrompido:

   ```
   docker ps | grep etcd
   ```
4. Edite o arquivo `/etc/cfc/etcd.json` para configurar `--max-wals=5`.
5. Inicie etcd executando o comando a seguir:

   ```
   mv /etc/cfc/etcd.json /etc/cfc/pods/etcd.json
   ```
6. Para verificar se o etcd foi executado, execute o comando a seguir:

   ```
   docker ps | grep etcd
   ```

   O comando pode ser semelhante à saída a seguir:

   ```
   # docker ps | grep etcd
   fbd4e804a818        e21fb69683f                          "etcd --name=etcd0 -…"   10 minutes
ago      Up
   10 minutes                          k8s_etcd_k8s-etcd-172.29.214.11_kube-
   system_b93a2f44fc31e2719f2ec07ae0f1bf43_3
   6de280044570        mycluster.icp:8500/ibmcom/pause:3.1   "/pause"                 12 minutes
ago
   Up 12 minutes                          k8s_POD_k8s-etcd-172.29.214.11_kube-
   system_b93a2f44fc31e2719f2ec07ae0f1bf43_3
   ```
7. Execute o comando a seguir para verificar o número do arquivo que está designado a `max-wals`:

   ```
   ps -ef | grep "name\=etcd" | grep max-wals
   ```

   Nota: talvez seja necessário aguardar alguns minutos para que o log do WAL em `/var/lib/etcd-wal` seja reduzido para 5.
8. Repita as etapas de 1 a 7 em cada nó principal (etcd).

Consulte Configurações do etcd para obter mais informações.

O pod etcd falhou ao ser iniciado devido ao espaço do banco de dados excedido

Sintomas:

Os pods etcd estão no estado de CrashloopBackOff. O log de erros mostra a mensagem de erro a seguir:

```
Error from server: etcdserver: mvcc: database space exceeded
```

Causa :

Os recursos de armazenamento etcd ou o diretório etcd WAL precisa de manutenção.

Resolvendo o problema :

1. Limpe o diretório do WAL etcd. Por padrão, o diretório é configurado como `/var/lib/etcd-wal`. É possível usar `df -h | grep etcd-wal` para verificar o uso do armazenamento.

   Se o disco estiver cheio, consulte Diretório de manutenção do etcd write-ahead log (WAL) para limpar os arquivos etcd WAL.

2. Para liberar espaço de armazenamento, siga as instruções para desfragmentação em Gerenciamento de clusters etcd.

O pod etcd falhou ao iniciar devido a dados inconsistentes

Sintomas:

Os pods etcd falharam ao iniciar. O log de erros mostra a saída a seguir:

```
  2018-12-27 17:54:22.267699 C | raft: 8362bb192cc722e8 state.commit 5801 is out of range [2320232,
2320232]
  panic: 8362bb192cc722e8 state.commit 5801 is out of range [2320232, 2320232]
  goroutine 1 [ executando ]: github.com/coreos/etcd/cmd/vendor/github.com/coreos/pkg/capnslog.
(*PackageLogger).Panicf(0xc420161420, 0xf975a1, 0x2b, 0xc420058340, 0x4, 0x4)

/tmp/etcd/release/etcd/gopath/src/github.com/coreos/etcd/cmd/vendor/github.com/coreos/pkg/capnslog/p
kg_logger.go:75 +0x15c
```

Causa: este erro é devido a dados inconsistentes do etcd.

Resolvendo o problema :

1. Configure `etcdctl`:

   1. Para acessar seu cluster do etcd usando a interface da linha de comandos (CLI), deve-se instalar e configurar o `etcdctl`, que é o cliente da linha de comandos para o etcd. É possível obter o arquivo binário `etcdctl` por meio da imagem `ibmcom/etcd:v3.2.18`, executando o comando a seguir:

      ```
      docker run --rm -v /usr/local/bin:/data <cluster_CA_domain>:8500/ibmcom/etcd:v3.2.18 cp
      /usr/local/bin/etcdctl /data
      ```

      Em que `<cluster_CA_domain>` é o domínio da autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

   2. Configure o terminal como um de seus membros do etcd disponíveis executando o comando a seguir:

      ```
      export endpoint=<Endpoint IP address>
      ```

   3. Para usar a API do etcdctl v3, configure um alias executando o comando a seguir:

      ```
      alias etcdctl3="ETCDCTL_API=3 etcdctl --endpoints=https://${endpoint}:4001 --
      cacert=/etc/cfc/conf/etcd/ca.pem --cert=/etc/cfc/conf/etcd/client.pem --
      key=/etc/cfc/conf/etcd/client-key.pem"
      ```

2. Atualize os membros etcd:

   1. Verifique seus membros de cluster do etcd existentes executando o comando a seguir. O comando pode ser semelhante à saída a seguir:

      ```
      # etcdctl3 member list
      2bc7764897fe35ec, started, etcd1, https://<Member IP address>, https://<Member IP
      address>:4001
      77a992292013374b, started, etcd0, https://<Member IP address>, https://<Member IP
      address>:4001
      f0f3d76c8bf22bca, started, etcd2, https://<Member IP address>, https://<Member IP
      address>:4001
      ```

      O nó etcd2 é o nó que falhou ao iniciar.

2. No nó com falha, etcd2 no exemplo, pare o etcd executando o comando a seguir:

   ```
   mv /etc/cfc/pods/etcd.json /etc/cfc/etcd.json
   ```

3. Execute o comando a seguir para verificar se o etcd é executado. Se não houver saída, o etcd não está em execução:

   ```
   docker ps | grep etcd
   ```

4. Remova o membro do etcd2 antigo executando o comando a seguir:

   ```
   # etcdctl3 member remove f0f3d76c8bf22bca
   Member f0f3d76c8bf22bca removed from cluster 71e83e6eb99a602f
   ```

5. Inclua o membro etcd2 de volta executando o comando a seguir:

   ```
   # etcdctl3 member add etcd2 --peer-urls="https://9.111.255.212:2380"
   Member 969909b46db234fe added to cluster 71e83e6eb99a602f

   ETCD_NAME="etcd2"
   ETCD_INITIAL_CLUSTER="etcd1=https://9.111.255.206:2380,etcd0=https://9.111.255.130:2380,e
   tcd2=https://9.111.255.212:2380"
   ETCD_INITIAL_CLUSTER_STATE="existing"
   ```

6. No nó etcd2, limpe o diretório de dados etcd executando os comandos a seguir:

   ```
   # rm -r /var/lib/etcd/*
   # rm -r /var/lib/etcd-wal/*
   ```

7. No nó etcd2, edite o arquivo `/etc/cfc/pods/etcd.json` e inclua `--initial-cluster-state=existing`.

8. No nó etcd2, reinicie etcd executando o comando a seguir:

   ```
   mv /etc/cfc/etcd.json /etc/cfc/pods/etcd.json
   ```

3. Para verificar se o etcd é executado, execute o comando a seguir:

   ```
   docker ps | grep etcd
   ```

Resolução de Problemas de Clusters Largeões (1000 Nós do Trabalhador)

Para clusters grandes que contêm mais de 1000 nós do trabalhador, os problemas e resoluções a seguir estão disponíveis para resolução de problemas.

Sobrecarga de SYN por Calico em cluster em larga escala

Sintomas:

```
[852955.742157] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
[853969.644648] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
[854761.264262] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
[855197.510945] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
[855914.609259] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
[856139.517655] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
[857159.485206] TCP: request_sock_TCP: Possible SYN flooding on port 179. Sending cookies.  Check
SNMP counters.
```

Causa :

As sessões TCP principais persistentes ou de longo prazo no cluster estão entre o controlador Kube e o kubelet e o Calico BGP. As sessões N-1 a partir dessas duas constituem pelo menos 999+999 = 1998 sessões TCP de longo prazo.

Resolvendo o problema :

Configure o valor de `net.core.somaxconn` para `2048` em todos os nós no cluster. Por exemplo:

```
sysctl -w net.core.somaxconn=2048
```

Como alternativa, é possível incluir o `sysctl -w net.core.somaxconn=2048` em `/etc/sysctl.conf` e, em seguida, executar `sysctl -p`.

Se o problema persistir, aumente o valor de `net.core.somaxconn` por incrementos de 128 até que o problema seja resolvido.

Peer tornou-se inativo no log `etcd`

Sintomas:

```
{"log":"2018-12-27 05:02:37.247750 I | rafthttp: peer f3826ae8ceca1970 became
inactive\n","stream":"stderr","time":"2018-12-27T05:02:37.247908001Z"}
{"log":"2018-12-27 05:02:37.247773 W | rafthttp: lost the TCP streaming connection with peer
f3826ae8ceca1970 (stream MsgApp v2 reader)\n","stream":"stderr","time":"2018-12-
27T05:02:37.247915819Z"}
{"log":"2018-12-27 05:03:09.913771 I | rafthttp: peer f3826ae8ceca1970 became
active\n","stream":"stderr","time":"2018-12-27T05:03:09.913957217Z"}
{"log":"2018-12-27 05:03:09.913815 I | rafthttp: established a TCP streaming connection with peer
f3826ae8ceca1970 (stream Message reader)\n","stream":"stderr","time":"2018-12-
27T05:03:09.913985504Z"}
```

Causa :

Se o líder etcd atender a uma montagem grande de solicitações simultâneas, ele poderá causar alta latência para a rede.

Resolvendo o problema :

Para corrigir o problema, ajuste os parâmetros etcd `heartbeat-interval` e `election-timeout"` no `/etc/cfc/pods/etcd.json`.

Em um cluster de 1000 nós do trabalhador, configure os parâmetros a seguir:

```
"-- heartbeat-interval=500", "-- election-timeout=2500",
```

A CLI do Kubernetes (kubectl) está lenta ou não responsiva

Sintomas:

O kubectl fica lento ou atinge o tempo limite no cluster em larga escala.

Causa :

O kube-apiserver responde lentamente.

Resolvendo o problema :

Atualize ou modifique os parâmetros a seguir no `/etc/cfc/pods/master.json` em todos os nós principais para o contêiner apiserver:

```
"--max-requests-inflight=1500", "-max-mutating-requests-inflight=500", "--target-ram-mb=65000", "--
runtime-config=scheduling.k8s.io/v1beta1 = true"
```

Para o contêiner gerenciador de controlador:

```
"--kube-api-qps=100", "--kube-api-burst=100"
```

Para o contêiner do planejador:

```
"--kube-api-qps=100"
```

Erro no estouro da tabela de vizinhos no log do kernel

Sintomas:

Erros semelhantes podem aparecer no diretório `kern.log` localizado no diretório `/var/log/`.

```
May  8 23:02:39 icp10m2 kernel: [524480.323013] neighbour: arp_cache: neighbor table overflow!
May  8 23:02:39 icp10m2 kernel: [524480.323072] neighbour: arp_cache: neighbor table overflow!
May  8 23:02:39 icp10m2 kernel: [524480.323414] neighbour: arp_cache: neighbor table overflow!
May  8 23:02:39 icp10m2 kernel: [524480.323560] neighbour: arp_cache: neighbor table overflow!
```

Causa :

Esses tipos de erros sempre ocorrem em redes grandes quando há muitas solicitações ARP que o servidor não é capaz de responder.

Resolvendo o problema :

Atualize `/etc/sysctl.conf` para configurar o `net.ipv4.vizinha .default.gc_thresh` com o valor apropriado em todos os nós.

Por exemplo, configure os valores a seguir em um cluster de 1000 nós:

```
net.ipv4.neigh.default.gc_thresh1 = 4096
net.ipv4.neigh.default.gc_thresh2 = 8192
net.ipv4.neigh.default.gc_thresh3 = 8192
```

Erro de excesso de arquivos abertos (24) no log do diário

Sintomas:

Aparecem erros semelhantes a `Excesso de arquivos abertos (24)` no log do diário.

Causa :

O aplicativo, comando ou script está atingindo o limite máximo de arquivos abertos que é permitido pelo Linux®.

Resolvendo o problema :

Execute o comando a seguir:

```
sysctl -w fs.file-max=512
```

Como alternativa, é possível incluir `sysctl -w fs.file-max=512` em `/etc/sysctl.conf` e, em seguida, executar `sysctl -p`.

Falha na instalação ao aguardar o Tiller iniciar

O instalador pode esperar um tempo para que o Tiller seja iniciado; e a instalação poderá falhar se outro processo usar a porta do Tiller 44134.

É possível verificar o log do Tiller nos nós principais e obter o ID do contêiner executando o seguinte comando:

```
docker ps -a |grep tiller
```

Observe que em um cluster de alta disponibilidade (HA), você deve executar o comando anterior em cada nó principal.

Depois de obter o ID do contêiner, é possível verificar o log executando o seguinte comando:

```
docker logs <tiller_container_id>
```

Você verá um erro `endereço já em uso`.

Causa

Outro processo, por exemplo, kube-apiserver, estava tentando se conectar ao etcd e o lado do cliente da conexão TCP ocupava a porta 44134.

Resolvendo o problema

A solução alternativa é encerrar forçosamente o processo que ocupa a porta do Tiller 44134. Para obter o ID do processo, execute o seguinte comando:

```
netstat -plan |grep 44134
```

Depois de obter o ID do processo que ocupa a porta 44134, execute o seguinte comando para encerrar o processo:

```
kill -9 <process_id>
```

Quando a porta 44134 for liberada, o Kubernetes irá reiniciar automaticamente o pod do Tiller novamente. O instalador continua se o Tiller for iniciado.

O IBM Cloud Private no Azure inicia incorretamente após a instalação

O IBM® Cloud Private no Azure inicia incorretamente em um ambiente de alta disponibilidade (HA) após a instalação do cluster.

Sintomas

- O gráfico do Helm `auth` para uma implementação de três nós principal falha para os seguintes pods:
  - `auth-pdp`
  - `auth-idp`
- Depois de reiniciar o cluster, cada nó principal não é sincronizado com seu cluster.
- A implementação do gráfico do Helm `logging` falha.

Depois de instalar o IBM Cloud Private, pode aparecer a seguinte mensagem de erro:

```
FALHA - TENTANDO NOVAMENTE: Esperando o auth-pdp iniciar (3 novas tentativas restantes).
```

Causas

Há duas causas possíveis:

- Os valores de parâmetro `aadClientId` e `aadClientSecret` em seu arquivo YAML são inválidos.
- Ao referenciar um principal de serviço que não tem a função do Azure de *Contribuidor*, o cluster do IBM Cloud Private no Azure inicia incorretamente.

Resolvendo o problema

Atualize os parâmetros `aadClientId` e `aadClientSecret` em seu arquivo YAML. Consulte Ativando o Azure como um provedor em nuvem para obter informações adicionais.

À medida que você configura o Azure, certifique-se de incluir o principal de serviço. Para obter informações adicionais, consulte a seção *Principal de serviço* na página de requisitos do Azure para obter informações adicionais.

Sobrecarga de logs devido ao uso de systemd como um driver cgroup

Ao usar o driver `systemd` cgroups no kubelet e no docker, ocorrerá a sobrecarga da mensagem.

Sintomas

- As mensagens de log estão sendo sobrecarregadas como o seguinte status:

```
  systemd[1]: libcontainer-29519-systemd-test-default-dependencies.scope: O escopo não possui
PIDs. Recusando.
  systemd[1]: libcontainer-29519-systemd-test-default-dependencies.scope: O escopo não possui
PIDs. Recusando.
  systemd[1]: Created slice libcontainer_29519_systemd_test_default.slice.
  kubelet[6547]: W1209 03:04:49.649981    6547 container.go:422] Falha ao obter
RecentStats("/libcontainer_29519_systemd_test_default.slice") ao determinar a próxima
manutenção: não é possível localizar dados no cache de memória
  systemd[1]: Removed slice libcontainer_29519_systemd_test_default.slice.
  systemd[1]: libcontainer-29554-systemd-test-default-dependencies.scope: O escopo não possui
PIDs. Recusando.
  systemd[1]: libcontainer-29554-systemd-test-default-dependencies.scope: O escopo não possui
PIDs. Recusando.
  systemd[1]: Created slice libcontainer_29554_systemd_test_default.slice.
  systemd[1]: Removed slice libcontainer_29554_systemd_test_default.slice.
  systemd[1]: libcontainer-29561-systemd-test-default-dependencies.scope: O escopo não possui
PIDs. Recusando.
  systemd[1]: libcontainer-29561-systemd-test-default-dependencies.scope: O escopo não possui
PIDs. Recusando.
  systemd[1]: Created slice libcontainer_29561_systemd_test_default.slice.
  kubelet[6547]: W1209 03:04:50.591527    6547 container.go:523] Falha ao atualizar
```

```
estatísticas para o contêiner "/libcontainer_29561_systemd_test_default.slice": open
/sys/fs/cgroup/memory/libcontainer_29561_systemd_test_default.slice/memory.use_hierarchy: não é
um arquivo ou diretório, continuando a enviar estatísticas por push
```

Causas

- Na discussão da comunidade, esse problema causado por um cgroup vazio é criado por runc para verificar o recurso systemd.
- Para obter mais detalhes, consulte:

```
https://github.com/opencontainers/runc/blob/master/libcontainer/cgroups/systemd/apply_systemd.go#L12
3
```

Resolvendo o problema

Como este problema não afeta as métricas do contêiner, podemos apenas configurar rsyslog para ignorar esses logs sobrecarregados.

O arquivo de configuração de rsyslog está localizado no arquivo `/etc/rsyslog.conf` ou `/etc/rsyslog.d/*.conf`.

Para ignorar a mensagem de erro anterior,

```
:rawmsg, contains, "libcontainer"        ~
```

deve ser incluído antes do comando de coleta de mensagens, por exemplo,

```
*.info;mail.none;authpriv.none;cron.none         /var/log/messages.
```

Em seguida, reinicie o serviço rsyslog, esse problema de mensagem de sobrecarga pode ser ignorado.

Transferindo funções do nó principal

Não há capacidade de recurso suficiente no nó principal quando os nós principal e de gerenciamento estão no mesmo arquivo host.

Sintomas

O arquivo de mensagens de log se torna muito grande e utiliza espaço em disco.

Causas

Há nós principais e de gerenciamento no mesmo arquivo host.

Resolvendo o problema

Transfira as funções de nó para um novo arquivo host: gerenciamento, proxy e Vulnerability Advisor.

- Transferindo funções de gerenciamento
- Transferindo funções de proxy
- Transferindo funções de Vulnerability Advisor

Transferindo funções de gerenciamento

Importante: os PersistentVolumes devem ser transferidos para um novo nó porque alguns pods possuem dados de armazenamento local no nó de gerenciamento.

1. Remova o rótulo de gerenciamento do nó principal:
    1. Obtenha o rótulo para seu nó principal executando o comando a seguir:

        ```
        kubectl get nodes <master.node.name> --show-labels
        ```

        A saída pode ser semelhante ao conteúdo a seguir:

```
 beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=
<master.node.name>,management=true,node-role.kubernetes.io/management=true
```

2. Remova os rótulos da função de `gerenciamento` do nó principal executando os comandos a seguir:

   ```
   kubectl label nodes <master.node.name> management- node-role.kubernetes.io/management-
   ```

   Sua saída pode ser semelhante à mensagem a seguir:

   ```
   <master.node.name> labeled
   ```

3. Verifique se os rótulos da função de `gerenciamento` são removidos do nó principal executando o seguinte comando:

   ```
   kubectl get nodes <master.node.name> --show-labels
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=
   <master.node.name>
   ```

2. Inclua seu novo nó de gerenciamento com as etapas a seguir:

   1. Prepare seu novo nó de gerenciamento para a instalação. Para obter mais informações, consulte Preparando o novo nó para a instalação.

   2. Remova o endereço IP do nó principal no arquivo host em seu nó de gerenciamento. Para acessar o arquivo host, execute o comando a seguir:

      ```
      vi /etc/hosts
      ```

   3. Inclua um novo nó de gerenciamento no cluster do IBM Cloud Private executando o seguinte comando:

      ```
      docker run -e LICENSE=accept --net=host \
      -v "$(pwd)":/installer/cluster \
      ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee management -l \
      ip_address_managementnode1,ip_address_managementnode2
      ```

3. Mova os dados de armazenamento local do nó principal para o novo nó de gerenciamento. Transfira os dados `elasticsearch` com as seguintes etapas:

   1. Faça backup dos dados `elasticsearch` criando um archive compactado dos dados a partir do diretório de armazenamento executando o comando a seguir:

      ```
      tar -czf ~/logging-backup.tar.gz /var/lib/icp/logging/elk-data/*
      ```

   2. Use a cópia do servidor para transferir o arquivo de dados para o novo nó principal executando o seguinte comando:

      ```
      scp ~/logging-backup.tar.gz <node_user>@<node_ip>:~/logging-backup.tar.gz
      ```

   3. Restaure os dados `elasticsearch` substituindo os arquivos no diretório de armazenamento local pelos arquivos que são extraídos do archive compactado executando os seguintes comandos:

      ```
      rm -r /var/lib/icp/logging/elk-data/*
       tar -C /var/lib/icp/logging/elk-data -xzf ~/logging-backup.tar.gz --strip-components 2
      ```

4. Migre os serviços `logging`, `metering`, `monitoring` e `key-management` para seu novo nó de gerenciamento.

   1. Desative os serviços nos nós principais. Consulte Desativar serviços para obter mais informações.

      - É possível desativar os serviços por meio de seu arquivo `config.yaml`. Atualize o parâmetro `management_services`. O arquivo `config.yaml` pode ser semelhante ao conteúdo a seguir:

        ```
        management_services:
        logging: disabled
        metering: disabled
        monitoring: disabled
        key-management: disabled
        ```

      - Também é possível desativar os serviços ao executar o comando a seguir:

        ```
        docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster
        ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
        ```

2. Exclua o PersistentVolumeClaim e os PersistentVolumes do serviço `logging` com as etapas a seguir:

   - Obtenha o PersistentVolumeClaim e os PersistentVolumes do serviço `logging` executando os comandos a seguir:

     ```
     kubectl get pvc -n kube-system | grep data-logging-elk-data
     kubectl get pv | grep logging-datanode
     ```

   - Exclua o PersistentVolumeClaim e os PersistentVolumes do serviço `loggin` executando os comandos a seguir:

     ```
     kubectl delete pvc <persistent-volume-claim-name> -n kube-system
     kubectl delete pv <persistent-volume-name>
     ```

3. Ative os serviços em seus nós principais.

   - É possível ativar os serviços a partir de seu arquivo `config.yaml`. Atualize o parâmetro `management_services`. O arquivo `config.yaml` pode ser semelhante ao conteúdo a seguir:

     ```
     management_services:
     logging: enabled
     metering: enabled
     monitoring: enabled
     key-management: enabled
     ```

   - Também é possível ativar os serviços executando o comando a seguir:

     ```
     docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster
     ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
     ```

5. Verifique se seus pods são transferidos para o novo nó de gerenciamento executando o seguinte comando:

   ```
   kubectl get pods -n kube-system -o custom-
   columns=Name:.metadata.name,STATUS:.status.phase,NODE:.spec.nodeName
   ```

   Sua saída pode ser semelhante às informações a seguir:

   ```
   logging-elk-client-74d744cdc6-l7ds5                          Running
   <new-node-name>
   logging-elk-data-0                                           Running
   <new-node-name>
   metering-*                                                   Running
   <new-node-name>
   monitoring-*                                                 Running
   <new-node-name>
   key-management-*                                             Running
   <new-node-name>
   ```

Suas funções de nó de gerenciamento são transferidas para um novo nó.

Transferindo funções de proxy

1. Prepare seu novo nó do proxy para a instalação. Para obter mais informações, verifique Preparando o novo nó para instalação

2. Inclua seu novo nó do proxy para o cluster do IBM Cloud Private executando o seguinte comando:

   ```
   docker run -e LICENSE=accept --net=host \
   -v "$(pwd)":/installer/cluster \
   ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee proxy -l \
   ip_address_proxynode1,ip_address_proxynode2
   ```

3. Remova o rótulo `proxy` do nó principal.

   1. Obtenha o rótulo para seu nó principal executando o comando a seguir:

      ```
      kubectl get nodes <master.node.name> --show-labels
      ```

   2. Remova os rótulos da função de `proxy` dos nós principais executando o seguinte comando:

      ```
      kubectl label nodes <master.node.name> proxy- node-role.kubernetes.io/proxy-
      ```

   3. Verifique se os rótulos da função de `proxy` foram removidos do nó principal executando o seguinte comando:

```
kubectl get nodes <master.node.name> --show-labels
```

4. Exclua e transfira os pods a seguir para o novo nó de gerenciamento:

   - nginx-ingress-controller
   - default-http-backend
   - istio-egressgateway
   - istio-ingressgateway

   Para excluir os pods, execute os comandos a seguir:

```
kubectl get pods --all-namespaces -o=custom-
columns=NAME:.metadata.name,NAMESPACE:.metadata.namespace | grep -E 'nginx-ingress-
controller|default-http-backend|istio-ingressgateway|istio-egressgateway' | while read pods; do
pods_name=$(echo $pods | awk '{print $1}');
pods_namespace=$(echo $pods | awk '{print $2}');
echo "-------------------------------------------------------------------------"
echo "|                      Pods: ${pods_name}"
echo "|                 Namespace: ${pods_namespace}"
echo "-------------------------------------------------------------------------"
echo "Deleting proxy pod ${pods_name} ..."
kubectl delete pods ${pods_name} -n ${pods_namespace} --grace-period=0 --force &>/dev/null
done
```

   Nota: se o pod `k8s-proxy-vip` existir em seu nó principal, deve-se mover o pod para o arquivo `/etc/cfc/pods/k8s-proxy-vip.json` para o novo nó do proxy. Verifique se o pod `k8s-proxy-vip` existe e mova-o executando os seguintes comandos:

```
kubectl get k8s-proxy-vip -n kube-system
scp etc/cfc/pods/k8s-proxy-vip.json <node_user>@<node_ip>:etc/cfc/pods/k8s-proxy-vip.json
```

   O `kube-scheduler` replaneja os pods para o novo nó do proxy.

5. Verifique se seus pods são transferidos para o novo nó do proxy executando o seguinte comando:

```
 kubectl get pods -n kube-system -o custom-
columns=Name:.metadata.name,STATUS:.status.phase,NODE:.spec.nodeName
```

   Sua saída pode ser semelhante às informações a seguir:

```
 nginx-ingress-controller                          Running                     <new-node-
name>
 default-http-backend                              Running                     <new-node-
name>
```

As funções do nó do proxy são transferidas para um novo nó.

Transferindo funções de consultor de vulnerabilidade

Importante: os PersistentVolumes devem ser transferidos para um novo nó porque alguns pods possuem dados de armazenamento local no nó de gerenciamento.

1. Remova o rótulo `va` do nó principal:

   1. Obtenha o rótulo para seu nó principal executando o comando a seguir:

```
kubectl get nodes <master.node.name> --show-labels
```

   2. Remova o rótulo da função `va` executando o seguinte comando:

```
kubectl label nodes <master.node.name> va- node-role.kubernetes.io/va-
```

   3. Verifique se o rótulo da função `va` foi removido do nó principal executando o seguinte comando:

```
kubectl get nodes <master.node.name> --show-labels
```

2. Inclua seu novo nó do VA com as etapas a seguir:

   1. Prepare o novo nó do VA para a instalação. Para obter mais informações, consulte Preparando o novo nó para a instalação.

2. Remova o endereço IP do nó principal do arquivo host no nó do VA. Para acessar o arquivo host, execute o comando a seguir:

```
vi /etc/hosts
```

3. Inclua um novo consultor de vulnerabilidade no cluster do IBM Cloud Private executando o seguinte comando:

```
docker run --rm -t -e LICENSE=accept --net=host -v \
$(pwd):/installer/cluster ibmcom/icp-inception-$(uname -m | sed
's/x86_64/amd64/g'):3.2.0-ee va -l \
ip_address_vanode1,ip_address_vanode2
```

3. Mova os dados de armazenamento local do nó principal para o novo nó do VA. Transfira os dados `minio`, `zookeeper` e `kafka` com as seguintes etapas:

1. Faça backup dos dados `minio` e `kafka` criando um archive compactado dos dados a partir do diretório de armazenamento executando os seguintes comandos:

```
tar -czf ~/minio-backup.tar.gz /var/lib/icp/va/minio/*
tar -czf ~/zookeeper-backup.tar.gz /var/lib/icp/va/zookeeper/*
tar -czf ~/kafka-backup.tar.gz /var/lib/icp/va/kafka/*
```

2. Use sua cópia de serviço para transferir o arquivo de dados para o novo nó principal executando os seguintes comandos:

```
scp ~/minio-backup.tar.gz <node_user>@<node_ip>:~/minio-backup.tar.gz
scp ~/zookeeper-backup.tar.gz <node_user>@<node_ip>:~/zookeeper-backup.tar.gz
scp ~/minio-backup.tar.gz <node_user>@<node_ip>:~/minio-backup.tar.gz
```

3. Restaure os dados `minio` e `kafka` substituindo os arquivos no diretório de armazenamento local pelos arquivos extraídos do archive compactado executando os seguintes comandos:

```
rm -r /var/lib/icp/va/minio/*
rm -r /var/lib/icp/va/zookeeper/*
rm -r /var/lib/icp/va/kafka/*
tar -C /var/lib/icp/va/minio -xzf ~/minio-backup.tar.gz --strip-components 2
tar -C /var/lib/icp/va/zookeeper -xzf ~/zookeeper-backup.tar.gz --strip-components 2
tar -C /var/lib/icp/va/kafka -xzf ~/kafka-backup.tar.gz --strip-components 2
```

4. Migre os serviços VA para seu novo nó de gerenciamento com as etapas a seguir:

1. Desative os serviços do VA em seus nós principais. Consulte Desativar serviços.

- É possível desativar os serviços VA a partir de seu arquivo `config.yaml`. Atualize o parâmetro `management_services`. O arquivo `config.yaml` pode ser semelhante ao conteúdo a seguir:

```
management_services:
vunerability=advisor: disabled
```

- Também é possível desativar o VA executando o comando a seguir:

```
docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster
ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
```

2. Exclua os PersistentVolumeClaims e os PersistentVolumes dos serviços `minio` e `kafka` com as etapas a seguir:

- Obtenha os PersistentVolumeClaims e PersistentVolumes de `minio` e `kafka` executando os comandos a seguir:

```
kubectl get pvc -n kube-system | grep minio
kubectl get pv | grep minio
kubectl get pvc -n kube-system | grep zookeeper
kubectl get pv | grep zookeeper
kubectl get pvc -n kube-system | grep kafka
kubectl get pv | grep kafka
```

- Exclua os PersistentVolumeClaims e os PersistentVolumes dos serviços `minio` e `kafka` executando os comandos a seguir:

```
kubectl delete pvc <persistent-volume-claim-name> -n kube-system
kubectl delete pv <persistent-volume-name>
```

3. Ative os serviços em seus nós principais.

- É possível ativar os serviços a partir de seu arquivo `config.yaml`. Atualize o parâmetro `management_services`. O arquivo `config.yaml` pode ser semelhante ao conteúdo a seguir:

```
   management_services:
vulnerability-advisor: enabled
```

- Também é possível ativar os serviços executando o comando a seguir:

```
   docker run --rm -t -e LICENSE=accept --net=host -v $(pwd):/installer/cluster
ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee addon
```

5. Verifique se seus pods são transferidos para o novo nó do VA executando o seguinte comando:

```
 kubectl get pods -n kube-system -o custom-
columns=Name:.metadata.name,STATUS:.status.phase,NODE:.spec.nodeName
```

   Sua saída pode ser semelhante às informações a seguir:

```
 vulnerability-advisor-compliance-annotator                          Running
<new-node-name>
```

As funções do nó do Vulnerability Advisor foram transferidas para um novo nó.

Erros de instalação com o SELinux ativado

Se você tiver o Security-Enhanced Linux (SELinux) ativado, poderá encontrar os erros a seguir quando estiver instalando o IBM Cloud Private:

- Permissão negada ao executar o Docker
- Falha na instalação ao Copiar o hyperkube

Sintoma - permissão negada ao executar o Docker

Quando o SELinux está ativado, você encontra um erro "permissão negada" ao executar o comando `docker run`. O comando e o erro resultante podem ser semelhantes ao comando e à saída a seguir:

```
# sudo docker run  -v $(pwd):/data:z  -e LICENSE=accept ibmcom/icp-inception-amd64:3.1.2-ee cp -r
cluster /data
standard_init_linux.go:190: exec user process caused "permission denied"
```

Causas

O Docker não inclui a configuração de contexto de segurança do SELinux correta.

Resolvendo o problema

Execute o comando a seguir para configurar o contexto de segurança do SELinux esperado:

```
/usr/sbin/restorecon -R /usr/bin/docker* /var/run/docker.sock /var/run/docker.pid /etc/docker
/usr/lib/systemd/system/docker.service
```

Sintoma - falha na instalação ao copiar o hyperkube

Quando o SELinux é ativado, a instalação do IBM Cloud Private falha ao executar o comando a seguir:

```
sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster:z ibmcom/icp-
inception-amd64:3.1.2-ee install
```

O log de erro resultante inclui os detalhes a seguir:

```
TASK [kubelet-config : Copying hyperkube onto operating system] ****************
FAILED - RETRYING: Copying hyperkube onto operating system (3 retries left).
FAILED - RETRYING: Copying hyperkube onto operating system (2 retries left).
```

```
FAILED - RETRYING: Copying hyperkube onto operating system (1 retries left).
fatal: [172.16.181.137]: FAILED! => changed=true
  attempts: 3
  cmd: docker run --rm -v /opt/kubernetes/:/data:z mycluster.icp:8500/ibmcom/hyperkube:v1.12.4-ee sh
-c 'cp -f /hyperkube /data/'
  delta: '0:00:02.413875'
  end: '2019-03-20 07:55:32.436609'
  msg: non-zero return code
  rc: 127
  start: '2019-03-20 07:55:30.022734'
  stderr: 'cp: error while loading shared libraries: cannot restore segment prot after reloc:
Permission denied'
  stderr_lines: <omitted>
  stdout: ''
  stdout_lines: <omitted>
```

Causas

O contêiner SELinux ativado não está no nível de versão necessário.

Resolvendo o problema

Faça upgrade de sua versão de contêiner do SELinux. Verifique a versão do contêiner do SELinux que você ativou executando o comando a seguir:

```
rpm -q container-selinux
```

Se a sua versão do contêiner SELinux não for `container-selinux-2.68-1.el7.noarch`, faça upgrade de seu contêiner para esta versão. É possível fazer download do pacote de instalação do RPM `container-selinux-2.68-1.el7.noarch.rpm` para a versão do SELinux a partir do website do CentOS Project Abre em uma nova guia.

Depois que o pacote for transferido por download, execute o comando a seguir para fazer upgrade de seu contêiner do SELinux:

```
rpm -e container-selinux
rpm -ivh container-selinux-2.68-1.el7.noarch.rpm
```

O pacote RPM container-selinux executa o processo `restorecon -R -v /var/lib/docker` em uma nova instalação. Esse processo pode demorar alguns minutos para ser concluído.

Resolução de problemas do IAM

Resolva problemas do Identity and Access Management (IAM).

- Efetuar login
- LDAP
- Problemas do pod

Login

Revisar problemas de login encontrados frequentemente.

- O administrador do cluster não pode efetuar login na console de gerenciamento
- Não é possível autenticar para o kubectl usando a CLI no Windows
- Resultados de login do Docker em aviso de senha não criptografada

O administrador do cluster não pode efetuar login na console de gerenciamento

Depois de configurar uma conexão LDAP, não será possível efetuar login em sua console de gerenciamento do cluster do IBM® Cloud Private usando o nome do usuário e a senha do administrador de cluster padrão.

Causas

Os parâmetros LDAP que você especificou ao configurar a conexão LDAP podem não estar corretos.

Resolvendo o problema

Remova a configuração LDAP atual.

A configuração LDAP é armazenada no arquivo `/config/configDropins/defaults/ldap-<LDAP_connection_name>.xml` no contêiner `platform-auth`. Para remover a configuração LDAP incorreta, deve-se excluir o arquivo ou movê-lo para outro local. Em seguida, é possível efetuar login no console e configurar a conexão LDAP novamente com os parâmetros corretos.

Para remover a configuração LDAP atual, execute os comandos a seguir com acesso raiz em seu nó principal:

1. Obtenha o ID do contêiner `platform-auth`:

   ```
   docker ps | grep platform-auth
   ```

   Exemplo de comando e saída:

   ```
   root@master:/opt/icp/cluster# docker ps | grep platform-auth
   d588a5b951b4        ibmcom/icp-platform-auth                    "/usr/bin/superv
   is..."   5 days ago          Up 5 days                      k8s_platfor            m-
   auth-service_auth-idp-ln0s6_kube-system_d5f4fe3c-c60e-11e7-8ea2-005056a85e40_0
   ```

2. Acesse o shell dentro do contêiner:

   ```
   docker exec -it <container ID> /bin/bash
   ```

   Exemplo de comando e saída:

   ```
   root@master:/opt/icp/cluster# docker exec -it d588a5b951b4 /bin/bash
   bash-4.3#
   ```

3. Anote o nome do arquivo de configurações LDAP localizado no diretório `/config/configDropins/defaults/`. Por exemplo, o nome do arquivo será `ldap-openLDAP.xml` se o nome de conexão LDAP for `openLDAP`.

4. Mova o arquivo de configurações de LDAP do diretório `/config/configDropins/defaults/`. Por exemplo, você pode mover o arquivo para o diretório `/config/configDropins/`:

   ```
   mv /config/configDropins/defaults/ldap-<LDAP_connection_name>.xml /config/configDropins/
   mv /config/configDropins/defaults/federated.xml /config/configDropins/
   ```

   Nota: em vez de mover o arquivo, é possível excluir o arquivo de configurações LDAP do diretório `/config/configDropins/defaults/`. No entanto, mover o arquivo assegura que você tenha um backup do arquivo de configuração, que pode ser usado para resolução de problemas. Também é possível corrigir as configurações no arquivo e reutilizá-lo para configurar a conexão LDAP. Se você está reutilizando o arquivo, deve-se mover o arquivo corrigido de volta para o diretório `/config/configDropins/defaults/`.

5. Aguarde um minuto e efetue login como um administrador do cluster.

6. Configure sua conexão LDAP com os parâmetros de configuração apropriados. Para obter mais informações sobre como configurar uma conexão LDAP, consulte Configurando a conexão LDAP.

Não é possível autenticar no kubectl usando a CLI no Windows

A conexão do cluster do IBM® Cloud Private com a linha de comandos kubectl usando a linha de comandos do IBM Cloud Private falha no Windows™.

Sintomas

Quando o comando `cloudctl login` é executado e o erro a seguir é exibido:

```
Unable to rename file to bat extension: C:\Users\<your_account>`\.cloudctl\clusters\mycluster\kube-
config
```

Em que `<your_account>` é o nome da conta do Windows.

Causas

O Windows não está convertendo corretamente o arquivo `kube-config` que o comando transfere por download para um arquivo `.bat`.

Resolvendo o problema

1. Mude o nome do arquivo de `C:\Users\<your_account>\.cloudctl\clusters\mycluster\kube-config` para `C:\Users\<your_account>\.bluemix\plugins\icp\clusters\mycluster\kube-config.bat`.
2. Execute o arquivo `C:\Users\<your_account>\.cloudctl\clusters\mycluster\kube-config.bat`. É possível executar comandos kubectl em seu cluster.

O login do Docker resulta em aviso de senha não criptografada

Depois de efetuar login em seu registro de imagem privado com o comando de login do Docker, um aviso é exibido indicando que sua senha é armazenada não criptografada.

Sintomas

Uma mensagem de aviso é exibida ao efetuar login em seu registro de imagem privado com o comando a seguir:

```
docker login <cluster_CA_domain>:8500
```

Nesse comando, o `<cluster_CA_domain>` é o domínio de autoridade de certificação (CA) que foi configurado no arquivo `config.yaml` durante a instalação.

A saída de comando inclui o aviso a seguir:

```
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

Causas

Por padrão, o Docker armazena a senha de login não criptografada dentro do arquivo de configuração do Docker `/root/.docker/config.json`. Esse é o comportamento padrão do Docker.

Resolvendo o problema

É possível armazenar suas credenciais do usuário em um armazenamento de credenciais externo em vez de no arquivo de configuração do Docker. É mais seguro armazenar suas credenciais em um armazenamento de credenciais do que armazenar as credenciais no arquivo de configuração do Docker. Para obter mais informações, consulte a Documentação do armazenamento de credenciais do Docker Abre em uma nova guia.

LDAP

Revise os problemas de configuração do LDAP (Lightweight Directory Access Protocol) encontrados com frequência.

- Ativar a depuração para problemas de autenticação do usuário
- Resolução de problemas de configuração do LDAP
- Configurando o LDAP sobre SSL
- Resolução de problemas de procura de usuários e de grupos de usuários

Ativar a depuração para problemas de autenticação do usuário

Ative a depuração para `platform-auth-service` Liberty para obter logs de rastreio para depuração de problemas de autenticação do usuário.

O parâmetro de configuração é `LIBERTY_DEBUG_ENABLED`. O valor padrão é `LIBERTY_DEBUG_ENABLED: false`.

Configure o valor do parâmetro `LIBERTY_DEBUG_ENABLED` como `true` ou `false`.

A seguir estão as etapas para mudar o valor:

Mudando o valor de parâmetro usando kubectl

1. Configure a CLI do `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Edite o ConfigMap `platform-auth-idp`.

   ```
   kubectl -n kube-system edit configmap platform-auth-idp
   ```

3. Configure `LIBERTY_DEBUG_ENABLED` como `true` ou `false`.
4. Salve e feche o ConfigMap.
5. Reinicie os pods `auth-idp`

   ```
   kubectl -n kube-system delete pod -l k8s-app=auth-idp
   ```

6. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp`. O status deve ser mostrado como `4/4 Running` para todos os pods.

   ```
   kubectl -n kube-system get pods | grep auth-idp
   ```

Mudando os valores de parâmetro usando a console de gerenciamento

1. Efetue login no console como um usuário com acesso de administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **ConfigMaps.**
3. Procure por `platform-auth-idp`.
4. Clique em **...** > `Editar`.
5. Mude o valor do parâmetro `LIBERTY_DEBUG_ENABLED` para `true` ou `false`.
6. Clique em **Enviar**.
7. No menu de navegação, clique em **Cargas de trabalho > DaemonSets**.
8. Localize `auth-idp`.
9. Clique em **...** > `Editar`. Uma janela `Editar DaemonSet` é exibida.
10. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é para recarregar os pods `auth-idp` com os valores de ConfigMap mais recentes.
11. Clique em `auth-idp`.
12. Aguarde um pouco. Em seguida, verifique o status dos pods `auth-idp` na área de janela **Pods**. O status de todos os pods deve ser mostrado como `4/4` sob o nome do campo **Pronto**.

Obtendo os logs

Siga estas etapas para obter os logs de configuração e de rastreio:

1. Efetue login no seu cluster do IBM® Cloud Private usando a console de gerenciamento ou a CLI do IBM Cloud Private algumas vezes para gerar logs. Para obter mais informações sobre como instalar a CLI do IBM Cloud Private, consulte Instalando a CLI do IBM® Cloud Private.
2. Efetue login no nó principal usando o shell seguro (SSH).
3. Obtenha o ID do contêiner `platform-auth` .

   ```
   docker ps | grep platform-auth
   ```

4. Copie a configuração e os logs para as pastas no nó principal.

   ```
   docker cp <container-id>:/config/configDropins/defaults auth-service_config
   docker cp <container-id>:/logs auth-service_logs
   ```

5. Repita da etapa 2 à etapa 4 para todos os outros nós principais, se você tiver múltiplos nós principais em seu cluster.
6. Obtenha as pastas de configuração (`auth-service_config`) e de logs (`auth-service_logs`) de todos os nós.

Resolução de problemas de configuração do LDAP

Use a ferramenta de linha de comandos `ldapsearch` para solucionar problemas de configuração do LDAP (Lightweight Directory Access Protocol).

Instale o `ldapsearch`

Instale o programa `ldapsearch`.

No Ubuntu, execute o comando a seguir:

```
sudo apt-get install ldap-utils
```

No Red Hat Enterprise Linux (RHEL), execute o seguinte comando:

```
sudo yum install openldap-clients
```

Testar conexão LDAP

Para testar sua conexão LDAP, execute o comando a seguir:

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub
```

A seguir estão as descrições de parâmetros:

- `<LDAP_URL>` é a URL do servidor LDAP. Por exemplo, `ldap://<LDAP server domain name or IP address>:<port>` ou `ldaps://<LDAP server domain name>:<port>`. O número da porta padrão é 389 para o protocolo LDAP e 636 para o protocolo LDAP sobre Secure Sockets Layer (LDAPS).
- `<LDAP_BASEDN>` é o nome distinto (DN) LDAP da base de procura. Por exemplo, `dc=abc, dc=com`.
- `<LDAP_BINDDN>` é o usuário LDAP que tem permissão para procurar o DN base. Por exemplo, `cn=admin, dc = abc, dc=com`.
- `<LDAP_BINDPASSWORD>` é a senha do usuário mencionado no DN de ligação.

Comandos de Exemplo

```
ldapsearch -x -H "ldap://<hostname or IP address>:389" -b "o=abc.com" -s sub
```

```
ldapsearch -x -H "ldap://<hostname or IP address>:389" -b "dc=abc,dc=com" -D
"cn=admin,dc=abc,dc=com" -w "password" -s sub
```

Validar filtros LDAP

Crie uma sequência de procura com base nos filtros LDAP para recuperar dados de seu servidor LDAP. Quando os resultados da procura mostram uma ou mais entradas LDAP, a configuração do filtro LDAP está correta. Quando os resultados da procura não mostram nenhuma entrada, o filtro LDAP não está correto ou não é compatível com o tipo de servidor LDAP.

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub
"<Search string>"
```

A seguir estão as descrições de parâmetros:

- `<LDAP_URL>` é a URL do servidor LDAP. Por exemplo, `ldap://<LDAP server domain name or IP address>:<port>` ou `ldaps://<LDAP server domain name>:<port>`. O número da porta padrão é 389 para o protocolo LDAP e 636 para o protocolo LDAPS.
- `<LDAP_BASEDN>` é o DN LDAP da base de procura. Por exemplo, `dc=abc, dc=com`.
- `<LDAP_BINDDN>` é o usuário LDAP que tem permissão para procurar o DN base. Por exemplo, `cn=admin, dc = abc, dc=com`.
- `<LDAP_BINDPASSWORD>` é a senha do usuário mencionado no DN de ligação.
- `<search string>` é a sequência de procura que é usada para procurar pelo seu servidor LDAP.

Filtros LDAP do IBM Tivoli Directory Server

Nome do Atributo	Valor Padrão
Filtro de Grupo	(&(cn=%v)(objectclass=groupOfUniqueNames))
Mapa de ID do	* :cn
Mapa de ID do Membro do	groupOfUniqueNames:uniqueMember
Filtro de usuário	(&(emailAddress=%v)(objectclass=person))

Nome do Atributo	Valor Padrão
Mapa de ID do	* :emailAddress

- Comando de exemplo para validar o filtro de grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(cn=*)(objectclass=groupOfUniqueNames))"
```

- Comando de exemplo para validar o mapa de ID do grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(cn=*))"
```

- Comando de exemplo para validar o mapa de ID do membro do grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=groupOfUniqueNames)(uniqueMember=*))"
```

- Comando de exemplo para validar filtro do usuário

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(emailAddress=*)(objectclass=person))"
```

- Comando de exemplo para validar o mapa de ID do usuário

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(emailAddress=*))"
```

Filtros LDAP do Microsoft Active Directory

Nome do Atributo	Valor Padrão
Filtro de Grupo	(&(cn=%v)(objectcategory=group))
Mapa de ID do	* :cn
Mapa de ID do Membro do	memberOf:member
Filtro de usuário	(&(sAMAccountName=%v)(objectcategory=user))
Mapa de ID do	user:sAMAccountName

- Comando de exemplo para validar o filtro de grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(cn=*)(objectcategory=group))"
```

- Comando de exemplo para validar o mapa de ID do grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectcategory=*)(cn=*))"
```

- Comando de exemplo para validar o mapa de ID do membro do grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectcategory=*)(member=*))"
```

- Comando de exemplo para validar filtro do usuário

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(sAMAccountName=*)(objectcategory=user))"
```

- Comando de exemplo para validar o mapa de ID do usuário

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectcategory=user)(sAMAccountName=*))"
```

Filtros LDAP do servidor Custom (OpenLDAP)

Nome do Atributo	Valor Padrão
Filtro de Grupo	(&(cn=%v)(objectclass=groupOfUniqueNames))
Mapa de ID do	* :cn
Mapa de ID do Membro do	groupOfUniqueNames:uniqueMember

Nome do Atributo	Valor Padrão
Filtro de usuário	(&(uid=%v)(objectclass=person))
Mapa de ID do	* :uid

- Comando de exemplo para validar o filtro de grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(cn=*)(objectclass=groupOfUniqueNames))"
```

- Comando de exemplo para validar o mapa de ID do grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(cn=*))"
```

- Comando de exemplo para validar o mapa de ID do membro do grupo

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=groupOfUniqueNames)(uniqueMember=*))"
```

- Comando de exemplo para validar filtro do usuário

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(uid=*)(objectclass=person))"
```

- Comando de exemplo para validar o mapa de ID do usuário

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(uid=*))"
```

Resolver problemas comuns

Não é possível efetuar login como um usuário LDAP se você usou o LDAPS para configurar sua

conexão LDAP.

Talvez você não consiga efetuar login como um usuário LDAP, mesmo quando o teste de conexão e a configuração do LDAP são bem-sucedidos.

Causas

- O certificado do servidor LDAP não foi importado no IBM® Cloud Private.
- Você usou o endereço IP em vez do nome do host do servidor LDAP na URL do LDAP.
- Você usou o nome do host do servidor LDAP na URL do LDAP. No entanto, o nome do host não é atingível. Esse problema pode ocorrer porque as entradas corretas do servidor DNS não foram incluídas durante a instalação do IBM Cloud Private.

Resolução

- Primeiramente, assegure-se de usar o nome do host do servidor LDAP na URL do LDAP. Em seguida, importe o certificado do servidor LDAP.
- Inclua o nome do host do servidor LDAP no arquivo `/etc/hosts` no nó principal ou no contêiner `platform-auth-service` do pod `auth-idp`.

Não é possível efetuar login como um usuário LDAP devido a credenciais do usuário inválidas.

Você vê um erro que indica um nome de usuário ou senha inválida.

Causas

- O nome de usuário não é o mesmo que o valor do atributo de filtro do mapa de ID do USER.
- A senha de usuário contém caracteres especiais baseados em XML, como `$ < > & ' "`.

Resolução

- Assegure-se de inserir o nome de usuário correto. O nome de usuário deve ser igual ao valor do atributo de filtro do mapa de ID do USER. O nome de usuário faz distinção entre maiúsculas e minúsculas.

  Considere os seguintes parâmetros usados para uma configuração LDAP:

  ```
  LDAP user details:
  dn: uid=testuser,ou=people,dc=abc,dc=com
  objectClass: inetOrgPerson
  objectClass: organizationalPerson
  objectClass: person
  objectClass: top
  cn: TestUser
  givenName: TestUser
  sn: SN
  uid: testuser
  userPassword: testuser
  mail: testuser@abc.com
  ```

  Se `*:uid` for usado como o filtro de mapa de ID do USER, o `testuser` deverá ser usado como o nome de usuário ao efetuar login.

- Tente remover os caracteres especiais de sua senha.

Não é possível procurar por usuários ou grupos enquanto você cria uma equipe.

Causa

Você usou uma sequência de procura inválida.

Resolução

Deve-se usar o valor do atributo `cn` ou o atributo de usuário ou grupo, como `uid` ou `emailaddress`, que é usado na configuração do LDAP.

Considere os seguintes parâmetros usados para uma configuração LDAP:

```
dn: uid=testuser,ou=people,dc=abc,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: TestUser
givenName: TestUser
sn: SN
uid: testuser
userPassword: testuser
mail: testuser@abc.com
```

O mapa do ID do USER que é usado é `*:uid`.

Os valores válidos que podem ser utilizados para procurar por um usuário são os seguintes:

- `TestUser` (valor cn)
- `testuser` (valor uid)

Configurando o LDAP sobre SSL

É possível proteger a conexão Lightweight Directory Access Protocol (LDAP) usando SSL (Secure Sockets Layer).

Se o LDAP sobre SSL (LDAPS) não puder ser configurado automaticamente em seu cluster, conclua estas etapas para configurar manualmente a conexão LDAPS.

Deve-se preparar o cluster do IBM Cloud Private para se conectar ao diretório LDAPS.

Antes de iniciar, deve-se importar o certificado SSL público ou privado que você usou para configurar seu diretório LDAPS.

Recuperando o certificado SSL

Se você tiver o certificado SSL de seu servidor LDAP, continue com Codificando o certificado SSL.

Se você não tiver o certificado SSL de seu servidor LDAP, conclua as etapas a seguir para recuperar o certificado SSL:

Nota: é necessário o programa `ldapsearch` para executar esses comandos. É possível instalá-lo executando `apt install ldap-utils` no Ubuntu e `yum install openldap-clients` no Red Hat Enterprise Linux (RHEL).

1. Certifique-se de que nenhum certificado SSL esteja no diretório `/etc/openldap/cacerts`.
2. Execute o comando `ldapsearch` a seguir para recuperar o nome do certificado:

   ```
   ldapsearch -H <LDAP server URL> -d 1  -b <searchbase> -D "" -s base "(<filter>)"
   ```

   em que

   - **URL do servidor LDAP** é o nome de domínio do diretório e a porta do LDAP. Formato: `ldaps://<LDAP server domain name or IP address>:<port>`.
   - **-d** é o nível de depuração.
   - **-b** é a base de procura.
   - **-D** é o DN de ligação. Esse parâmetro é opcional.
   - **-s** é o escopo da procura.
   - **filter** é o filtro LDAP. Filtro padrão: `(objectClass=*)`.

   O seguinte é um exemplo de comando e de saída:

   ```
     $ ldapsearch -H ldaps://corp.example.com:636 -d 1  -b o=example.com -D "" -s base "
   (objectclass=*)"
     ldap_url_parse_ext(ldaps://corp.example.com:636)
     ldap_create
     ldap_url_parse_ext(ldaps://corp.example.com:636/??base)
     ldap_sasl_bind
     ldap_send_initial_request
     ldap_new_connection 1 1 0
     ldap_int_open_connection
     ldap_connect_to_host: TCP corp.example.com:636
     ldap_new_socket: 3
     ldap_prepare_socket: 3
     ldap_connect_to_host: Trying 9.17.186.253:636
     ldap_pvt_connect: fd: 3 tm: -1 async: 0
     attempting to connect:
     connect success
     TLS: certdb config: configDir='/etc/openldap' tokenDescription='ldap(0)' certPrefix='cacerts'
   keyPrefix='cacerts' flags=readOnly
     TLS: cannot open certdb '/etc/openldap', error -8018:Unknown PKCS #11 error.
     TLS: could not get info about the CA certificate directory /etc/openldap/cacerts - error
   -5950:File not found.
     TLS: certificate [CN=DigiCert Global Root G2,OU=www.digicert.com,O=DigiCert Inc,C=US] is not
   valid - error -8172:Peer's certificate issuer has been marked as not trusted by the user..
     TLS: error: connect - force handshake failure: errno 2 - moznss error -8172
     TLS: can't connect: TLS error -8172:Peer's certificate issuer has been marked as not trusted
   by the user..
     ldap_err2string
     ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
   ```

   As informações de certificado estão na seção `TLS: certificate [CN=DigiCert Global Root G2,OU=www.digicert.com,O=DigiCert Inc,C=US] is not valid - error -8172:Peer's certificate issuer has been marked as not trusted by the user..` da saída.

3. Localize o certificado em seu navegador da web, exporte-o para um arquivo em um formato PEM e salve o arquivo PEM com uma extensão `.crt`.
   **
4. Copie o arquivo `.crt` no nó principal de seu cluster do IBM Cloud Private.

Preparando-se para uma conexão LDAPS única

Se você estiver configurando uma conexão LDAPS única, conclua as etapas na seção Codificando o certificado SSL.

Codificando o certificado SSL

Conclua as etapas a seguir para codificar o certificado em base64:

1. Efetue logon no nó principal do cluster do IBM Cloud Private.
2. Se o seu servidor LDAP usa certificados em cadeia (certificados de CA raiz e intermediários), combine os certificados em um arquivo antes de codificar. Use o seguinte comando para combinar os certificados em um arquivo:

   ```
   cat < first_cert.pem> < second_cert.pem> .. < n_cert.pem> > combined_cert.pem
   ```

3. Codifique seu certificado em base64.

   ```
   cat <LDAPS SSL certificate name>.crt | base64 -w 0
   ```

   A saída se assemelha ao código a seguir:

   ```
   LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdDRENDQS9DZ
   0F3SUJBZ0lRS3k1dTZ0bDFObXdVaW03Ym8zeU1CekFOQmdrcWhraUc5
   ...
   lDOHg0OU9oZ1E9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0KDQo=
   ```

Em seguida, continue com Preparando seu cluster.

Preparando-se para múltiplas conexões LDAPS

Se você estiver configurando múltiplas conexões LDAPS, primeiramente verifique se já há um certificado na seção "dados" > "certificado" de `platform-auth-ldaps-ca-cert`. Caso haja um certificado, conclua as etapas nas seções a seguir:

- Recupere o certificado atual
- Codifique o certificado SSL

Se nenhum certificado existir, conclua as etapas em Codificando o certificado SSL.

Recupere o certificado atual

É possível recuperar o certificado usando o console de gerenciamento ou usando a interface da linha de comandos (CLI).

Se você estiver usando o console de gerenciamento, siga essas etapas:

1. Efetue login na console de gerenciamento como um administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **Segredos**.
3. Localize `platform-auth-ldaps-ca-cert` e clique em **AÇÃO** > **Editar**. Uma janela**Editar segredo** é exibida.

4. Copie o valor do certificado codificado em base64 da seção "data" > "certificate".

   ```
   "data": {
   "certificate": "LS0tLS1...ASDFASDo="
   },
   ```

5. Converta o certificado codificado existente e salve em um arquivo.

   ```
   echo "<copied_cert_value>" | base64 --decode > existing_cert.pem
   ```

Se você estiver usando a CLI, siga essas etapas:

1. Acesse a CLI de seu nó principal. É necessário o `kubectl`, a ferramenta de linha de comandos Kubernetes para concluir as tarefas a seguir. Para obter informações adicionais sobre como instalar o `kubectl`, consulte Instalando a CLI do Kubernetes (kubectl).
2. Obtenha o certificado base64.

   ```
   kubectl -n kube-system get secret platform-auth-ldaps-ca-cert -o "jsonpath=
   {.data['certificate']}" | base64 --decode > existing_cert.pem
   ```

Em seguida, continue com Codificar o certificado SSL.

Codifique o certificado SSL

Conclua as etapas a seguir para codificar o certificado em base64:

1. Efetue logon no nó principal do cluster do IBM Cloud Private.

2. Se o seu servidor LDAP usa certificados em cadeia (certificados de CA raiz e intermediários), combine os certificados em um arquivo antes de codificar. Use o comando a seguir para anexar os certificados a um arquivo:

   ```
   cat < first_cert.pem> < second_cert.pem> .. < n_cert.pem> > combined_cert.pem
   ```

3. Inclua o certificado existente no novo certificado ou no certificado combinado se você usar certificados em cadeia.

   ```
   cat existing_cert.pem <new_cert.pem or combined_cert.pem> > final_combined_cert.pem
   ```

4. Codifique seu certificado em base64.

   ```
   cat <LDAPS SSL certificate name>.crt | base64 -w 0
   ```

   A saída se assemelha ao código a seguir:

   ```
   LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdDRENDQS9DZ
   0F3SUJBZ0lRS3k1dTZ0bDFObXdVaW03Ym8zeU1CekFOQmdrcWhraUc5
   ...
   lDOHg0OU9oZ1E9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0KDQo=
   ```

Em seguida, conclua as etapas na seção Preparando seu cluster.

Preparando seu Cluster

É possível preparar seu cluster usando o console de gerenciamento ou usando a CLI.

Se você estiver usando o console de gerenciamento, siga essas etapas:

1. Efetue login na console de gerenciamento como um administrador de cluster.
2. No menu de navegação, clique em **Configuração** > **Segredos**.
3. Localize `platform-auth-ldaps-ca-cert` e clique em **AÇÃO** > **Editar**. Uma janela**Editar segredo** é exibida.

4. Cole o certificado base64 na seção a seguir:

   ```
   "data": {
    "certificate": ""
   },
   ```

   A seção atualizada é semelhante ao texto a seguir:

   ```
   "data": {
    "certificate": "LS0tLS1<very_long_base64_string>ASDFASDo="
   },
   ```

5. Clique em **Enviar**.
6. No menu de navegação, clique em **Cargas de trabalho** > **DaemonSets**.
7. Localize `auth-idp` e clique em **AÇÃO** > **Editar**. Uma janela **Editar DaemonSet** é exibida.
8. Clique em **Enviar** sem fazer nenhuma mudança. Esta etapa é recarregar o pod `auth-idp` com os valores de segredos e de configmap mais recentes.

9. Aguarde um minuto ou dois e, em seguida, verifique se o certificado está montado no pod.

   1. Obtenha os pods `auth-idp`.

      ```
      kubectl -n kube-system get pods | grep auth-idp
      ```

   2. Verifique se o certificado está montado no pod.

      ```
      kubectl -n kube-system exec -it auth-idp-<pod-id> -c platform-auth-service cat
      /opt/ibm/ldaps/ldaps-ca.crt
      ```

Se você estiver usando a CLI, siga essas etapas:

1. Acesse a CLI de seu nó principal. É necessário o `kubectl`, a ferramenta de linha de comandos Kubernetes para concluir as tarefas a seguir. Para obter informações adicionais sobre como instalar o `kubectl`, consulte Instalando a CLI do Kubernetes (kubectl).

2. Copie o segredo do certificado de CA LDAPS em um arquivo:

   ```
   kubectl get secret platform-auth-ldaps-ca-cert -o yaml -n kube-system > platform-auth-ldaps-ca-
   cert-secret.yaml
   ```

3. Copie e cole o certificado base64 para o arquivo `platform-auth-ldaps-ca-cert-secret.yaml`.
4. Obtenha o arquivo YAML do DaemonSet do serviço de segurança:

   ```
   kubectl -n kube-system get ds auth-idp -o yaml > auth-idp.yaml
   ```

5. Aplique as mudanças:

   ```
   kubectl apply -f platform-auth-ldaps-ca-cert-secret.yaml

   kubectl aplicar -f auth-idp.yaml
   ```

Em seguida, conclua as etapas na seção Conectando-se ao seu diretório LDAP.

Resolução de problemas de procura de usuários e de grupos de usuários

Verifique ou teste a procura de usuários e grupos usando a ferramenta de comando `ldapsearch`.

Procurar usuários

Use o comando `ldapsearch` a seguir:

```
ldapsearch -x -l <TIME_LIMIT> -z <SIZE_LIMIT> -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -w
<LDAP_BINDPASSWORD> -s sub "<search query>"
```

em que

- `<LDAP_URL>` é a URL do servidor Lightweight Directory Access Protocol (LDAP).
- `<LDAP_BASEDN>` é o DN Base LDAP.
- `<LDAP_BINDDN>` é o DN de Ligação LDAP.
- `<LDAP_BINDPASSWORD>` é a senha do DN de Ligação LDAP.
- `<TIME_LIMIT>` é o limite de tempo em segundos para a procura. O valor padrão é 5 segundos.
- `<SIZE_LIMIT>` é o limite de tamanho para procura. O valor padrão é 50 entradas.

No IBM® Cloud Private, a sequência de procura é baseada nos atributos `cn` e `User ID map` que estão configurados nos filtros de usuário de conexão LDAP.

Por exemplo, considere a configuração de filtros de usuário LDAP a seguir:

```
User filter: (&(uid=%v)(objectclass=person))
User ID map: *:uid
```

Para a configuração de exemplo, a seguir está a consulta de procura:

```
(|(&(cn=*<searchstring>*)(objectclass=person))(&(uid=*<searchstring>*)(objectclass=person)))
```

Em que o valor de nome do usuário ou de ID do usuário é `<searchstring>`. Por exemplo, `John` ou `robbie`.

A seguir está um exemplo de comando `ldapsearch`:

```
ldapsearch -x -l 5 -z 50 -H "ldap://X.X.X.X:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w
'password' -s sub "(|(&(cn=*user*)(objectclass=person))(&(uid=*user*)(objectclass=person)))"
```

Em que `<searchstring>` é `user`.

O seguinte é uma saída de amostra:

```
# extended LDIF
#
# LDAPv3
# base <dc=abc,dc=com> with scope subtree
# filter: (|(&(cn=*user*)(objectclass=person))(&(uid=*user*)(objectclass=person)))
# requesting: ALL
#

# user1, Users, abc.com
dn: uid=user1,ou=Users,dc=abc,dc=com
cn: User One
objectClass: inetOrgPerson
objectClass: person
```

```
objectClass: top
sn: One
uid: user1
userPassword:: dXNlcjE=

# user2, Users, abc.com
dn: uid=user2,ou=Users,dc=abc,dc=com
cn: User Two
objectClass: inetOrgPerson
objectClass: person
objectClass: top
sn: Two
uid: user2
userPassword:: dXNlcjI=

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Procurar grupos

Use o comando `ldapsearch` a seguir:

```
ldapsearch -x -l <TIME_LIMIT> -z <SIZE_LIMIT> -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -w
<LDAP_BINDPASSWORD> -s sub "<search query>"
```

em que

- `<LDAP_URL>` é a URL do servidor LDAP.
- `<LDAP_BASEDN>` é o DN Base LDAP.
- `<LDAP_BINDDN>` é o DN de Ligação LDAP.
- `<LDAP_BINDPASSWORD>` é a senha do DN de Ligação LDAP.
- `<TIME_LIMIT>` é o limite de tempo em segundos para a procura. O valor padrão é 5 segundos.
- `<SIZE_LIMIT>` é o limite de tamanho para procura. O valor padrão é 50 entradas.

No IBM Cloud Private, a sequência de procura é baseada nos atributos `cn` e `Group ID map` que estão configurados nos filtros de grupo de conexões LDAP.

Por exemplo, considere a configuração de filtros de grupo LDAP a seguir:

```
Group filter: (&(cn=%v)(objectclass=groupOfUniqueNames))
Group ID map: *:cn
```

Para a configuração de exemplo, a seguir está a consulta de procura:

```
(&(cn=*<searchstring>*)(objectclass=groupOfUniqueNames))
```

Em que o valor do nome do grupo é `<searchstring>`. Por exemplo, `security` ou `administrators`.

Se o atributo `Group ID map` for diferente de `cn`, por exemplo, `gid`, a consulta de procura a seguir poderá ser usada:

```
(|(&(cn=*<searchstring>*)(objectclass=groupOfUniqueNames))(&(gid=*<searchstring>*)
(objectclass=groupOfUniqueNames)))
```

Em que o nome do grupo ou o valor `gid` é `<searchstring>`.

A seguir está um exemplo de comando `ldapsearch`:

```
ldapsearch -x -l 50 -z 100 -H "ldap://X.X.X.X:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w
'password' -s sub "(&(cn=*gr*)(objectclass=groupOfUniqueNames))"
```

Em que `<searchstring>` é `gr`.

O seguinte é uma saída de amostra:

```
# extended LDIF
#
# LDAPv3
# base <dc=abc,dc=com> with scope subtree
```

```
# filter: (&(cn=*gr*)(objectclass=groupOfUniqueNames))
# requesting: ALL
#

# group1, Groups, abc.com
dn: cn=group1,ou=Groups,dc=abc,dc=com
cn: group1
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: cn=group2,ou=Groups,dc=abc,dc=com
uniqueMember: uid=user1,ou=Users,dc=abc,dc=com

# group2, Groups, abc.com
dn: cn=group2,ou=Groups,dc=abc,dc=com
cn: group2
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=user2,ou=Users,dc=abc,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Recupere usuários de um grupo LDAP

Use o comando `ldapsearch` a seguir:

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub
"(cn=<GROUP_NAME>)"
```

em que

- `<LDAP_URL>` é a URL do servidor LDAP.
- `<LDAP_BASEDN>` é o DN Base LDAP.
- `<LDAP_BINDDN>` é o DN de Ligação LDAP.
- `<LDAP_BINDPASSWORD>` é a senha do DN de Ligação LDAP.
- `<GROUP_NAME>` é o nome do grupo.

Considere o exemplo de comando a seguir:

```
ldapsearch -x -H "ldap://X.X.X.X:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w 'password' -
s sub "(cn=group2)"
```

O seguinte é uma saída de amostra:

```
# extended LDIF
#
# LDAPv3
# base <dc=abc,dc=com> with scope subtree
# filter: (cn=group2)
# requesting: ALL
#

# group2, Groups, abc.com
dn: cn=group2,ou=Groups,dc=abc,dc=com
cn: group2
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=user2,ou=Users,dc=abc,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Problemas do pod

Resolva problemas do pod de segurança.

O pod auth-idp reinicia várias vezes

O pod `auth-idp` reinicia várias vezes.

Sintomas

O pod `auth-idp` reinicia várias vezes. O `platform-auth-service` mostra os seguintes logs:

```
kubectl logs auth-idp -n kube-system -c platform-auth-service
```

```
2019-02-12 18:18:12,170 WARN received SIGTERM indicating exit request
2019-02-12 18:18:12,170 INFO waiting for 01-wlp-platform-auth-service, 02-directory-service to die
2019-02-12 18:18:12,183 INFO stopped: 02-directory-service (terminated by SIGTERM)
[AUDIT   ] CWWKE0085I: The server defaultServer is stopping because the JVM is exiting.
[AUDIT   ] CWWKE1100I: Waiting for up to 30 seconds for the server to quiesce.
[AUDIT   ] CWWKT0017I: Web application removed (default_host): http://auth-idp-v9h8q:9080/oidc/
[AUDIT   ] CWWKT0017I: Web application removed (default_host): http://auth-idp-v9h8q:9080/oauth2/
2019-02-12 18:18:15,385 INFO waiting for 01-wlp-platform-auth-service to die
2019-02-12 18:18:18,387 INFO waiting for 01-wlp-platform-auth-service to die
2019-02-12 18:18:21,390 INFO waiting for 01-wlp-platform-auth-service to die
2019-02-12 18:18:22,391 WARN killing '01-wlp-platform-auth-service' (9) with SIGKILL
2019-02-12 18:18:22,514 INFO stopped: 01-wlp-platform-auth-service (terminated by SIGKILL)
```

Causas

O `platform-auth-service` está usando mais recursos do que o limite configurado. Esse problema é visto mais frequentemente em plataformas Linux on IBM Z and LinuxONE.

Resolvendo o problema

Para resolver o problema, remova os limites de recursos.

1. Instale o kubectl. Consulte Instalando a CLI do Kubernetes (kubectl) .
2. Edite o daemonset `auth-idp`.

   ```
   kubectl edit ds auth-idp -n kube-system
   ```

3. Localize a seção a seguir:

   ```
   name: platform-auth-service
   ports:
   - containerPort: 9443
     hostPort: 9443
     name: http
     protocol: TCP
   readinessProbe:
     failureThreshold: 3
     httpGet:
       path: /
       port: 9443
       scheme: HTTPS
     periodSeconds: 10
     successThreshold: 1
     timeoutSeconds: 1
   resources:
     limits:
       cpu: "1"
       memory: 1Gi
     requests:
       cpu: 100m
       memory: 256Mi
   ```

4. Remova a seção `limits` sob a seção `resources`. A seção `resources` deve ser semelhante à seguinte parte de código após a mudança:

```
resources:
  requests:
    cpu: 100m
    memory: 256Mi
```

5. Salve o arquivo e espere até que todos os pods `auth-idp` sejam reiniciados. Os pods podem levar alguns minutos para reiniciar.

Os pods não são planejados

Os pods de segurança não são planejados, pois os nós principais não atendem aos requisitos necessários de memória ou de CPU.

Se não for possível efetuar login na console de gerenciamento, verifique se os seguintes pods são planejados:

```
auth-idp-xxx
auth-apikeys-xxx
auth-pap-xxx
auth-pdp-xxx
secret-watcher-xxx
security-onboarding-xxx
```

Se algum desses pods não estiver planejado, obtenha informações sobre o controlador do pod (daemonset, statefulset, deployment e outros componentes) e verifique se o pod não está sendo planejado devido à indisponibilidade de recursos. Use o seguinte comando para obter as informações:

```
kubectl describe daemonset auth-idp -n kube-system
```

A saída tem as informações sobre requisitos de memória e de CPU. Se o requisito do recurso não for atendido, aumente a memória ou a CPU do nó principal. Para obter informações adicionais, consulte Requisitos e recomendações de hardware.

Depois de aumentar os recursos, os pods são planejados automaticamente.

Segurança

Revise problemas de segurança encontrados frequentemente.

- Resolvendo problemas do Key Management Service
- Resolvendo problemas do plug-in do Key Management Service
- O pod auth-idp reinicia várias vezes

Resolução de problemas do Key Management Service

Resolva problemas comuns do Key Management Service.

Instale a CLI Kubernetes para executar os comandos de resolução de problemas. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

- Erro FAILED do UPGRADE
- A rotação de chave não funciona - mostra o erro 501 Not Implemented
- As operações de chave não funcionam - mostram o erro 400 Bad Request
- As operações de chave não funcionam - mostram o Erro 500 Internal Server
- As operações de chave não funcionam - mostram o erro 503 Unavailable Experiencing delays
- A conexão HSM não funciona em todos os nós de gerenciamento
- Não é possível importar chave raiz
- Log key-management-persistence relata erros após a configuração do Key Management Service
- O Kubernetes Ingress Controller Fake Certificate é retornado pelo controlador de ingresso do NGINX
- Não é possível importar chave raiz

Erro FAILED do UPGRADE

Sintoma

O upgrade do gráfico do Helm do 3.1.2 para o 3.2.0 não funciona. Você vê o erro `Erro: UPGRADE FAILED`.

Causa

Você não especificou o arquivo de configuração `overrides.yaml` durante o upgrade do Helm.

Solução

1. Crie um arquivo de configuração `overrides.yaml` separado e especifique o novo caminho da imagem para o IBM® Cloud Private 3.2.0 no arquivo.

A seguir há um arquivo `overrides.yaml` de amostra:

```
api:
  image:
    repository: mycluster.icp:8500/ibmcom/kms-api-amd64
    tag: <ICP_VERSION, like 3.2.0>

persistence:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-persistence
    tag: <ICP_VERSION, like 3.2.0>

storage:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-onboarding
    tag: <ICP_VERSION, like 3.2.0>

lifecycle:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-lifecycle
    tag: <ICP_VERSION, like 3.2.0>

pep:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-pep
    tag: <ICP_VERSION, like 3.2.0>

crypto:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/kms-crypto
    tag: <ICP_VERSION, like 3.2.0>

auditService:
  image:
    repository: <CLUSTER_NAME>.icp:8500/ibmcom/icp-audit-service
    tag: <ICP_VERSION, like 3.2.0>
```

1. Especifique o arquivo quando você executar o comando `helm upgrade`.

```
helm upgrade -f overrides.yaml
```

A rotação de chave não funciona - mostra o erro 501 Not Implemented

Sintoma

Depois da instalação do gráfico do Helm `key-management-hsm`, a rotação de teclas não funciona com o Hardware Security Module (HSM). Você vê o erro `501 Not Implemented Error`.

Causa

A rotação de chave é suportada a partir da versão do 3.2.0 .

Solução

Instale o gráfico do Helm `key-management-3.1.2.tgz` ou faça upgrade da liberação.

As operações de chave não funcionam - mostram o erro 400 Bad Request

Sintoma

Após o upgrade do gráfico do Helm de gerenciamento de chaves, as operações para criação, agrupamento ou desagrupamento de chaves não funcionam com o HSM. O log contém o erro a seguir: `400 Bad Request Error: "Provided API key could not be found"`.

Causa

Os dados `kms-api-key` que estão contidos no `key-management-secret` foram sobrescritos para um valor inválido de "default_kms_api_key".

Solução

1. Crie um novo `api-key` seguindo as instruções em APIs de gerenciamento de chaves da API.
2. Criptografe a chave com a criptografia base64.
3. Sobrescreva os dados existentes na seção `kms-api-key` do segredo usando o {{site.data.keyword.console}}.
4. Reinicie o pod removendo o pod `key-management-pep`.

As operações de chave não funcionam - mostram o Erro 500 Internal Server

Sintoma

Depois da instalação do gráfico do Helm `key-management-hsm`, não é possível criar chaves ou agrupar ou desagrupar chaves com HSM. Você vê o erro `500 Internal Server Error`.

Causa

A limpeza da tarefa não é concluída devido à incompatibilidade do caminho do repositório de imagem.

Solução

1. Remova a tarefa em lote `key-management-hsm-cleanup`.
    1. Efetue login no console de gerenciamento.
    2. No menu de navegação, selecione **Cargas de Trabalho** > **Tarefas** > **Tarefas em Lote**.
    3. Coloque o cursor sobre a tarefa em lote `key-management-hsm-cleanup`.
    4. Clique em **...** > **Remover** para remover a tarefa em lote.
2. Reimplemente o gráfico do Helm `key-management-hsm`.

As operações de chave não funcionam - mostram o erro 503 Unavailable Experiencing delays

Solução: verifique o status do HSM e veja se ele está off-line ou se a configuração foi mudada.

Sintoma

Após o upgrade do gráfico do Helm de gerenciamento de chaves, as operações para criação, agrupamento ou desagrupamento de chaves não funcionam com o HSM. O log contém o erro a seguir: `503 Service Error “Unavailable Experiencing delays. Please try again in few minutes.”`

Causa

O HSM que está conectado ao `key-management-hsm-middleware` está indisponível ou encerrado.

Solução

1. Verifique o status do HSM para determinar se ele está off-line ou se sua configuração foi mudada.
2. Restaure as definições de configuração originais para o HSM, se elas foram mudadas.

3. Reinicie o HSM.

A conexão HSM não funciona em todos os nós de gerenciamento

Sintoma

A conexão HSM funciona em alguns, mas não em todos os nós de gerenciamento.

Causa

O certificado e os pares de chaves não são localizados nos nós de gerenciamento nos quais o HSM não funciona.

Solução

1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Verifique o segredo do HSM para confirmar se o certificado e os pares de chaves são listados para todos os nós de gerenciamento.

   ```
   kubectl get secret hsm-secret -o yaml --namespace kube-system
   ```

   As informações estão disponíveis no formato a seguir:

   ```
   <master-node-IP>: <BASE64_ENCODED_CERTIFICATE>
   <master-node-IP-key>: <BASE64_ENCODED_KEY>
   ```

Não é possível importar chave raiz

É possível importar chaves raiz somente quando você usa um modelo HSM suportado. O SoftHSM não é suportado.

Para obter os modelos HSM suportados, consulte Configurando o Key Management Service.

Log key-management-persistence relata erros após a configuração do Key Management Service

Sintoma

Depois de configurar o Key Management Service, você verá erros no log `key-management-persistence`.

```
kubectl logs key-management-persistence-5d6974bf8c-vxxwl -- namespace kube-system
```

O seguinte é uma saída de amostra:

```
2018/11/27 14:31:13 maxprocs: Leaving GOMAXPROCS=8: CPU quota undefined
{"caller":"config.go:402","component":"config","file":"/opt/keyprotect/config//production","location
":"local","msg":"config loaded from local","ts":"2018-11-27T14:31:13.891450032Z"}
{"caller":"root.go:104","commit":"5bbc1228","component":"root","semver":"2.1.0","ts":"2018-11-
27T14:31:15.157576488Z"}
Creating MongoDB session with options: [mongodb:27017],  rs0
Failed to create session:  no reachable servers
Creating MongoDB session with options: [mongodb:27017],  rs0
Failed to create session:  no reachable servers
Creating MongoDB session with options: [mongodb:27017],  rs0
Failed to create session:  no reachable servers
Creating MongoDB session with options: [mongodb:27017],  rs0
```

Causa

Os contêineres no nó de gerenciamento falharam ao consultar outros serviços no nó principal. A tabela de roteamento não foi configurada corretamente devido a um problema de configuração com o `kube-controller`.

Solução

Atualize a configuração do `kube-controller`.

O Kubernetes Ingress Controller Fake Certificate é retornado pelo controlador de

ingresso do NGINX

Sintoma

Ao chamar https://proxy_ip/, um Kubernetes Ingress Controller Fake Certificate é retornado.

Causa

O Kubernetes Ingress Controller Fake Certificate é usado como o certificado SSL padrão no controlador de ingresso do NGINX.

Solução

É possível configurar o `--default-ssl-certificate` no daemonset nginx-ingress-controller para substituir o "Kubernetes Ingress Controller Fake Certificate".

Por exemplo:

1. Crie um segredo que contenha um certificado SSL:

```
openssl genrsa -out ing-tls.key 4096
openssl req -new -key ing-tls.key -out ing-tls.csr -subj "/CN=TTTEEESSSTTT"
openssl x509 -req -days 36500 -in ing-tls.csr -signkey ing-tls.key -out ing-tls.crt
kubectl create secret tls ing-tls-secret --cert=ing-tls.crt --key=ing-tls.key -n kube-system
```

2. Configure `--default-ssl-certificate` no daemonset nginx-ingress-controller. Por exemplo:

```
kubectl edit ds -n kube-system nginx-ingress-controller
```

```
      contêineres:
      - args:
        - /nginx-ingress-controller
        - --default-backend-service=$(POD_NAMESPACE)/default-http-backend
        - --configmap=$(POD_NAMESPACE)/nginx-ingress-controller
        - --annotations-prefix=ingress.kubernetes.io
        - --enable-ssl-passthrough=true
        - --publish-status-address=172.16.247.161
        - --default-ssl-certificate=$(POD_NAMESPACE)/ing-tls-secret
```

3. Verifique o resultado. Por exemplo:

```
 # ps -ef | grep nginx-ingress-controller | grep default-ssl-certificate
 33       23251 23207  0 22:45 ?        00:00:00 /usr/bin/dumb-init -- /nginx-ingress-
controller --default-backend-service=kube-system/default-http-backend --configmap=kube-
system/nginx-ingress-controller --annotations-prefix=ingress.kubernetes.io --enable-ssl-
passthrough=true --publish-status-address=172.16.247.161 --default-ssl-certificate=kube-
system/ing-tls-secret
 33       23308 23251  0 22:45 ?        00:00:02 /nginx-ingress-controller --default-backend-
service=kube-system/default-http-backend --configmap=kube-system/nginx-ingress-controller --
annotations-prefix=ingress.kubernetes.io --enable-ssl-passthrough=true --publish-status-
address=172.16.247.161 --default-ssl-certificate=kube-system/ing-tls-secret

 # curl -kv  https://172.16.247.161
 * About to connect() to 172.16.247.161 port 443 (#0)
 *   Trying 172.16.247.161...
 * Connected to 172.16.247.161 (172.16.247.161) port 443 (#0)
 * Initializing NSS with certpath: sql:/etc/pki/nssdb
 * skipping SSL peer certificate verification
 * SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 * Certificado do servidor:
 *       subject: CN=TTTEEESSSTTT
 *       start date: May 05 05:44:02 2019 GMT
 *       expire date: Apr 11 05:44:02 2119 GMT
 *       common name: TTTEEESSSTTT
 *       issuer: CN=TTTEEESSSTTT
 > GET / HTTP/1.1
 > User-Agent: curl/7.29.0
 > Host: 172.16.247.161
 > Accept: */*
 >
 < HTTP/1.1 404 Not Found
 < Date: Sun, 05 May 2019 05:49:49 GMT
 < Content-Type: text/plain; charset=utf-8
```

```
< Content-Length: 21
< Connection: keep-alive
< Strict-Transport-Security: max-age=15724800; includeSubDomains
<
* Connection #0 to host 172.16.247.161 left intact
```

Pod key-management-pep não está em execução

Sintoma

O pod `key-management-pep` não está em execução e exibe "`CreateContainerConfigError`".

Causa

Os dados `kms-api-key` dentro do valor de `key-management-secret` não são válidos.

Solução

1. Verifique o status do pod secret-watcher.
2. Se o pod estiver em execução, reinicie-o.
3. Se ele não estiver em execução, consulte o guia de resolução de problemas para o serviço do observador secreto.

Resolvendo problemas do plug-in do Key Management Service

Resolva problemas comuns do plug-in do Key Management Service.

Instale a CLI Kubernetes para executar os comandos de resolução de problemas. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

Falha ao criar um segredo: a chave de API não pôde ser localizada

Sintoma

Falha ao criar um segredo usando a CLI ou a console de gerenciamento. Você vê o erro `Error from server (InternalError):`
`Internal error occurred: rpc error: code = Unknown desc = BXNIM0415E:Provided API key could not be found.`

Causa

A chave de API fornecida não está correta.

Solução

1. Especifique a `API_Key` correta no arquivo `/etc/cfc/conf/kmsplugin-config.yaml`.
2. Reinicie o contêiner de plug-in do KMS depois de atualizar o arquivo. É possível reiniciar o contêiner de plug-in KMS excluindo o pod de plug-in KMS existente.

```
kubectl delete pods k8s_kmsplugin-<master_node_IP_address>
```

Falha ao criar um segredo: a conexão está indisponível

Sintoma

Falha ao criar um segredo usando a CLI ou a console de gerenciamento. Você vê o erro `Internal error occurred: rpc error: code = Unavailable desc = grpc: the connection is unavailable.`

Causa

O ID da Chave Raiz do Cliente não está correto.

Solução

1. Corrija `CRK_ID` no arquivo `/etc/cfc/conf/kmsplugin-config.yaml`.
2. Reinicie o contêiner de plug-in do KMS depois de atualizar o arquivo. É possível reiniciar o contêiner de plug-in KMS excluindo o pod de plug-in KMS existente.

```
kubectl delete pods k8s_kmsplugin-<master_node_IP_address>
```

Falha ao criar um segredo: a solicitação requer um Cabeçalho da Instância válido contendo um UUID válido

Sintoma

Falha ao criar um segredo usando a CLI ou a console de gerenciamento. Você vê o erro `Internal error occurred: rpc error: code = Unknown desc = Bad Request: Request requires valid Instance Header containing a valid UUID`.

Causa

O ID da instância do Key Management Service não está correto.

Solução

1. Corrija `INSTANCE_ID` no arquivo `/etc/cfc/conf/kmsplugin-config.yaml`.
2. Reinicie o contêiner de plug-in do KMS depois de atualizar o arquivo. É possível reiniciar o contêiner de plug-in KMS excluindo o pod de plug-in KMS existente.

```
kubectl delete pods k8s_kmsplugin-<master_node_IP_address>
```

Falha ao criar um segredo: Client.Timeout excedido enquanto aguardava cabeçalhos

Sintoma

Falha ao criar um segredo usando a CLI ou a console de gerenciamento. Você vê o erro `Error from server (InternalError): Internal error occurred: rpc error: code = Unknown desc = Post https://kms-api.kube-system:28674/api/v2/keys/3ecbc3be-3534-41cd-9898-a224134fbb55?action=wrap: net/http: request canceled while waiting for connection (Client.Timeout exceeded while awaiting headers)`.

Causa

O Key Management Service não respondeu.

Solução

1. Efetue login no console de gerenciamento.
2. A partir do menu de navegação, selecione **Cargas de Trabalho** > **Implementações**.
3. Selecione `key-management-api`.
4. Role para baixo até Pods.
5. Coloque o cursor sobre a única linha listada em **Pods**.
6. Clique em **...** > **Remover** para remover o pod e para criar um novo pod.

console de gerenciamento

Revise problemas do console de gerenciamento encontrados frequentemente.

- Não é possível acessar o console de gerenciamento (após a reinicialização do nó principal)
- O Catalog está vazio após a reinicialização do nó principal
- Um pod está travado no estado *Finalizando*
- A conexão falha no terminal da web

- Um namespace está travado no estado *Finalizando*
- As liberações do Helm não são exibidas
- Não é possível enviar por push novas imagens para o IBM Cloud Private

Não é possível acessar o console de gerenciamento (após a reinicialização do nó principal)

Impossível acessar o console de gerenciamento após reiniciar o nó principal.

Sintomas

Não é possível visualizar o console de gerenciamento após a reinicialização de um nó principal.

Causas

As portas necessárias podem não estar mais acessíveis.

Resolvendo o problema

Verifique se todas as portas padrão estão abertas e disponíveis. Para obter uma lista de portas padrão, consulte Portas padrão

O Catalog está vazio após a reinicialização do nó principal

Depois de reiniciar o nó principal, o Catalog está vazio.

Sintomas

Não é possível visualizar os aplicativos disponíveis no Catalog depois que um nó principal é reinicializado.

Causas

Se o serviço `helm-api` iniciar antes que o serviço `MongoDB` seja iniciado, a sincronização inicial do Catalog falhará.

Resolvendo o problema

É possível resolver esse problema executando uma sincronização manual de seus repositórios Helm para preencher novamente o Catalog. Conclua o procedimento a seguir para executar uma sincronização manual:

1. Na navegação, selecione **Gerenciar** > **Repositórios Helm**.
2. Selecione **Sincronizar repositórios** na página principal *Repositórios Helm*. **Dica:** Também é possível sincronizar um único repositório, selecionando o menu de ação (**...**), em seguida, selecionando **Sincronizar este repositório**.

Depois que os repositórios Helm concluem a sincronização, é possível visualizar os aplicativos de destaque no Catalog.

Um pod está preso no estado *Terminating*

Sintoma

Um pod está preso no estado *Terminating* depois de tentar excluí-lo.

Resolvendo o problema

Você deve excluir manualmente o pod. Execute o comando a seguir:

```
kubectl -n <namespace> delete pods --grace-period=0 --force <pod_name>
```

Para obter mais informações sobre a CLI `kubectl`, consulte Comandos do kubectl.

A conexão falha no terminal da web

Ao abrir o terminal da web, uma conexão é feita para um contêiner em execução. A conexão falha no terminal da web e o Catalog parece estar vazio.

Sintomas

Não é possível usar o terminal da web a partir do console de gerenciamento para se comunicar com seu cluster.

Causas

Não há pods suficientes em execução para concluir a conexão.

Resolvendo o problema

É possível resolver esse problema diminuindo a implementação do `web-terminal`. Para obter mais detalhes sobre as implementações de ajuste de escala, consulte Implementações de ajuste de escala

Um namespace está parado no estado *Finalizando*

Sintoma

Um namespace está parado no estado *Finalizando*

Causa

Se uma extensão de API do Kubernetes não estiver disponível, os recursos que são gerenciados pela extensão não poderão ser excluídos. A falha ao excluir a extensão de API faz com que a exclusão de namespace falhe.

Resolvendo o problema

Obtenha as descrições de API que não são excluídas

Conclua as etapas a seguir para obter uma descrição das APIs que não são excluídas:

1. Visualize os namespaces que estão parados em um estado *Finalizando*:

   ```
   kubectl get namespaces
   ```

2. Localize os recursos que não são excluídos:

   ```
   kubectl api-resources --verbs=list --namespaced -o name | xargs -n 1 kubectl get --show-kind --show-all --ignore-not-found -n <terminating-namespace>
   ```

3. Se o comando anterior retornar a seguinte mensagem de erro: `não é possível recuperar a lista completa de APIs do servidor: <api-resource>/<version>: atualmente o servidor é incapaz de manipular a solicitação`, continue executando o seguinte comando com as informações recebidas:

   ```
   kubectl get APIService <version>.<api-resource>
   ```

   Por exemplo, execute o comando a seguir para um serviço de API que é denominado `custom.metrics.k8s.io/v1beta1`:

   ```
   kubectl obter APIService v1beta1.custom.metrics.k8s.io
   ```

4. Obtenha uma descrição do serviço de API para continuar a depurar seu serviço de API. Execute o comando a seguir:

   ```
   kubectl describe APIService <version>.<api-resource>
   ```

5. Certifique-se de que o problema tenha sido resolvido. Execute o comando a seguir para verificar se seu namespace pode ser excluído:

   ```
   kubectl get namespace
   ```

Excluir manualmente um namespace de finalização

Se o problema não for resolvido, será possível excluir manualmente seu namespace que está parado no estado *Finalizando*.

1. Visualize os namespaces que estão parados no estado *Finalizando*:

   ```
   kubectl get namespaces
   ```

2. Selecione um namespace de finalização e visualize o conteúdo do namespace para localizar o `finalizador`:

   ```
   kubectl get namespace < terminating-namespace> -o yaml
   ```

   O conteúdo de YAML pode ser semelhante à saída a seguir:

   ```
   apiVersion: v1
   kind: Namespace
   metadata:
     creationTimestamp: 2018-11-19T18:48:30Z
     deletionTimestamp: 2018-11-19T18:59:36Z
     name: <terminating-namespace>
     resourceVersion: "1385077"
     selfLink: /api/v1/namespaces/<terminating-namespace>
     uid: b50c9ea4-ec2b-11e8-a0be-fa163eeb47a5
   spec:
     finalizers:
     - kubernetes
   status:
     phase: Terminating
   ```

3. Crie um arquivo JSON temporário:

   ```
   kubectl get namespace <terminating-namespace> -o json >tmp.json
   ```

4. Edite seu arquivo `tmp.json` . Remova o valor de `kubernetes` do campo `finalizers` e salve o arquivo.

   Seu arquivo `tmp.json` pode ser semelhante à saída a seguir:

   ```
    {
        "apiVersion": "v1",
        "kind": "Namespace",
        "metadata": {
            "creationTimestamp": "2018-11-19T18:48:30Z",
            "deletionTimestamp": "2018-11-19T18:59:36Z",
            "name": "<terminating-namespace>",
            "resourceVersion": "1385077",
            "selfLink": "/api/v1/namespaces/<terminating-namespace>",
            "uid": "b50c9ea4-ec2b-11e8-a0be-fa163eeb47a5"
        }, "spec": {
           "finalizers":
        },
        "status": {
            "phase": "Terminating" }
    }
   ```

5. Para configurar um IP e uma porta do proxy temporário, execute o comando a seguir. Certifique-se de que a janela do terminal fique aberta até você excluir o namespace parado:

   ```
    kubectl proxy
   ```

   Seu IP do proxy e a porta podem ser semelhantes à saída a seguir:

   ```
    Iniciando a servir em 127.0.0.1:8001
   ```

6. Em uma nova janela do terminal, faça uma chamada da API com a porta e o IP temporários do proxy:

   ```
    curl -k -H "Content-Type: application/json" -X PUT --data-binary @tmp.json
   http://127.0.0.1:8001/api/v1/namespaces/<terminating-namespace>/finalize
   ```

   A saída pode ser semelhante ao conteúdo a seguir:

   ```
   {
     "kind": "Namespace",
     "apiVersion": "v1",
     "metadata": {
       "name": "<terminating-namespace>",
       "selfLink": "/api/v1/namespaces/<terminating-namespace>/finalize",
   ```

```
    "uid": "b50c9ea4-ec2b-11e8-a0be-fa163eeb47a5",
    "resourceVersion": "1602981",
    "creationTimestamp": "2018-11-19T18:48:30Z",
    "deletionTimestamp": "2018-11-19T18:59:36Z"
  }, "spec": {

  },
  "status": {
    "phase": "Terminating" }
}
```

Note: O parâmetro `finalizer` é removido.

7. Verifique se o namespace em finalização foi removido:

```
kubectl get namespaces
```

8. Continue seguindo as etapas para outros namespaces que travaram no estado *Finalizando*.

As liberações do Helm não são exibidas

As liberações do Helm não são exibidas na página *Liberações do Helm* na console de gerenciamento.

Sintomas

Ao acessar a página *Liberações do Helm* selecionando **Cargas de trabalho** > **Liberações do Helm** no menu de navegação, as liberações não são listadas.

Causas

O número de liberações que são solicitadas excede o número de liberações que podem ser carregadas e exibidas na console de gerenciamento.

Resolvendo o problema

Para exibir o conteúdo, acesse a lista usando os comandos da CLI e limite o número de liberações que são exibidas. As etapas são descritas no procedimento a seguir:

1. Efetue login em seu cluster com a CLI do Helm. Consulte Configurando a API Helm para a estrutura de comando.
2. Execute o comando `helm list` e limite o número de liberações retornadas com as opções a seguir:
   - Use a opção `-m` ou `--max`: a opção `-m` ou `--max` configura o número máximo de liberações que são retornadas da solicitação para exibir a lista de liberações. A variável deve ser um número que especifica quantas devem ser retornadas. Quando o número for baixo o suficiente para que a memória esteja disponível, as liberações serão exibidas. O exemplo a seguir exibe as *10* primeiras liberações usando a opção `-m`:

     ```
     helm list --tls -m 10
     ```

   - Use a opção `-o` ou `--offset`: a opção `-o` ou `--offset` especifica o nome da liberação com a qual iniciar ao retornar a lista de liberações. O exemplo exibe a lista de todas as liberações que começam com a liberação intitulada *release1* e termina com a mais recente:

     ```
     helm list --tls -o release1
     ```

   - Use as opções `-m` ou `--max` e `-o` ou `--offset` para limitar a lista: é possível usar essas opções juntas para filtrar ainda mais as liberações que são exibidas. O exemplo a seguir exibe a lista de 10 liberações que começam com a liberação intitulada *release1*.

     ```
     helm list --tls -o release1 -m 10
     ```

Não é possível enviar por push novas imagens para o IBM Cloud Private

Não é possível enviar por push uma nova imagem de um namespace existente em sua unidade local para o registro de imagem do IBM® Cloud Private.

Cenário 1: Não é possível enviar por push uma imagem para um registro privado com um terminal *https*

Sintomas

Conforme você efetua logon no cluster do IBM Cloud Private, não é possível enviar por push novas imagens de sua unidade local com um terminal `https`. Seu terminal pode exibir a seguinte mensagem de erro:

```
# docker push mycluster.icp:8500/open-liberty:latest
The push refers to a repository [mycluster.icp:8500/open-liberty]
ce633891d99e: Pushing [==================================================>]  6.656kB
16ee2ef8c0a9: Layer already exists
d83aa1f72c39: Layer already exists
6b8c4250e63e: Pushing   2.56kB
000ea2a5eb7d: Layer already exists
5f2b8ff02676: Retrying in 5 seconds
c1eb2e939cb4: Layer already exists
ad60ad386e49: Retrying in 5 seconds
db584c622b50: Retrying in 5 seconds
52a7ea2bb533: Waiting
52f389ea437e: Waiting
88888b9b1b5b: Waiting
a94e0d5a7c40: Waiting
unknown blob
```

Causa

O balanceador de carga do cluster do IBM Cloud Private está configurado para balancear solicitações para nós principais. O balanceador de carga que se comunica com o registro foi direcionado incorretamente pelo servidor.

Resolvendo o problema

Se o balanceador de carga do cluster estiver configurado, inclua `http-request set-header X-Forwarded-Proto https if { ssl_fc }` em sua configuração de back-end. Para obter informações adicionais, consulte o Repositório do Docker no GitHub.

Cenário 2: O armazenamento NFS não está auxiliando o armazenamento do registro]

Sintomas

Conforme você efetua logon no cluster do IBM Cloud Private, não é possível enviar por push novas imagens de sua unidade local, porque o armazenamento de back-end do registro do Docker não foi configurado para replicação. Seu terminal pode exibir a seguinte mensagem de erro:

```
# docker push mycluster.icp:8500/open-liberty:latest
The push refers to a repository [mycluster.icp:8500/open-liberty]
ce633891d99e: Pushing [==================================================>]  6.656kB
16ee2ef8c0a9: Layer already exists
d83aa1f72c39: Layer already exists
6b8c4250e63e: Pushing   2.56kB
000ea2a5eb7d: Layer already exists
5f2b8ff02676: Retrying in 5 seconds
c1eb2e939cb4: Layer already exists
ad60ad386e49: Retrying in 5 seconds
db584c622b50: Retrying in 5 seconds
52a7ea2bb533: Waiting
52f389ea437e: Waiting
88888b9b1b5b: Waiting
a94e0d5a7c40: Waiting
unknown blob
```

Causa

Se o seu cluster IBM Cloud Private atender às seguintes condições, o problema está ocorrendo porque as configurações do servidor NFS não atendem aos requisitos para o NFS do registro:

- O cluster do IBM Cloud Private está configurado para usar uma arquitetura de alta disponibilidade.
- As montagens de NFS para todas as réplicas de Docker Trusted Registry (DTR) estão configuradas.

Resolvendo o problema

Se você enviar por push uma imagem para um DTR de alta disponibilidade, certifique-se de configurar o DTR e usar as opções do servidor NFS necessárias, `sync` e `actimeo=0`. Para obter informações adicionais, consulte Enviando por push para um DTR HA na página do centro de sucesso do Docker.

Redes

Revisão frequentemente encontrou problemas de rede.

- Resolução de problemas de redes Calico
- Resolução de problemas do F5 BIG-IP LTMF5 BIG-IP
- Comandos da CLI do Helm falham

Resolução de problemas de redes Calico

Identificando e investigando problemas de rede do Calico.

Podem ocorrer problemas de rede do Calico durante ou após a instalação do IBM® Cloud Private. Durante a instalação, o instalador executa verificações para assegurar a conectividade contínua de um pod a outro no cluster. No entanto, se ainda houver problemas, as instruções a seguir podem ajudar a identificar as causas e resolver os problemas.

- Resolução de problemas de isolamento do ambiente
- Resolução de problemas de malha do IPsec
- Resolução de problemas do NSX-T

Problemas durante a instalação do IBM Cloud Private

Para evitar problemas de rede do Calico durante a instalação, assegure-se de que as configurações a seguir estejam definidas corretamente.

- O parâmetro `calico_ipip_mode` deverá ser configurado como `Always`, se todos os nós em seu cluster não pertencerem à mesma sub-rede.

  Deve-se também configurar esse parâmetro como `Always` quando os nós são implementados em ambientes de nuvem, como o OpenStack, em que as verificações de origem e de destino evitam o tráfego de IP de intervalos de endereços IP desconhecidos. O parâmetro deve ser configurado mesmo que todos os nós pertençam à mesma sub-rede. Essa configuração permite o encapsulamento do tráfego entre um pod e outro pela infraestrutura de rede subjacente.

- O parâmetro `calico_ip_autodetection_method` deve ser configurado para que o Calico selecione a interface correta no nó. Caso haja várias interfaces, aliases, interfaces lógicas, interfaces de ponte e ou qualquer outro tipo de interface nos nós, use uma das configurações a seguir para assegurar que o mecanismo de detecção automática escolha a interface correta.

  - calico_ip_autodetection_method: can-reach= >>> **Nota:** esta é a configuração padrão.
  - calico_ip_autodetection_method: interface= < nome da interface >

- O parâmetro `calico_tunnel_mtu` deve ser configurado com base na MTUda interface que está configurada para ser usada pelo Calico.

  Se o parâmetro `calico_ipip_mode` estiver configurado como `Always`, 20 bytes serão usados para o cabeçalho do túnel IP-IP. Deve-se configurar o parâmetro `calico_tunnel_mtu` para ter pelo menos 20 bytes a menos que a MTU real da interface.

  Se o IPsec estiver ativado, serão necessários 40 bytes para o cabeçalho do pacote IPsec. Além disso, como você configurou o `calico_ipip_mode` como `Always` ao ativar o IPsec, os 20 bytes também são necessários para o cabeçalho do túnel IP-IP. Portanto, deve-se configurar o parâmetro `calico_tunnel_mtu` para ter pelo menos 60 bytes a menos que a MTU real da interface.

- O CIDR de rede, a rede de host existente e o intervalo de IP do cluster de serviço não devem estar em conflito entre si.

Problemas após a instalação do IBM Cloud Private

Depois que seu cluster é instalado, é possível ver os problemas de conectividade IP entre os pods. Os problemas de resolução de nome do serviço são um sintoma de pods que não podem atingir o serviço DNS, mas nem sempre estão relacionados às redes do Calico.

Nessas situações, reúna as informações a seguir a partir de seu cluster para resolução de problemas. Se você entrar em contato com a equipe de suporte para obter assistência, será possível fornecer essas informações para a equipe.

1. Configure a CLI do Kubernetes (kubectl). Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Configure o arquivo binário calicoctl que está disponível na mídia de instalação do IBM Cloud Private. Consulte Instalando a CLI do Calico (calicoctl) .
3. Obtenha a lista de nós em seu cluster.

   ```
   kubectl get nodes -owide
   ```

4. Colete logs do pod `calico-node-*` que é executado no nó que está experimentando o problema de malha. Por exemplo, conclua as etapas a seguir para obter os logs do `calico-node-amd64-48lf9` que são executados no nó `10.10.25.71`.

   1. Obtenha uma lista de pods do Calico.

      ```
      kubectl get pods -o wide | grep calico-node
      ```

      O seguinte é uma saída de amostra:

      ```
      calico-node-amd64-2cbjh                                 2/2       Running   0          7h
      10.10.25.70    10.10.25.70
      calico-node-amd64-48lf9                                 2/2       Running   0          7h
      10.10.25.71    10.10.25.71
      calico-node-amd64-75667                                 2/2       Running   0          7h
      10.10.25.7     10.10.25.7
      ```

   2. Recupere os logs a partir do contêiner `calico-node` no pod.

      ```
      kubectl logs calico-node-amd64-48lf9 -c calico-node
      ```

5. Diagnostique o problema.

   1. Obtenha a tabela de roteamento e detalhes da interface. Execute estes comandos em todos os nós principais e nos nós que possuem os pods que estão tendo problemas de conectividade.

      1. Obtenha detalhes da tabela de roteamento

         ```
         route -n
         ```

      2. Obtenha detalhes da interface.

         ```
         ifconfig -a
         ```

   2. Obtenha a lista de nós do Calico. Execute o comando em qualquer nó principal.

      ```
      calicoctl  get nodes
      ```

   3. Obtenha todos os pods ou os terminais que estão na malha do Calico. Execute o comando em qualquer nó principal.

      ```
      calicoctl  get workloadendpoints
      ```

   4. Obtenha informações de status e de diagnósticos de nó do Calico. Execute estes comandos em qualquer nó principal e nos nós que possuem os pods que estão tendo problemas de conectividade.

      ```
      calicoctl node status
      calicoctl node diags
      ```

   5. Verifique os arquivos `config.yaml` e `host` que estão em seu nó de inicialização.

Resolução de problemas de isolamento do ambiente

Resolução de problemas de rede de isolamento do ambiente.

MustGather

MustGather para o isolamento do ambiente é escrito com um exemplo.

namespaces: devops e produção

dois nós isolados e dedicados para namespace: devops

dois nós isolados e dedicados para namespace: produção

dois nós de proxy isolados e dedicados para namespace: devops

dois nós de proxy isolados e dedicados para namespace: produção

Configuração

Config.yaml

```
## Environment Isolation
# Example:[{namespace: production, hostgroup: prod}, {namespace:devops, hostgroup: dev}, {namespace:
preproduction, hostgroup: preprod}]
isolated_namespaces: [{namespace: devops,  hostgroup: worker-dev}, {namespace: production,
hostgroup: worker-prod}]
isolated_proxies: [{namespace: devops,  hostgroup: proxy-dev, lb_address: x.x.x.x}, {namespace:
production, hostgroup: proxy-prod, lb_address: y.y.y.y}]
```

Arquivo host

hosts

```
[hostgroup-worker-dev]
172.16.206.190
172.16.207.105

[hostgroup-worker-prod]
172.16.208.37
172.16.208.194

[hostgroup-proxy-dev]
172.16.208.195
172.16.159.167

[hostgroup-proxy-prod]
172.16.208.197
172.16.209.180
```

Resolução de problemas

Configuração do serviço da API do Kube

```
Pod mainfest file:
/etc/cfc/pods/master.json

Parameter:
--enable-admission-plugins=PodNodeSelector,PodTolerationRestriction
--admission-control-config-file=/etc/cfc/conf/admission-control-config.yaml

      {
        "name": "apiserver",
        "image": "hyc-cloud-private-stable-docker-local.artifactory.swg-devops.com/ibmcom-
amd64/hyperkube:v1.11.1-ee",
        "imagePullPolicy": "IfNotPresent",



          "--enable-admission-
plugins=Initializers,NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageClass,PodSecurityPo
licy,MutatingAdmissionWebhook,ValidatingAdmissionWebhook,ResourceQuota,Priority,EventRateLimit,PodNo
deSelector,PodTolerationRestriction",
          "--admission-control-config-file=/etc/cfc/conf/admission-control-config.yaml",
```

Configuração de controle de admissão

Arquivo: /etc/cfc/conf/admission-control-config.yaml

```
---
kind: AdmissionConfiguration
apiVersion: apiserver.k8s.io/v1alpha1
plugins:
  - name: EventRateLimit
    path: /etc/cfc/conf/eventconfig.yaml
  - name: PodNodeSelector
    path: /etc/cfc/conf/podnodeselector.yaml
```

Política do seletor do nó do pod

Arquivo: /etc/cfc/conf/podnodeselector.yaml

```
podNodeSelectorPluginConfig:
  clusterDefaultNodeSelector: ""
  production: "worker-prod=true"
  devops: "worker-dev=true"
```

Rótulos de nó

Os nós dedicados para o namespace: "devops" são rotulados como "worker-dev"

```
kubectl -n kube-system get nodes -l "worker-dev"

NAME             STATUS    ROLES        AGE       VERSION
172.16.206.190   Ready     worker-dev   1d        v1.11.1+icp-ee
172.16.207.105   Ready     worker-dev   1d        v1.11.1+icp-ee
```

Os nós dedicados para namespace: "production" são rotulados como "worker-prod"

```
kubectl -n kube-system get nodes -l "worker-prod"

NAME             STATUS    ROLES         AGE       VERSION
172.16.208.194   Ready     worker-prod   1d        v1.11.1+icp-ee
172.16.208.37    Ready     worker-prod   1d        v1.11.1+icp-ee
```

Anotações em namespaces

Namespace: "devops" `kubectl -n kube-system get ns devops -o yaml`

```
apiVersion: v1
kind: Namespace
metadata:
  annotations:
    scheduler.alpha.kubernetes.io/defaultTolerations:
'[{"key":"dedicated","operator":"Equal","value":"worker-dev","effect":"NoSchedule"}]'
    scheduler.alpha.kubernetes.io/node-selector: worker-dev=true
  creationTimestamp: 2018-09-12T08:21:29Z
  name: devops
  resourceVersion: "4857"
  selfLink: /api/v1/namespaces/devops
  uid: d9579db3-b664-11e8-a04b-00163e01af61
spec:
  finalizers:
  - kubernetes
status:
  phase: Active
```

Namespace: "production" `kubectl -n kube-system get ns production  -o yaml`

```
apiVersion: v1
kind: Namespace
metadata:
  annotations:
    scheduler.alpha.kubernetes.io/defaultTolerations:
'[{"key":"dedicated","operator":"Equal","value":"worker-prod","effect":"NoSchedule"}]'
    scheduler.alpha.kubernetes.io/node-selector: worker-prod=true
  creationTimestamp: 2018-09-12T08:21:27Z
  name: production
```

```
  resourceVersion: "4848"
  selfLink: /api/v1/namespaces/production
  uid: d7f29b43-b664-11e8-a04b-00163e01af61
spec:
  finalizers:
  - kubernetes
status:
  phase: Active
```

Proxies isolados

Nós do proxy isolados para namespace: devops

```
kubectl -n kube-system get nodes -l "proxy-dev"

NAME             STATUS    ROLES       AGE       VERSION
172.16.159.167   Ready     proxy-dev   1d        v1.11.1+icp-ee
172.16.208.195   Ready     proxy-dev   1d        v1.11.1+icp-ee
```

Nós do proxy isolados para o namespace: produção

```
kubectl -n kube-system get nodes -l "proxy-prod"

root@envisobase1:~# kc get nodes -l "proxy-prod"
NAME             STATUS    ROLES        AGE       VERSION
172.16.208.197   Ready     proxy-prod   1d        v1.11.1+icp-ee
172.16.209.180   Ready     proxy-prod   1d        v1.11.1+icp-ee
```

Controladores de ingresso

```
kubectl -n kube-system get ds | grep "nginx-ingress"

nginx-ingress-controller              1         1         1         1            1
proxy=true        1d
nginx-ingress-controller-proxy-dev    2         2         2         2            2           proxy-
dev=true    1d
nginx-ingress-controller-proxy-prod   2         2         2         2            2           proxy-
prod=true   1d
```

Controlador de ingresso que entrega devops de namespace

```
kubectl -n kube-system get ds nginx-ingress-controller-proxy-dev -o yaml | grep "watch"

        - --watch-namespace=devops
```

Controlador de ingresso que entrega produção de namespace

```
kubectl -n kube-system get ds nginx-ingress-controller-proxy-prod -o yaml | grep "watch"

        - --watch-namespace=production
```

Resolução de problemas da malha do IPsec

Resolução de problemas de rede da malha do IPsec.

Pré-requisito

1. Cada nó no cluster deve ter pelo menos duas interfaces de rede. Uma é uma interface de gerenciamento e a outra interface fornece redes seguras para os pods. Forneça o endereço IP da interface de gerenciamento em cluster/hosts e outro nome da interface (interface do plano de dados) nas configurações do Calico e do IPsec no cluster/config.yaml.
2. As redes do Calico devem ser ativadas no modo IP-in-IP. A MTU do túnel do Calico deve ser configurada corretamente.
3. O pacote do IPsec que é usado para criptografia deve ser instalado em todos os nós no cluster. O pacote do IPsec que é usado para o RHEL é `libreswan` e no Ubuntu e SLES é `strongswan`.

Nota: todos os nós no cluster devem executar o mesmo sistema operacional.

Configuração

1. Assegure-se de que as configurações do Calico a seguir sejam fornecidas em `config.yaml`.

```
network_type: calico
calico_ipip_mode: Always
calico_tunnel_mtu: 1390
calico_ip_autodetection_method: interface=eth0
```

   - `calico_ipip_mode` deve ser `Always`. O tunelamento IPIP deve ser ativado para o IPsec.
   - O `calico_tunnel_mtu` deve ter pelo menos 60 bytes a menos que a MTU da interface. Se o mtu da interface `eth0` for de 1450 bytes, o `calico_tunnel_mtu` deverá ser configurado como no máximo 1390 bytes.
   - O `calico_ip_autodetection_method` deve ser configurado para escolher a interface do plano de dados.

2. Verifique a configuração do IPsec em `config.yaml`.

```
ipsec_mesh:
  enable: true
  interface: eth0
  subnets: [10.24.10.0/24]
  exclude_ips: [10.24.10.1/32, 10.24.10.2, 10.24.10.192/28]
  cipher_suite: aes128gcm16!
```

   - A `interface` deve ser a mesma que está configurada no parâmetro `calico_ip_autodetection_method`.
   - As `subnets` são os intervalos de endereços. Os pacotes que são destinados a esses intervalos de sub-rede são criptografados. O endereço IP da interface do plano de dados deve cair em um dos intervalos de sub-rede fornecidos.
   - `exclude_ips` são os endereços IP que são excluídos da sub-rede IPsec. O tráfego desses endereços IP não está criptografado.
   - `cipher_suite: aes128gcm16!` é a lista de algoritmos de criptografia ou de autenticação do Encapsulating Security Payload (ESP) a serem usados. O conjunto de cifras padrão que é usado é aes128gcm16!. Assegure-se de que esse módulo esteja disponível e carregado em seu sistema operacional em todos os hosts. Também é possível alterá-lo para qualquer conjunto de cifras de sua escolha.

Pós-instalação

No RHEL

1. Verifique a configuração do libreswan.

```
cat /etc/ipsec.conf
cat /etc/ipsec.d/ipsec-libreswan.conf
```

2. Verifique o status do processo de IPsec.

```
ipsec status
```

   Se o status do IPsec não exibir conexões estabelecidas, verifique `/var/log/messages` para obter um erro relacionado ao IPsec. Ative a criação de log `libreswan` ativando o `plutodebug` no arquivo `/etc/ipsec.conf`.

```
# /etc/ipsec.conf - libreswan IPsec configuration file

Instalação da configuração...
       ...
       plutodebug = all       # <<<<<<<<<<<<
```

No Ubuntu/SLES

1. Verifique a configuração do strongswan.

```
cat /etc/ipsec.conf
```

2. Verifique o status do processo de IPsec.

```
ipsec status
service strongswan status
```

   Se o status do IPsec não exibir conexões estabelecidas, verifique `/var/log/syslog` para obter um erro relacionado ao IPsec.

Ative a criação de log `strongswan` ativando `charondebug` no arquivo `/etc/ipsec.conf`.

```
# /etc/ipsec.conf - libreswan IPsec configuration file

Instalação da configuração...
       ...
       charondebug="ike 2, knl 2, cfg 2"       # <<<<<<<<<<<<
```

Resolução de problemas do NSX-T

Resolução de problemas de rede do NSX-T.

Problemas Conhecidos

- Falha intermitente ao efetuar login no console de gerenciamento em clusters de HA que usam o NSX-T 2.2.

Em clusters de HA que usam o NSX-T 2.2, talvez você não consiga efetuar login no console de gerenciamento. Depois de especificar as credenciais de login, você será redirecionado para a página de login. Talvez seja necessário efetuar login várias vezes até que você consiga. Esta questão é intermitente.

- Em um ambiente NSX-T, ao reiniciar um nó principal, o console de gerenciamento se torna inacessível.

Em um ambiente NSX-T, ao reiniciar um nó principal, o console de gerenciamento fica inacessível, mesmo que todos os pods de serviço estejam em um estado bom. Esse problema é causado pelas regras NAT IPtable não persistentes, que ajudam a comunicação de porta e pod do host por meio do IP do host. O NSX-T não suporta a porta do host. O IBM Cloud Private usa a porta do host para o console de gerenciamento.

Para resolver o problema, execute os comandos a seguir em todos os nós principais. Use o CIDR de rede que você especificou no arquivo / /cluster/config.yaml.

```
iptables -tnat -N ICP-NSXT
iptables -tnat -A POSTROUTING -j ICP-NSXT
iptables -tnat -A ICP-NSXT ! -s  <network_cidr> -d <network_cidr> -j MASQUERADE
```

MustGather

1. Obtenha o status do pod do controlador do NCP e dos pods do node-agent.

   ```
   kubectl -n kube-system get pods -o wide -l tier=nsx-networking
   ```

2. Obtenha o log dos pods (o controlador do NCP e os pods do node-agent) que não estejam no estado pronto.

3. Quando o pod do node-agent está falhando em um nó, obtenha os logs do kubelet e o log ovs do nó.

   ```
   journalctl -u kubelet
   ovs-vsctl show
   ```

4. Crie o valor de configmap de `nsx-ncp-config` e `nsx-node-agent-config`.

   ```
   kubectl -n kube-system get cm nsx-ncp-config -o yaml
   kubectl -n kube-system get cm nsx-node-agent-config -o yaml
   ```

5. Crie o valor de segredo de `nsx-secrets`.

   ```
   kubectl -n kube-system get secrets nsx-secrets -o yaml
   ```

Resolução de problemas

Para evitar problemas de rede do NSX-T durante a instalação, assegure-se de que as configurações a seguir estejam configuradas corretamente.

1. Instale o plug-in CNI do NSX-T em todos os nós. Como parte da instalação, uma verificação de validação é incluída e um erro adequado é produzido.
2. Instale e configure o vSwitch aberto em todos os nós. Como parte da instalação, uma verificação de validação é incluída e um erro adequado é produzido.
3. Os recursos NSX-T obrigatórios que são fornecidos no config.yaml são configurados no gerenciador NSX-T. Se eles não estiverem configurados, o controlador do NCP não entrará no estado pronto e a instalação aguardará por kube-dns. `TASK`

```
[waitfor : Waiting for kube-dns to start]
```

4. Identifique a porta do comutador lógico com o `node name` e o `cluster name` no gerenciador NSX-T. Se a tag não estiver correta, o pod node-agent para o nó entrará no estado pronto.

Resolução de problemas do F5 BIG-IP LTM

Resolução de problemas para problemas de rede do F5 BIG-IP LTM.

Este documento é específico para as versões de gráfico:

Tabela 1. Versões do gráfico

Gráfico	versão	URL
ibm-f5bigip-controller	v1.1.0	https://github.com/IBM/charts/tree/master/stable/ibm-f5bigip-controller
ibm-calico-bgp-peer	v1.0.0	https://github.com/IBM/charts/tree/master/stable/ibm-calico-bgp-peer

1. Verifique se o dispositivo F5 BIG-IP foi incluído com sucesso como um peer BGP na malha do Calico de cluster do IBM® Cloud Private.

2. Verifique se o status da tarefa `<release-name>-calicobgppeer-addpeer` indica `SUCCESSFUL`.

```
# kubectl -n <release-namespace> get jobs
NAME                           DESIRED   SUCCESSFUL   AGE
f5peer-calicobgppeer-addpeer   1         1            7d
#
```

3. Verifique se o status do pod é `Concluído`

```
# kubectl -n <release-namespace> get po -l job-name=<job-name> (f5peer-calicobgppeer-addpeer)
NAME                                 READY     STATUS      RESTARTS   AGE
f5peer-calicobgppeer-addpeer-bq6h8   0/1       Completed   0          7d
#
```

   Se concluído com sucesso, é possível ver o log a seguir.

```
# kubectl -n <release-namespace> logs <pod-name> (f5peer-calicobgppeer-addpeer-bq6h8)
Successfully created 1 'BGPPeer' resource(s)
#
```

   Os erros de configuração do peer do Calico bgp, se houver, podem ser vistos nos logs deste pod.

   Se a tarefa falhar:

   - Verifique se o endereço IP interno fornecido está correto e se os nós no cluster estão aptos a acessar o dispositivo BIG-IP por meio dessa rede interna.
   - Verifique se o número do AS está configurado como 64512 na liberação.
   - Verifique se o `etcd endpoint` e `secret` apropriados são fornecidos. Se o `terminal etcd` ou o `secret` ou ambos estiverem errados, os logs de status do pod deverão fornecer mais informações.

4. Verifique o status do nó em um dos nós no cluster usando o utilitário `calicoctl`.

   Verifique a exatidão do `Endereço do peer`. `INFO` deve dizer `Estabelecido`.

```
# calicoctl node status
Calico process is running.
IPv4 BGP status
+---------------+-------------------+-------+------------+-------------+
|  PEER ADDRESS  |     PEER TYPE     | STATE |   SINCE    |    INFO     |
+---------------+-------------------+-------+------------+-------------+
| 192.168.70.226 | node-to-node mesh | up    | 2018-09-05 | Established |
| 192.168.70.227 | node-to-node mesh | up    | 2018-09-05 | Established |
| 192.168.70.254 | global            | up    | 2018-09-05 | Established |     <<<<<<<<<<<<<<<<<
+---------------+-------------------+-------+------------+-------------+

IPv6 BGP status
No IPv6 peers found.
#
```

5. Efetue login no F5 BIG-IP Device e verifique a configuração do vizinho bgp. Assegure-se de que todos os nós desejados no cluster tenham conectividade com o F5 BIG-IP Device. Verifique se a tabela de rotas no F5 BIG-IP Device é atualizada com

as rotas para todos os nós em seu cluster do IBM Cloud Private.

6. Verifique se o pod do controlador do F5 BIG-IP está sendo executado com sucesso e observando os recursos nos namespaces necessários

   - Verifique o status da implementação f5bigip-k8s-ctlr.

     ```
     # kubectl -n <release-namespace> get deployments
     NAME                        DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
     f5ctlr-f5bigipctlr-ctlr   1         1         1            1           7d
     #
     ```

   - Verifique os logs do f5-bigip-ctlr.

     ```
     # kubectl -n <release-namespace> get po -l app=f5bigipctlr
     NAME                                      READY     STATUS    RESTARTS   AGE
     f5ctlr-f5bigipctlr-ctlr-fd7f88bf9-j5jkq   1/1       Running   0          8d
     #

     # kubectl -n <release-namespace> logs f5ctlr-f5bigipctlr-ctlr-fd7f88bf9-j5jkq
     2018/09/05 15:36:11 [INFO] Starting: Version: v1.6.0, BuildInfo: n1156-409084177
     2018/09/05 15:36:11 [INFO] ConfigWriter started: 0xc4201cab10
     2018/09/05 15:36:11 [INFO] Started config driver sub-process at pid: 18
     2018/09/05 15:36:11 [INFO] NodePoller (0xc420572090) registering new listener: 0x407ce0
     2018/09/05 15:36:12 [INFO] NodePoller started: (0xc420572090)
     2018/09/05 15:36:12 [INFO] Registered BigIP Metrics
     2018/09/05 15:36:12 [INFO] Wrote 0 Virtual Server and 0 IApp configs
     2018/09/05 15:36:15 [INFO][2018-09-05 15:36:15,560 __main__ INFO] entering inotify loop
     to watch /tmp/k8s-bigip-ctlr.config773261415/config.json
     ...
     ...
     #
     ```

   - Certifique-se de que uma partição seja gerenciada a partir de um controlador do F5 BIG-IP k8s. O gerenciamento da mesma partição a partir de múltiplos controladores do F5 BIG-IP do Kubernetes pode levar a um comportamento inesperado.

A CLI do Helm retorna erros de conexão ou erros sobre versões do Helm incompatíveis

Após a instalação da CLI do Helm, os comandos da CLI do Helm falham com erros de conexão ou erros sobre versões incompatíveis.

Sintomas

Os comandos da CLI do Helm falham com erros de conexão ou erros sobre versões incompatíveis.

Causas

A versão do Helm Tiller que está instalada é uma versão diferente da versão 2.12.3+icp, que é instalada com o IBM Cloud Private.

Isso provavelmente aconteceu quando você configurou o cliente da linha de comandos usando os comandos `kubectl` e executou o comando `helm init` com a sinalização `--upgrade`. Ao executar o comando `helm init` sem especificar a sinalização `--client-only` ou `-c`, a CLI do Helm exibe uma mensagem de aviso. A mensagem informa que o Tiller já está implementado no cluster e que a sinalização `--upgrade` é necessária se você deseja fazer upgrade do Tiller. Se você continuar com o comando `helm init` usando a sinalização `--upgrade`, ele sobrescreverá a versão do Tiller.

É possível verificar se esse é o problema concluindo um dos procedimentos a seguir:

1. Efetue login no console do IBM Cloud Private com um ID que tenha a função **ClusterAdministrator**.
2. Navegue para **Cargas de Trabalho** > **Implementações** .
3. Localize a implementação tiller-deploy.
4. Selecione **Editar** para visualizar as informações de implementação.

Localize a sequência que mostra o caminho para a imagem do Tiller. Ela deve terminar com `tiller:v2.12.3-icp`. Se ela mostrar uma versão diferente, a imagem do Tiller foi mudada.

Também é possível concluir este procedimento:

1. Execute um dos comandos a seguir na linha de comandos:

   ```
   helm version --tls
   ```

   ou

   ```
   helm version
   ```

   Se ele for capaz de se comunicar com o Tiller, as informações sobre a versão do cliente da CLI do Helm e sobre a versão do servidor do Tiller serão exibidas.

2. Certifique-se de que a versão do servidor seja *2.12.3+icp*, e que a versão do cliente seja *2.12.3*.

   Se as versões estiverem corretas, as informações que são retornadas serão semelhantes ao exemplo a seguir:

   ```
   $ helm version --tls
   Client: &version.Version{SemVer:"v2.12.3",
   GitCommit:"8478fb4fc723885b155c924d1c8c410b7a9444e6", GitTreeState:"clean"}
   Server: &version.Version{SemVer:"v2.12.3+icp",
   GitCommit:"27442e4cfd324d8f82f935fe0b7b492994d4c289", GitTreeState:"clean"}
   ```

   Se esse comando retornar uma versão de servidor diferente, a imagem do Tiller foi mudada.

   Nota: Um cliente Helm que é mais recente que a versão necessária não é compatível com a versão 2.12.3+icp que é executada no cluster.

Resolvendo o problema

Para resolver o problema, deve-se restaurar a versão da imagem do Tiller no IBM Cloud Private 3.2.0 e no IBM Cloud Private 3.2.0 Fix Pack. Para restaurar a versão para o Tiller versão 2.12.3, conclua as seguintes etapas:

1. Configure o cliente com um usuário que tenha acesso de função **ClusterAdministrator**.

2. Execute o comando `kubectl` a seguir para listar o histórico de implementação da implementação do Tiller-Deploy no namespace do kube-system:

   ```
   kubectl rollout history deployment tiller-deploy -n kube-system
   ```

   Uma lista que contém duas ou mais revisões dessa implementação é exibida. Cada revisão é identificada por um número de revisão sequencial.

3. Execute o comando `kubectl` a seguir para exibir as informações de implementação de cada revisão, começando com a mais recente. Por exemplo:

   ```
   kubectl rollout history deployment tiller-deploy -n kube-system --revision=4
   ```

   Examine as informações de implementação que são retornadas para a referência à imagem do Tiller. O nome da imagem deve ser uma sequência que termina com "tiller:v2.12.3-icp". As informações para o número de revisão mais alto é o estado atual da implementação com a versão de imagem do Tiller incorreta. Exiba o histórico de implementação para revisões anteriores até que você localize a primeira revisão com a imagem do Tiller correta. Observe o número de revisão a ser usado na etapa 4.

4. Use o número de revisão localizado na etapa 3 para desfazer a implementação atual e reverter para a revisão correta usando o comando `kubectl rollout undo`. É possível usar o exemplo a seguir como um guia:

   ```
   kubectl rollout undo deployment tiller-deploy -n kube-system --to-revision=3
   ```

5. Verifique se a implementação foi atualizada corretamente concluindo as etapas a seguir:

   1. Navegue para **Carga de trabalho** > **Implementações** e localize a implementação *Tiller-Deploy*.
   2. Visualize as informações de implementação selecionando **Editar**.
   3. Localize a referência de imagem do Tiller nas informações de implementação e assegure-se de que ela seja a versão correta.

6. Execute o comando a seguir para efetuar login no cluster e configurar o cliente Helm:

```
cloudctl login -a https://<icp-cluster-ip>:8443 --skip-ssl-validation
```

7. Inicialize a CLI do Helm novamente usando a sinalização --client-only no comando `helm init`, conforme mostrado nos exemplos a seguir:

   ```
   helm init --client-only
   ```

   ou

   ```
   helm init -c
   ```

8. Assegure-se de que você esteja executando as versões corretas executando o comando a seguir:

   ```
   helm version --tls
   ```

Lembre-se de que o ICP 3.2.0 não suporta versões do cliente Helm mais recentes que a v2.12.3

Memória

Revisar problemas de armazenamento encontrados frequentemente.

- Resolução de problemas do GlusterFS
- Resolução de Problemas do Minio
- Resolução de Problemas do Rook Ceph
- Resolução de problemas do vSphere Cloud Provider
- Interação lenta entre o kubelet e o Docker causa problemas de PLEG

Resolução de problemas do GlusterFS

Revise problemas do GlusterFS frequentemente encontrados.

- Falha na pré-verificação de instalação do GlusterFS
- Dispositivo GlusterFS não localizado após reinicialização do sistema
- Travamento do nó GlusterFS
- A reinstalação do IBM Cloud Private não resolve problemas do GlusterFS
- Não é possível criar um PersistentVolumeClaim
- A reinicialização simultânea de nós do trabalhador faz com que o GlusterFS falhe
- Não é possível criar ou excluir um volume persistente ou solicitação de volume persistente
- O status do nó GlusterFS é mostrado como peer rejeitado
- A exclusão de uma solicitação de volume persistente do GlusterFS pode mostrar o status do volume persistente como com falha
- O pod GlusterFS não está planejado após a reinicialização de um nó
- Incompatibilidade de uso do disco do Heketi
- Pod Heketi falha ao iniciar após a reinicialização do Docker
- Pod Heketi parado no estado de inicialização quando o firewall é ativado e as portas necessárias não são abertas
- O pod GlusterFS pode falhar ao iniciar após a reinicialização de um nó do IBM® Z

Falha na pré-verificação de instalação do GlusterFS

O GlusterFS não pode ser instalado porque a pré-verificação falha.

Resolvendo o problema

- Se você estiver instalando o GlusterFS durante a instalação do IBM® Cloud Private, poderá ver uma mensagem de erro semelhante à mensagem a seguir:

  ```
  end: '2018-09-18 10 :36:59.331916'
   msg: non-zero return code
   rc: 1
   start: '2018-09-18 10:36:41.873800'
   stderr: |-
     E0918 10:36:43.815489    7463 portforward.go:316] erro ao copiar da conexão local para o
  fluxo remoto: read tcp4 127.0.0.1:38630->127.0.0.1:37206: read: conexão reconfigurada pelo peer
  ```

```
  Erro: tarefa com falha: BackoffLimitExceeded
stderr_lines: <omitted>
stdout: |-
  A liberação "storage-glusterfs" não existe. Instalando-o agora.
  ============================================================
  O log do Tiller pode ser localizado em cluster/logs/tiller-deploy-646ff69689-ww4tm
  ============================================================
```

Para identificar a razão para a falha de pré-verificação, conclua as etapas a seguir no nó principal:

1. Configure o contexto de alias kubectl.

   ```
   alias kc='kubectl --kubeconfig=/etc/cfc/conf/admin.kubeconfig -n kube-system'
   ```

2. Obtenha o nome do configmap.

   ```
   kc get configmap -l glusterfs-precheck = precheck-results-cm
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME                                                DATA      AGE
   storage-glusterfs-glusterfs-precheck-results-cm   4         10m
   ```

3. Obtenha os detalhes do configmap.

   ```
   kc describe configmap storage-glusterfs-glusterfs-precheck-results-cm
   ```

   O seguinte é uma saída de amostra:

   ```
   Nome:         storage-glusterfs-glusterfs-precheck-results-cm
   Namespace:    kube-system
   Rótulos:      app=glusterfs
                 chart=ibm-glusterfs-99.99.99
                 component=precheck-results-cm
                 glusterfs-precheck=precheck-results-cm
                 heritage=Tiller
                 release=storage-glusterfs
   Anotações:    description=Configmap de resultados de pré-verificação do GlusterFS
                 helm.sh/hook=pre-install
                 helm.sh/hook-delete-policy=before-hook-creation
                 helm.sh/hook-weight=-5

   Dados == ==
   10.41.3.19:
   ----
   status : success # msg: validação com êxito
   10.41.3.38:
   ----
   status : fail # msg: Os caminhos do dispositivo de armazenamento GlusterFS [udevice-1]
   não existem
   10.41.4.108:
   ----
   status : success # msg: validação com êxito
   precheckJobStatus:
   ----
   status : fail # msg: o nome da classe de armazenamento do GlusterFS glusterf&$s é
   inválido. O nome deve consistir em caracteres alfanuméricos minúsculos, - ou ., e deve
   iniciar e terminar com um caractere alfanumérico.
   Eventos:  <none>
   ```

   É possível identificar a razão da falha nos detalhes do Configmap.

- Se você estiver instalando o gráfico Helm do GlusterFS após a instalação do IBM® Cloud Private, execute estes comandos para identificar a causa da falha. Deve-se configurar a CLI do Kubectl para executar estes comandos. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

  1. Obtenha o nome do configmap.

     ```
     kubectl --namespace kube-system get configmap -l glusterfs-precheck=precheck-results-cm
     ```

  2. Obtenha os detalhes do configmap.

     ```
     kubectl --namespace kube-system describe configmap <configmap-name>
     ```

Dispositivo GlusterFS não localizado após reinicialização do sistema

O dispositivo que você tinha conectado para criar um volume GlusterFS não foi localizado.

Sintomas

GlusterFS se torna instável quando não vê o dispositivo correto como conectado.

Causas

Durante a reinicializações do sistema, os nomes de dispositivo podem mudar.

Resolvendo o problema

Use o link simbólico do dispositivo (symlink) como o identificador de dispositivo.

Recrie seu cluster GlusterFS. Siga as instruções na seção GlusterFS .

Reinstale o IBM® Cloud Private.

Travamentos do nó GlusterFS

Um nó do trabalhador que fazia parte do cluster GlusterFS pode travar.

É possível incluir um novo nó no cluster GlusterFS. Para obter mais informações, consulte Aumentando a capacidade de armazenamento de um cluster GlusterFS.

A reinstalação do IBM Cloud Private não resolve problemas do GlusterFS

A reinstalação do IBM® Cloud Private não resolve problemas do dispositivo GlusterFS.

Identificando o problema

Reúna informações sobre seu ambiente para identificar as razões para os problemas.

1. Execute o comando de instalação com a opção detalhada para que seja possível capturar os logs.
   - Edições Standard:

     ```
     docker run -e LICENSE=accept --net=host -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0-ee install -vvv
     ```

   - Edição da comunidade:

     ```
     docker run -e LICENSE=accept --net=host -v "$(pwd)":/installer/cluster ibmcom/icp-inception-$(uname -m | sed 's/x86_64/amd64/g'):3.2.0 install -vvv
     ```

2. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
3. Localize os pods que são relevantes para GlusterFS e Heketi.

   ```
   kubectl -n kube-system get po | grep -E 'gluster|heketi'
   ```

4. Obtenha os logs de cada um dos pods que são relevantes para GlusterFS e Heketi. Identifique os pods GlusterFS e Heketi que estão em execução.

   ```
   kubectl -n kube-system logs glusterfs-<pod name/ID>
   kubectl -n kube-system logs heketi-<pod name/ID> > heketi.txt
   kubectl -n kube-system exec glusterfs-<pod name/ID> -- cat /var/log/glusterfs/glusterd.log
   ```

5. Localize o estado do volume dos pods GlusterFS que estão em execução:

   ```
   kubectl -n kube-system exec glusterfs-<pod name/ID> -- gluster volume status
   ```

6. Localize o estado da topologia dos pods Heketi que estão em execução:

   ```
   kubectl -n kube-system exec heketi-<pod name/ID> -- heketi-cli topology info
   ```

7. Resolva os problemas:

   - Se os logs GlusterFS indicarem um problema com o acesso ao dispositivo, verifique se o caminho no host está correto e se as permissões corretas estão ativadas. Use o comando `ls -l` para confirmar se o arquivo de dispositivo existe e está acessível.
   - Se o dispositivo estiver corrompido ou se o GlusterFS não puder carregar o tijolo, substitua o mapeamento de dispositivo no GlusterFS.
   - Se o status da topologia Heketi mostrar que os volumes de GlusterFS não foram inicializados corretamente, substitua os dispositivos. Para que o Heketi seja inicializado, os dispositivos não devem ser formatados. Siga as instruções em Configurando o GlusterFS durante a instalação do IBM Cloud Private e atualize o arquivo `config.yaml`.

8. Desinstale o IBM Cloud Private. Consulte Desinstalando.
9. Instale o IBM Cloud Private. Consulte Instalando .

Não é possível criar um PersistentVolumeClaim do GlusterFS

Não é possível criar um PersistentVolumeClaim (PVC) do GlusterFS.

Causas

O nome de PVC é muito longo. Não é possível ligar um terminal de serviço PVC se seu nome tem mais de 63 caracteres. Quando GlusterFS cria um nome de terminal de serviço, ele inclui `glusterfs-dynamic` no nome de PVC e esses caracteres extras podem fazer com que o nome de PVC exceda o limite.

Por exemplo, ao criar um PVC em um StatefulSet usando `volumeClaimTemplates`, o PVC que é criado automaticamente é chamado `<pvc name>-<statefulset name>-<ordinal>`. Por exemplo, se o PVC e o StatefulSet são denominados `default-mq-stocktrader-m`, o novo PVC pode ser denominado `default-mq-stocktrader-m-default-mq-stocktrader-m-0`. Se você usar GlusterFS para criar esse PVC, o prefixo `glusterfs-dynamic-` será incluído no nome de PVC para criar um terminal em serviço. O nome do terminal em serviço `glusterfs-dynamic-default-mq-stocktrader-m-default-mq-stocktrader-m-0` excede 63 caracteres e a ligação de PVC falha.

Ao criar o PVC, você poderá ver `Status: Pending`. Também poderá ver uma mensagem semelhante à mensagem a seguir na seção `Events:` da saída de comando.

```
Service "glusterfs-dynamic-default-mq-stocktrader-m-default-mq-stocktrader-m-0" is invalid: \
metadata.name: Invalid value: \
"glusterfs-dynamic-default-mq-stocktrader-m-default-mq-stocktrader-m-0": \
must be no more than 63 characters
```

Resolvendo o problema

Se você usar GlusterFS, limite seu nome de PVC a 45 caracteres. Se você usar `volumeClaimTemplates` em StatefulSets, use nomes abreviados para o nome de StatefulSet e o nome de PVC.

Se sua ligação de PVC falhar, reduza o comprimento do nome do StatefulSet ou do nome do PVC para que o comprimento total do terminal em serviço GlusterFS, `glusterfs-dynamic-<pvc name>-<statefulset name>-<ordinal>`, não exceda 63 caracteres.

Para obter mais informações sobre esse problema, consulte o problema glusterfs create pvc falhou na comunidade do Kubernetes.

A reinicialização simultânea de nós do trabalhador faz com que o GlusterFS falhe

Quando você reinicia todos os nós do trabalhador no mesmo tempo, o GlusterFS não inicia.

Causas

Devido a uma reinicialização simultânea dos nós do trabalhador, o pod Heketi não inicia. O contêiner Heketi falha ao iniciar pois é impossível montar volumes heketidbstorage. O status de heketidbstorage é exibido como off-line porque os tijolos correspondentes não estão on-line devido a um encerramento não limpo.

Resolvendo o problema

Obtenha as informações do pod do GlusterFS executando o comando a seguir:

```
kubectl -n kube-system get pod | grep gluster
```

A seguir há um exemplo da saída de comando:

```
glusterfs-36nd0 1/1 Running 4 7d
glusterfs-3m5ql 1/1 Running 3 7d
glusterfs-tc279 1/1 Running 16 7d
```

Conclua as etapas a seguir para todos os pods do GlusterFS:

1. Efetue login no pod do GlusterFS:

   ```
   kubectl -n kube-system exec -it <POD ID> bash
   ```

   A seguir está um exemplo do comando e de sua saída:

   ```
   root@BPILICPMSTR001:~/cluster# kubectl -n kube-system exec -it glusterfs-36nd0 bash
   [root@bpilicpwrk001 /]#
   ```

2. Verifique o status do volume do GlusterFS no pod:

   ```
   gluster volume status
   ```

   A seguir está um exemplo do comando e de sua saída:

   ```
   [root@bpilicpwrk001 /]# gluster volume status
   Status of volume: heketidbstorage
   Gluster process TCP Port RDMA Port Online Pid

   Brick 10.10.25.49:/var/lib/heketi/mounts/vg
   _22bbf0fbb483f9c170774d83081c3420/brick_2fb
   3a10c7eafb8bed375829e8aaf782a/brick 49153 0 Y 5858
   Brick 10.10.25.51:/var/lib/heketi/mounts/vg
   _118f22bc13626321606280ea1d79fdc3/brick_649
   4a3b077c38667f07a59197efabea7/brick 49153 0 Y 5318
   Brick 10.10.25.50:/var/lib/heketi/mounts/vg
   _d4d4f2e86c08f571befe7fc272dc4aae/brick_dc9
   416bf4d88e45ff4d0061c08ef5b19/brick 49153 0 Y 5441
   Self-heal Daemon on localhost N/A N/A Y 5878
   Self-heal Daemon on 10.10.25.50 N/A N/A Y 5461
   Self-heal Daemon on 10.10.25.51 N/A N/A Y 5338
   Task Status of Volume heketidbstorage

   There are no active volume tasks

   [root@bpilicpwrk001 /]#
   ```

   Se os tijolos correspondentes ao heketidbstorage estiverem inativos, reinicie os tijolos executando os comandos a seguir:

   ```
   gluster volume stop heketidbstorage

   gluster volume start heketidbstorage force
   ```

3. Verifique o status do pod Heketi:

   ```
   kubectl -n kube-system get pod | grep heketi
   ```

   O status exibe uma mensagem semelhante à mensagem a seguir:

   ```
   heketi-402978595-pjnd7 1/1 Running 0 2h
   ```

Não é possível criar ou excluir um volume persistente ou solicitação de volume persistente

Não é possível criar ou excluir um PersistentVolume (PV) ou PersistentVolumeClaim (PVC).

Se estiver usando o armazenamento GlusterFS, quando tentar criar ou excluir um PV ou PVC, você poderá ver a mensagem `database is in read-only mode`.

Causas

- Se você reiniciou o pod do Heketi, o pod não obteve acesso `write` ao arquivo de banco de dados do Heketi.
- Um nó do cluster GlusterFS está inativo.

Resolvendo o problema

Conclua estas etapas para resolver o problema:

1. Assegure-se de que todos os nós do cluster do GlusterFS estejam ativos e em execução.
2. Assegure-se de que a CLI `kubectl` esteja configurada. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
3. Diminua a implementação do Heketi.

   a. Identifique o nome da implementação do Heketi executando o comando a seguir:

   ```
   kubectl -- namespace = kube-system get deployments -l glusterfs=heketi-deployment
   ```

   A maioria das configurações está executando zero ou uma implementação do Heketi. Use o nome da implementação que é retornado para as etapas restantes que requerem que você especifique o nome da implementação.

   b. Escale a implementação do Heketi para 0 inserindo o comando a seguir:

   ```
   kubectl scale --namespace=kube-system deploy -l glusterfs=heketi-deployment --replicas=0
   deployment.extensions "heketi-deployment" scaled
   ```

   c. Aguarde o pod para finalizar. Execute o comando a seguir para visualizar o status do pod:

   ```
   kubectl -- namespace = kube-system get pods -l glusterfs=heketi-pod
   ```

   Se o pod é finalizado com êxito, o comando não retorna saída.

4. Aumente a implementação do Heketi.

   a. Aumente o número de instâncias do Heketi inserindo o comando a seguir:

   ```
   kubectl scale --namespace=kube-system deploy -l glusterfs=heketi-deployment --replicas=1
   ```

   b. Valide o aumento da implementação inserindo o comando a seguir:

   ```
   kubectl --namespace=kube-system rollout status deployments [name-of-your-deployment]
   ```

   Após a implementação ser apresentada com sucesso, a saída é semelhante ao código a seguir:

   ```
   implementação "heketi" apresentada com sucesso
   ```

   c. Aguarde o pod para iniciar. Execute o comando a seguir para visualizar o status do pod:

   ```
   kubectl -- namespace = kube-system get pods -l glusterfs=heketi-pod
   ```

   Depois que o pod é iniciado, a saída é semelhante ao código a seguir:

   ```
   heketi-68549fdf65-sm8tl                                  1/1       Running
   0          23s
   ```

O status do nó GlusterFS é mostrado como peer rejeitado

O status do nó GlusterFS pode ser mostrado como "Peer rejeitado".

Nota: se múltiplos nós estiverem em um estado "Peer rejeitado", você poderá não ser capaz de recuperar os nós do GlusterFS. Você pode precisar configurar o cluster do GlusterFS novamente.

Resolvendo o problema

1. Assegure-se de que a CLI `kubectl` esteja configurada. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

2. Verifique o status de pods GlusterFS.

   ```
   kubectl -n kube-system get pods -o wide | grep gluster
   ```

   A saída se assemelha ao código a seguir:

   ```
   glusterfs-l95bp        1/1       Running     3          6d        192.168.0.184   worker-2
   glusterfs-n85vr        1/1       Running     0          4d        192.168.0.56    worker-1
   glusterfs-p66jq        1/1       Running     3          6d        192.168.0.96    worker-3
   ```

3. Verifique o status do peer de qualquer pod do GlusterFS.

   ```
   kubectl -n kube-system exec <pod name> -- gluster peer status
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec glusterfs-l95bp -- gluster peer status
   Number of Peers: 2

   Hostname: 192.168.0.96
   Uuid: 3e518a43-b59f-45e7-a62c-5b213e0fece8
   State: Peer in Cluster (Connected)
   Other names:
   192.168.0.96

   Hostname: worker-1
   Uuid: c2a81937-e94f-4a22-86e7-bc9c8929c2d3
   State: Peer Rejected (Connected)
   ```

   Se você vir um status "Peer rejeitado", isso indica que a configuração do volume nesse peer está fora de sincronização com o resto do cluster.

Para sincronizar com o cluster do GlusterFS, conclua estas etapas no nó do GlusterFS que está no estado "Peer rejeitado":

1. Acesse o shell dentro do pod do GlusterFS:

   ```
   kubectl -n kube-system exec -it <pod name> bash
   ```

2. Mude para o `/var/lib/glusterd` diretório.

   ```
   Cd /var/lib/glusterd
   ```

   A pasta pode conter os arquivos e pastas a seguir:

   ```
   .  ..  bitd  geo-replication  glusterd.info  glusterfind  glustershd  groups  hooks  nfs
   options  peers  quotad  scrub  snaps  ss_brick  vols
   ```

3. Exclua tudo, exceto `glusterd.info`, que é o arquivo de identificador exclusivo universal (UUID).

4. Reinicie o daemon do Gluster.

   ```
   Reinicie glusterd de serviço
   ```

5. Analise um peer que não esteja em um estado "Peer rejeitado".

   ```
   gluster peer probe <node name>
   ```

6. Verifique o status de peer.

   ```
   Gluster peer status
   ```

Note: pode ser necessário repetir as etapas até você não veja mais o status "Peer rejeitado".

A exclusão de uma solicitação de volume persistente do GlusterFS pode mostrar o status do volume persistente como com falha

Um status de PersistentVolume (PV) do GlusterFS é mostrado como "Com falha" quando você exclui o PersistentVolumeClaim (PVC) que está ligado a ele.

Veja os comandos e a saída de exemplo a seguir:

Antes de continuar, configure a CLI `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

1. Obtenha uma lista de todos os PVCs.

   ```
   Kubectl get pvc
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                      STATUS    VOLUME                                     CAPACITY   ACCESS
   MODES   STORAGECLASS          AGE
   gfs-endpoint-test         Bound     pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO
   gluster              22h
   noreplica-pvc-gluster     Bound     pvc-08f87fa7-542f-11e8-89ac-005056a8640c   1Gi        RWO
   gluster-no-replica   5d
   test-pvc-gluster-r2-1     Bound     pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO
   gluster-replica2     5d
   ```

2. Excluir um PVC.

   ```
   kubectl delete pvc <PVC name>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl delete pvc noreplica-pvc-gluster
   persistentvolumeclaim "my-release-grafana" deleted
   ```

3. Verifique o status do PVCs.

   ```
   Kubectl get pvc
   ```

   A saída é semelhante ao código a seguir. O PVC é excluído com êxito.

   ```
   NAME                      STATUS    VOLUME                                     CAPACITY   ACCESS
   MODES   STORAGECLASS          AGE
   gfs-endpoint-test         Bound     pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO
   gluster              22h
   test-pvc-gluster-r2-1     Bound     pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO
   gluster-replica2     5d
   ```

4. Verifique o status do VFs. O status do PV ao qual o PVC excluído foi ligado pode ser mostrado como "Com falha".

   ```
   kubectl get pv
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
   CLAIM                                       STORAGECLASS               REASON    AGE
   pvc-08f87fa7-542f-11e8-89ac-005056a8640c   1Gi        RWO            Delete           Failed
   default/noreplica-pvc-gluster               gluster-no-replica                  5d
   pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO            Delete           Bound
   default/test-pvc-gluster-r2-1               gluster-replica2                    5d
   pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO            Delete           Bound
   default/gfs-endpoint-test                   gluster                             22h
   ```

Causas

Um nó do GlusterFS está inativo ou nem todos os pods do GlusterFS estão em um estado de execução.

Resolvendo o problema

1. Verifique se todos os nós do GlusterFS estão em execução e são registrados com o cluster do IBM Cloud Private.

   ```
   kubectl get nodes
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME       STATUS    ROLES     AGE       VERSION
   master     Ready     <none>    5d        v1.11.0+icp-ee
   ```

```
worker-1   Ready     <none>    5d        v1.11.0+icp-ee
worker-2   Ready     <none>    5d        v1.11.0+icp-ee
worker-3   Ready     <none>    5d        v1.11.0+icp-ee
```

2. Assegure-se de que o status de todos os pods do GlusterFS seja mostrado como "Em execução". Além disso, verifique se o número de pods é o mesmo que o número de nós do GlusterFS que você configurou.

   ```
   kubectl -n kube-system get pods | grep gluster
   ```

   A saída se assemelha ao código a seguir:

   ```
   glusterfs-l95bp                                                  1/1       Running     1
   5d
   glusterfs-n85vr                                                  1/1       Running     0
   3d
   glusterfs-p66jq                                                  1/1       Running     3
   5d
   ```

   Se a contagem de nós e pods não corresponder, verifique os rótulos dos nós. Todos os nós do GlusterFS devem ter o rótulo `storagenode=glusterfs`.

   ```
   kubectl get nodes --show-labels
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                 STATUS    ROLES     AGE       VERSION          LABELS
   master     Ready     <none>    6d        v1.11.0+icp-ee
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,etcd=true,gpu/nvidia=NA,kubernetes.io
   /hostname=master,management=true,master=true,proxy=true,role=master
   worker-1   Ready     <none>    6d        v1.11.0+icp-ee
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
   worker-1,storagenode=glusterfs
   worker-2   Ready     <none>    6d        v1.11.0+icp-ee
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
   worker-2,storagenode=glusterfs
   worker-3   Ready     <none>    6d        v1.11.0+icp-ee
   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,gpu/nvidia=NA,kubernetes.io/hostname=
   worker-3,storagenode=glusterfs
   ```

3. Para todos os nós do GlusterFS que não possuem um rótulo `storagenode=glusterfs`, inclua o rótulo.

   ```
   Kubectl label node < nós name > storagenode=glusterfs
   ```

4. Verifique o status dos pods do GlusterFS e Heketi.

   ```
   kubectl -n kube-system get pods | egrep "gluster|heketi"
   ```

   A saída se assemelha ao código a seguir:

   ```
   glusterfs-l95bp               1/1       Running     1          5d
   glusterfs-n85vr               1/1       Running     0          3d
   glusterfs-p66jq               1/1       Running     3          5d
   heketi-9f984d759-zcmvb        1/1       Running     0          4d
   ```

5. Quando todos os pods mostrarem o status como "Em execução", verifique o status do PV que foi mostrado como "Com calha". **Nota:** o PV será excluído se a política de recuperação foi "Excluir". Além disso, o status do PV é mostrado como "Liberado".

   ```
   kubectl get pv
   ```

   A saída se assemelha ao código a seguir:

   ```
   NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
   CLAIM                                       STORAGECLASS               REASON    AGE
   pvc-7ab00b20-5394-11e8-89ac-005056a8640c   1Gi        RWO            Delete           Bound
   default/test-pvc-gluster-r2-1               gluster-replica2                    5d
   pvc-812dd5df-575f-11e8-9b8b-005056a8640c   1Gi        RWO            Delete           Bound
   default/gfs-endpoint-test                   gluster                             22h
   ```

O pod do GlusterFS não está planejado após a reinicialização de um nó

Ao reiniciar um nó GlusterFS, o pod GlusterFS não é planejado.

Causas

O rótulo `storagenode=glusterfs` é perdido durante uma reinicialização.

Resolvendo o problema

1. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Rotule o nó.

   ```
   kubectl label nodes <node IP address> storagenode=glusterfs
   ```

3. Atualize o arquivo de manifesto do serviço do kubelet para tornar a etiqueta persistente entre as reinicializações do sistema.
   1. Abra o arquivo `/etc/systemd/system/kubelet.service` para edição.
   2. Inclua a seguinte parte do código na seção `[Service]`:

      ```
      --node-labels=storagenode=glusterfs\
      ```

      Nota: se outros rótulos forem listados, retenha-os. Inclua o rótulo `storagenode=glusterfs` na lista de rótulos. Por exemplo: `--node-labels=disktype=ssd, foo=bar, storagenode=glusterfs\`.

      Após incluir o código, o conteúdo do arquivo é semelhante ao texto a seguir:

      ```
      [Service] EnvironmentFile=-/etc/environment ExecStart=/opt/kubernetes/hyperkube kubelet \
      ...
      --node-labels=storagenode=glusterfs\
      ...
      ```

4. Verifique o status do pod.

   ```
   kubectl -n kube-system get po -owide | grep -E "gluster|heketi"
   ```

   Se o status for mostrado como `ContainerCreating`, exclua o pod. Quando o pod é recriado, ele é exibido em um estado `Em execução`.

Incompatibilidade do uso do disco Heketi

Incompatibilidade do espaço em disco real e do espaço em disco usado que é relatado por Heketi.

Ao criar uma solicitação de volume persistente (PVC), é possível ver uma mensagem de erro de que nenhum espaço em disco está disponível.

Resolvendo o problema

Se você vir uma incompatibilidade de uso do disco nas informações de topologia do Heketi, execute estes comandos para sincronizar o uso do disco.

1. Configure a CLI kubectl. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. Obtenha o nome do pod de implementação Heketi.

   ```
   kubectl -n kube-system get pods | grep heketi
   ```

   A seguir está uma saída de exemplo do comando:

   ```
   storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf   1/1       Running     0
   18h
   ```

3. Efetue login no pod Heketi e obtenha a lista de nós.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   node list
   ```

   A seguir está uma saída de exemplo do comando:

   ```
   Id:10cc13f4cd18136fc9b6d2a6d1eac733     Cluster:34fe224a09d8022a3582da817c31a81b
   Id:157012bb7b0391b2c116b35de8d6e7ba     Cluster:34fe224a09d8022a3582da817c31a81b
   ```

```
Id:95c72d452bc2c21b3a1e2fa87e63f902     Cluster:34fe224a09d8022a3582da817c31a81b
```

4. Obtenha informações sobre os dispositivos nos nós. Repita esta etapa para cada nó.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   node info <node-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin node info 10cc13f4cd18136fc9b6d2a6d1eac733
   Node Id: 10cc13f4cd18136fc9b6d2a6d1eac733
   State: online
   Cluster Id: 34fe224a09d8022a3582da817c31a81b
   Zone: 1
   Management Hostname: 10.41.4.108
   Storage Hostname: 10.41.4.108
   Devices:
   Id:c7e93cd3d3d293a78dfb99cb2809f699   Name:/dev/disk/by-path/virtio-pci-0000_00_11_0
   State:online    Size (GiB):699      Used (GiB):497       Free (GiB):202
              Bricks:
                        Id:33cb391e0113ec96ccc546a4e4288018   Size (GiB):100     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_33cb391e0113ec96ccc546a4e42880
   18/brick
                        Id:41a508db035b042b1d35839f0d40f0c5   Size (GiB):20     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_41a508db035b042b1d35839f0d40f0
   c5/brick
   ```

5. Obtenha informações sobre o uso do disco nos dispositivos. Repita esta etapa para cada dispositivo em cada nó.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   device info <device-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin device info c7e93cd3d3d293a78dfb99cb2809f699
   Device Id: c7e93cd3d3d293a78dfb99cb2809f699
   Name: /dev/disk/by-path/virtio-pci-0000_00_11_0
   State: online
   Size (GiB): 699
   Used (GiB): 497
   Free (GiB): 202
   Bricks:
   Id:33cb391e0113ec96ccc546a4e4288018   Size (GiB):100     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_33cb391e0113ec96ccc546a4e42880
   18/brick
   Id:41a508db035b042b1d35839f0d40f0c5   Size (GiB):20     Path:
   /var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_41a508db035b042b1d35839f0d40f0
   c5/brick
   ```

6. Sincronize o dispositivo para refletir o uso real do disco. Repita esta etapa para cada dispositivo em cada nó.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   device resync <device-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin device resync c7e93cd3d3d293a78dfb99cb2809f699
   Device  updated
   ```

7. Verifique se as informações de uso do disco estão sincronizadas.

   ```
   kubectl -n kube-system exec -it <Heketi-pod-name> -- heketi-cli --user admin --secret admin
   device info <device-ID>
   ```

   O seguinte é um exemplo de comando e de saída:

   ```
   kubectl -n kube-system exec -it storage-glusterfs-glusterfs-heketi-deployment-85844b495f-wbxhf
   -- heketi-cli --user admin --secret admin device info c7e93cd3d3d293a78dfb99cb2809f699
   Device Id: c7e93cd3d3d293a78dfb99cb2809f699
   Name: /dev/disk/by-path/virtio-pci-0000_00_11_0
   State: online
   Size (GiB): 699
   ```

```
Used (GiB): 120
Free (GiB): 579
Bricks:
Id:33cb391e0113ec96ccc546a4e4288018   Size (GiB):100     Path:
/var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_33cb391e0113ec96ccc546a4e42880
18/brick
Id:41a508db035b042b1d35839f0d40f0c5   Size (GiB):20     Path:
/var/lib/heketi/mounts/vg_d991f7315d9e13f9ef4d8044cd474569/brick_41a508db035b042b1d35839f0d40f0
c5/brick
```

O pod Heketi falha ao iniciar após o reinício do docker

O pod Heketi falha ao iniciar após a reinicialização do Docker devido a um problema com o Docker Versão 18.03.1. Você vê o erro a seguir do serviço kubelet para o pod Heketi quando você reinicia o Docker:

```
Error: failed to start container "heketi": Error response from daemon: OCI runtime create failed:
container_linux.go:348: \
starting container process caused "process_linux.go:402: container init caused \"rootfs_linux.go:58:
\
mounting \\\"/var/lib/kubelet/pods/7e9cb34c-b2bf-11e8-a9eb-0050569bdc9f/volume-subpaths/heketi-db-
secret/heketi/0\\\" \
to rootfs
\\\"/var/lib/docker/overlay2/ca0a54812c6f5718559cc401d9b73fb7ebe43b2055a175ee03cdffaffada2585/merged
\\\" at \
\\\"/var/lib/docker/overlay2/ca0a54812c6f5718559cc401d9b73fb7ebe43b2055a175ee03cdffaffada2585/merged
/backupdb/heketi.db.gz\\\" \
caused \\\"no such file or directory\\\"\"": unknown
```

Para resolver o problema, primeiro diminua a capacidade, depois aumente a capacidade da implementação do Heketi.

1. Instale a CLI `kubectl` . Consulte Instalando a CLI do Kubernetes (kubectl) .
2. Diminua a implementação do Heketi.

   a. Identifique o nome da implementação do Heketi executando o comando a seguir:

   ```
   kubectl -- namespace = kube-system get deployments -l glusterfs=heketi-deployment
   ```

   A maioria das configurações está executando zero ou uma implementação do Heketi. Use o nome da implementação nas etapas que requerem que você especifique o nome da implementação.

   b. Escale a implementação do Heketi para 0 inserindo o comando a seguir:

   ```
   kubectl scale --namespace=kube-system deploy -l glusterfs=heketi-deployment --replicas=0
   deployment.extensions "heketi-deployment" scaled
   ```

   c. Aguarde o pod para finalizar. Execute o comando a seguir para visualizar o status do pod:

   ```
   kubectl -- namespace = kube-system get pods -l glusterfs=heketi-pod
   ```

   Se o pod é finalizado com êxito, o comando não retorna saída.

3. Aumente a implementação do Heketi.

   a. Aumente o número de instâncias do Heketi inserindo o comando a seguir:

   ```
   kubectl scale --namespace=kube-system deploy -l glusterfs=heketi-deployment --replicas=1
   ```

   b. Valide o aumento da implementação inserindo o comando a seguir:

   ```
   kubectl --namespace=kube-system rollout status deployments [name-of-your-deployment]
   ```

   Após a implementação ser apresentada com sucesso, a saída é semelhante ao código a seguir:

   ```
   implementação "heketi" apresentada com sucesso
   ```

   c. Aguarde o pod para iniciar. Execute o comando a seguir para visualizar o status do pod:

   ```
   kubectl -- namespace = kube-system get pods -l glusterfs=heketi-pod
   ```

   Depois que o pod é iniciado, a saída é semelhante ao código a seguir:

```
heketi-68549fdf65-sm8tl                                   1/1       Running
0          23s
```

Pod Heketi parado no estado de inicialização quando o firewall é ativado e as portas necessárias não são abertas

O pod Heketi permanece no estado inicializando. Você vê a seguinte instrução de erro nos logs do contêiner `init-heketi`:

```
Falha ao executar o comando [gluster peer probe x.x.x.x ]
```

Para visualizar os logs, conclua as etapas a seguir:

1. Instale a CLI `kubectl` . Consulte Instalando a CLI do Kubernetes (kubectl) .
2. Obtenha os pods Heketi.

   ```
   kubectl -n kube-system get pods | grep heketi
   ```

   O seguinte é uma saída de amostra:

   ```
   storage-glusterfs-glusterfs-heketi-deployment-77fd4dbcb8-twhhq   0/1     Init:1/3            0
   3m38s
   ```

3. Obtenha os logs.

   ```
   kubectl -n kube-system logs <Heketi-pod-name> -c init-heketi
   ```

   O seguinte é uma saída de amostra:

   ```
   [kubeexec] ERROR 2019/02/06 09:16:42
   /src/github.com/heketi/heketi/executors/kubeexec/kubeexec.go:242: Failed to run command \
   [gluster peer probe 10.41.13.146] on storage-glusterfs-glusterfs-daemonset-qvhzq: Err[command
   terminated with exit code 1]: \
   Stdout [Error : Request timed out
   ]: Stderr []
   [asynchttp] INFO 2019/02/06 09:16:42 asynchttp.go:292: Completed job
   b961f40b211ecbe320b841a0fc0b511f in 2m0.831981756s
   [negroni] Started GET /queue/b961f40b211ecbe320b841a0fc0b511f
   [negroni] Completed 500 Internal Server Error in 410.049µs
   ```

Causas

O pod Heketi permanece no estado de inicialização quando o sinalizador `firewall_enabled` é configurado como `true` no `config.yaml` e as portas necessárias não são abertas nos nós de armazenamento do GlusterFS. Para obter mais informações sobre as portas necessárias, consulte Portas necessárias.

Resolvendo o problema

Você deve abrir manualmente todas as portas necessárias em todos os nós de armazenamento GlusterFS.

A seguir estão os comandos para abrir as portas necessárias:

- Para Ubuntu, execute os comandos a seguir:

  ```
  ufw allow 24007:24008/tcp
  ufw allow 49152:49251/tcp
  ufw allow 2222/tcp
  ufw reload
  ```

- Para o Red Hat Enterprise Linux (RHEL), execute os seguintes comandos:

  ```
  firewall-cmd  --add-port=24007-24008/tcp --permanent
  firewall-cmd  --add-port=49152-49251/tcp --permanent
  firewall-cmd  --add-port=2222/tcp --permanent
  firewall-cmd --reload
  ```

Para obter mais informações, consulte Configurando o GlusterFS durante a instalação do IBM Cloud Private.

O pod GlusterFS pode falhar ao iniciar após reiniciar um nó do IBM® Z

Se qualquer nó do cluster GlusterFS do IBM® Z for reinicializado, o daemonset `glusterfs` se recuperará automaticamente. No entanto, é possível observar que o daemonset GlusterFS no nó reinicializado reinicia continuamente.

Resolvendo o problema

1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
2. Localize o nó GlusterFS no qual o daemonset GlusterFS é reinicializado continuamente.

   ```
   kubectl get po -owide -n kube-system | grep glusterfs
   ```

3. Efetue login no nó GlusterFS com falha e faça backup do diretório `/var/lib/heketi`.

   ```
   mv /var/lib/heketi /var/lib/heketi-bkp
   ```

4. Exclua o pod que reinicia continuamente.

   ```
   kubectl delete pod <pod> -n kube-system
   ```

5. Recupere os dados `heketi` depois que o daemonset `glusterfs` iniciar a execução.

   ```
   cp -r /var/lib/heketi-bkp/* /var/lib/heketi
   ```

Nota: se o daemonset `glusterfs` no nó com falha não puder ser iniciado, ainda será possível recuperar os dados `heketi`. Depois de recuperar os dados, inclua um nó GlusterFS e, em seguida, exclua o nó com falha.

Resolução de problemas do Minio

Revise problemas do Minio encontrados frequentemente.

- Reunindo informações
- Os pods do Minio travam com o status ContainerCreating
- O pod do servidor Minio trava no STATUS Pendente
- O Minio no modo distribuído não é acessível ao fornecer um certificado TLS
- Os depósitos e objetos Minio estão intermitentemente inacessíveis

Reunindo informações

Reúna informações para resolução de problemas do Minio.

Para resolução de problemas, deve-se reunir as informações a seguir:

Nota: é necessário configurar a CLI do kubectl para executar esses comandos. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

1. Versão do IBM Cloud Private.
2. Tipo de arquitetura dos nós em seu cluster. Por exemplo, Linux® ou Linux® on Power® (ppc64le)
3. Versão do gráfico Helm do Minio que você instalou. Use o comando a seguir:

   ```
   helm list --tls | grep mini
   ```

   O código a seguir é uma saída de amostra:

   ```
   minio                           1           Fri Sep 14 05:10:28 2018    DEPLOYED    ibm-minio-
   objectstore-1.6.0            default
   ```

4. Estado de implementação do Minio ou estado statefulset. Use os comandos a seguir:

   ```
   kubectl get statefulsets
   ```

   O código a seguir é uma saída de amostra:

```
NAME                          DESIRED   CURRENT   AGE
minio-ibm-minio-objectstore   4         4         2m

kubectl describe  statefulsets
```

O código a seguir é uma saída de amostra:

```
Name:               minio-ibm-minio-objectstore
Namespace:          default
CreationTimestamp:  Fri, 14 Sep 2018 05:10:37 -0700
Selector:           app=ibm-minio-objectstore,release=minio
Labels:             app=ibm-minio-objectstore
                   chart=ibm-minio-objectstore-1.6.0
                   heritage=Tiller
...
```

5. Status do serviço do Minio. Use os comandos a seguir:

    1. Obtenha o serviço.

    ```
    kubectl get svc
    ```

    O código a seguir é uma saída de amostra:

    ```
    NAME                          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
    kubernetes                    ClusterIP   10.0.0.1     <none>        443/TCP    10d
    minio-ibm-minio-objectstore   ClusterIP   10.0.0.68    <none>        9000/TCP   4m
    ```

    2. Obtenha a descrição do serviço.

    ```
    kubectl describe svc
    ```

    O código a seguir é uma saída de amostra:

    ```
    Name:              kubernetes
    Namespace:         default
    Labels:            component=apiserver
                      provider=kubernetes
    Annotations:       <none>
    Selector:          <none>
    Type:              ClusterIP
    IP:                10.0.0.1
    Port:              https  443/TCP
    TargetPort:        8001/TCP
    Endpoints:         10.41.1.182:8001
    Session Affinity:  None
    Events:            <none>
    .
    .
    Name:              minio-ibm-minio-objectstore
    Namespace:         default
    Labels:            app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-1.6.0
                      heritage=Tiller
                      release=minio
    Annotations:       prometheus.io/path=/minio/prometheus/metrics
                      prometheus.io/port=9000
                      prometheus.io/scrape=false
    Selector:          app=ibm-minio-objectstore,release=minio
    Type:              ClusterIP
    IP:                10.0.0.68
    Port:              service  9000/TCP
    TargetPort:        9000/TCP
    Endpoints:         10.1.137.211:9000,10.1.180.251:9000,10.1.236.84:9000 + 1 more...
    Session Affinity:  None
    Events:            <none>
    ```

6. Os pods, os logs e a descrição do servidor Minio.

    1. Obtenha todos os pods do Minio.

    ```
    kubectl get po | grep mini
    ```

    O código a seguir é uma saída de amostra:

```
minio-ibm-minio-objectstore-0         1/1       Running   0          4m
minio-ibm-minio-objectstore-1         1/1       Running   0          4m
minio-ibm-minio-objectstore-2         1/1       Running   0          4m
minio-ibm-minio-objectstore-3         1/1       Running   0          4m
```

2. Obtenha logs e descrição de todos os pods. O código a seguir é um exemplo de comando:

```
kubectl describe po minio-ibm-minio-objectstore-0
```

O código a seguir é uma saída de amostra:

```
Name:               minio-ibm-minio-objectstore-0
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               10.41.4.202/10.41.4.202
Start Time:         Fri, 14 Sep 2018 05:10:37 -0700
Labels:             app=ibm-minio-objectstore
                  chart=ibm-minio-objectstore-1.6.0
                  controller-revision-hash=minio-ibm-minio-objectstore-7b77fd5658
                  heritage=Tiller
                  release=minio
                  statefulset.kubernetes.io/pod-name=minio-ibm-minio-objectstore-0
Annotations:        kubernetes.io/psp=00-rook-ceph-operator
                  productID=Minio_RELEASE.2018-08-21T00-37-20Z_free_00000
                  productName=Minio
                  productVersion=RELEASE.2018-08-21T00-37-20Z
                  scheduler.alpha.kubernetes.io/critical-pod=
Status:             Running
IP:                 10.1.236.84
Controlled By:      StatefulSet/minio-ibm-minio-objectstore
Containers:
ibm-minio-objectstore:
  Container ID:
docker://5e71782564d1c956d6855006f06472773da59ad22743a52bb64f83f4ac0ccf02
  Image:         minio/minio:RELEASE.2018-08-21T00-37-20Z
  Image ID:      docker-
pullable://minio/minio@sha256:3145ff901d491f46e59dd9fb79dc2771e75a524bbfdba8fa8cd35723960
fe7d5
  Port:          9000/TCP
  Host Port:     0/TCP
  Command:
    /bin/sh
    -ce
    cp /tmp/config.json /root/.minio/ && /usr/bin/docker-entrypoint.sh minio -C
/root/.minio/ server http://minio-ibm-minio-objectstore-0.minio-ibm-minio-
objectstore.default.svc.cluster.local/export http://minio-ibm-minio-objectstore-1.minio-
ibm-minio-objectstore.default.svc.cluster.local/export http://minio-ibm-minio-
objectstore-2.minio-ibm-minio-objectstore.default.svc.cluster.local/export http://minio-
ibm-minio-objectstore-3.minio-ibm-minio-objectstore.default.svc.cluster.local/export
  State:          Running
    Started:      Fri, 14 Sep 2018 05:10:39 -0700
  Ready:          True
  Restart Count:  0
  Requests:
    cpu:     250m
    memory:  256Mi
  Environment:
    MINIO_ACCESS_KEY:  <set to the key 'accesskey' in secret 'minio'>  Optional: false
    MINIO_SECRET_KEY:  <set to the key 'secretkey' in secret 'minio'>  Optional: false
  Mounts:
    /export from export (rw)
    /root/.minio/ from minio-config-dir (rw)
    /tmp/config.json from minio-server-config (rw)
    /var/run/secrets/kubernetes.io/serviceaccount from default-token-tzxvl (ro)
Conditions:
Type              Status
Initialized       True
Ready             True
ContainersReady   True
PodScheduled      True
Volumes:
export:
  Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same
namespace)
```

```
  ClaimName:  export-minio-ibm-minio-objectstore-0
  ReadOnly:   false
minio-user:
  Type:        Secret (a volume populated by a Secret)
  SecretName:  minio
  Optional:    false
minio-server-config:
  Type:      ConfigMap (a volume populated by a ConfigMap)
  Name:      minio-ibm-minio-objectstore
  Optional:  false
minio-config-dir:
  Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
default-token-tzxvl:
  Type:        Secret (a volume populated by a Secret)
  SecretName:  default-token-tzxvl
  Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     CriticalAddonsOnly
                 dedicated
                 node.kubernetes.io/memory-pressure:NoSchedule
Events:
Type    Reason     Age   From                 Message
----    ------     ----  ----                 -------
Normal  Scheduled  5m    default-scheduler    Successfully assigned default/minio-ibm-
minio-objectstore-0 to 10.41.4.202
Normal  Pulled     5m    kubelet, 10.41.4.202  Container image "minio/minio:RELEASE.2018-
08-21T00-37-20Z" already present on machine
Normal  Created    5m    kubelet, 10.41.4.202  Created container
Normal  Started    5m    kubelet, 10.41.4.202  Started container
```

7. Informações sobre a solicitação de volume persistente (PVC), se você usou o fornecimento de armazenamento dinâmico.

    1. Obtenha todos os PVCs.

        ```
        Kubectl get pvc
        ```

        O código a seguir é uma saída de amostra:

        ```
        NAME                                    STATUS    VOLUME
        CAPACITY   ACCESS MODES   STORAGECLASS        AGE
        export-minio-ibm-minio-objectstore-0   Bound     pvc-a35afd44-b811-11e8-bc28-00000a2901b6
        5Gi        RWO            rook-ceph-block   46m
        export-minio-ibm-minio-objectstore-1   Bound     pvc-a71ea92b-b811-11e8-bc28-00000a2901b6
        5Gi        RWO            rook-ceph-block   46m
        export-minio-ibm-minio-objectstore-2   Bound     pvc-ab6f00af-b811-11e8-bc28-00000a2901b6
        5Gi        RWO            rook-ceph-block   46m
        export-minio-ibm-minio-objectstore-3   Bound     pvc-b27b35fc-b811-11e8-bc28-00000a2901b6
        5Gi        RWO            rook-ceph-block   45m
        ```

    2. Obtenha informações sobre um PVC. O código a seguir é um comando de amostra:

        ```
        kubectl describe pvc export-minio-ibm-minio-objectstore-0
        ```

        O código a seguir é uma saída de amostra:

        ```
        Name:          export-minio-ibm-minio-objectstore-0
        Namespace:     default
        StorageClass:  rook-ceph-block
        Status:        Bound
        Volume:        pvc-a35afd44-b811-11e8-bc28-00000a2901b6
        Labels:        app=ibm-minio-objectstore
                       release=minio
        Annotations:   control-plane.alpha.kubernetes.io/leader={"holderIdentity":"ee888338-b654-
        11e8-86f0-16fe371b5da0","leaseDurationSeconds":15,"acquireTime":"2018-09-
        14T11:30:52Z","renewTime":"2018-09-14T11:30:57Z","lea...
                       pv.kubernetes.io/bind-completed=yes
                       pv.kubernetes.io/bound-by-controller=yes
                       volume.beta.kubernetes.io/storage-provisioner=ceph.rook.io/block
        Finalizers:    [kubernetes.io/pvc-protection]
        Capacity:      5Gi
        Access Modes:  RWO
        Events:
        Type    Reason                 Age                From
        ```

```
Message
----    ------                  ----               ----
-------
Normal  Provisioning            46m                ceph.rook.io/block rook-ceph-operator-
5f84847c67-c6nzl ee888338-b654-11e8-86f0-16fe371b5da0  External provisioner is
provisioning volume for claim "default/export-minio-ibm-minio-objectstore-0"
Normal  ExternalProvisioning    46m (x2 over 46m)  persistentvolume-controller
waiting for a volume to be created, either by external provisioner "ceph.rook.io/block"
or manually created by system administrator
Normal  ProvisioningSucceeded   46m                ceph.rook.io/block rook-ceph-operator-
5f84847c67-c6nzl ee888338-b654-11e8-86f0-16fe371b5da0  Successfully provisioned volume
pvc-a35afd44-b811-11e8-bc28-00000a2901b6
```

Os pods do Minio travam com o status ContainerCreating

Quando o Minio é implementado em qualquer modo, o primeiro pod do Minio pode travar com o status *ContainerCreating*.

Reúna informações sobre o problema

1. Obtenha a lista de pods.

   ```
   kubectl get po
   ```

   O código a seguir é uma saída de amostra:

   ```
   NAME                                              READY     STATUS              RESTARTS   AGE
   mc2                                               1/1       Running             53         2d
   minio-ibm-minio-objectstore-848fbcb6f5-2wpq2      0/1       ContainerCreating   0          3m
   ```

2. Verifique os logs. Se os logs estiverem vazios, descreva o pod.

   ```
   kubectl logs minio-ibm-minio-objectstore-848fbcb6f5-2wpq2
   ```

   O código a seguir é uma saída de amostra:

   ```
   Error from server (BadRequest): container "ibm-minio-objectstore" in pod "minio-ibm-minio-
   objectstore-848fbcb6f5-2wpq2" is waiting to start: ContainerCreating
   ```

3. Obtenha a descrição do pod.

   ```
   kubectl describe po minio-ibm-minio-objectstore-848fbcb6f5-2wpq2
   ```

   O código a seguir é uma saída de amostra:

   ```
   Name:               minio-ibm-minio-objectstore-848fbcb6f5-2wpq2
   Namespace:          default
   Priority:           0
   PriorityClassName:  <none>
   Node:               10.41.4.202/10.41.4.202
   Start Time:         Fri, 14 Sep 2018 05:52:01 -0700
   ...

   ...
   Events:
     Type     Reason       Age              From                  Message
     ----     ------       ----             ----                  -------
     Normal   Scheduled    5m               default-scheduler     Successfully assigned
   default/minio-ibm-minio-objectstore-848fbcb6f5-2wpq2 to 10.41.4.202
     Warning  FailedMount  1m (x10 over 5m)  kubelet, 10.41.4.202  MountVolume.SetUp failed for
   volume "minio-user" : secrets "minio" not found
     Warning  FailedMount  1m (x2 over 3m)   kubelet, 10.41.4.202  Unable to mount volumes for
   pod "minio-ibm-minio-objectstore-848fbcb6f5-2wpq2_default(f9036aa0-b81c-11e8-bc28-
   00000a2901b6)": timeout expired waiting for volumes to attach or mount for pod
   "default"/"minio-ibm-minio-objectstore-848fbcb6f5-2wpq2". list of unmounted volumes=[minio-
   user]. list of unattached volumes=[export minio-server-config minio-user minio-config-dir
   default-token-tzxvl]
   ```

   A descrição do pod indica que o pod não é capaz de montar o volume, já que o segredo do Minio está indisponível.

4. Verifique se o segredo está disponível no namespace no qual o Minio está implementado.

   ```
   kubectl get secret minio
   ```

O código a seguir é uma saída de amostra:

```
Nenhum recurso localizado.
Error from server (NotFound): secrets "minio" not found
```

A saída indica que o segredo não está disponível no namespace. Crie o segredo seguindo as instruções que estão no Arquivo Leia-me.

Resolva o problema

Para resolver o problema, conclua as etapas a seguir:

1. Exclua a liberação do Helm.
2. Inclua o segredo na configuração do gráfico Helm.
3. Implemente o gráfico de Helm.

O pod do servidor Minio trava no STATUS Pendente

Quando o Minio é implementado no modo distribuído com a alocação de armazenamento dinâmico, o pod do servidor pode travar com o status `Pendente`.

Reúna informações sobre o problema

1. Obtenha a lista de pods.

   ```
   kubectl get po
   ```

   O código a seguir é uma saída de amostra:

   ```
   NAME                                READY     STATUS    RESTARTS   AGE
   mc2                                 1/1       Running   54         2d
   minio-ibm-minio-objectstore-0       0/1       Pending   0          7s
   ```

2. Obtenha a descrição do pod.

   ```
   kubectl describe po minio-ibm-minio-objectstore-0
   ```

   O código a seguir é uma saída de amostra:

   ```
   Name:               minio-ibm-minio-objectstore-0
   Namespace:          default
   Priority:           0
   PriorityClassName:  <none>
   Node:               <none>
   Labels:             app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-1.6.0
                      controller-revision-hash=minio-ibm-minio-objectstore-7b77fd5658
                      heritage=Tiller
                      release=minio
                      statefulset.kubernetes.io/pod-name=minio-ibm-minio-objectstore-0

   ...

   Volumes:
    export:
      Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same
   namespace)
      ClaimName:  export-minio-ibm-minio-objectstore-0
      ReadOnly:   false
   ...

   Events:
    Type     Reason            Age                 From               Message
    ----     ------            ----                ----               -------
    Warning  FailedScheduling  14s (x25 over 57s)  default-scheduler  pod has unbound
   PersistentVolumeClaims (repeated 5 times)
   ```

   A saída indica que os PVCs estão desvinculados.

3. Descreva o PVC.

```
kubectl describe pvc export-minio-ibm-minio-objectstore-0
```

O código a seguir é uma saída de amostra:

```
Name:          export-minio-ibm-minio-objectstore-0
Namespace:     default
StorageClass:  standard
Status:        Pending
Volume:
Labels:        app=ibm-minio-objectstore
               release=minio
Annotations:   <none>
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
Events:
 Type     Reason              Age               From                         Message
 ----     ------              ----              ----                         -------
 Warning  ProvisioningFailed  8s (x19 over 4m)  persistentvolume-controller
storageclass.storage.k8s.io "standard" not found
```

A saída indica que o volume de persistência está tentando ligar por meio da classe de armazenamento denominada `standard`. Verifique se a classe de armazenamento existe em seu cluster.

```
kubectl get sc standard
```

O código a seguir é uma saída de amostra:

```
Nenhum recurso localizado.
Error from server (NotFound): storageclasses.storage.k8s.io "standard" not found
```

A saída indica que a classe de armazenamento não existe.

Resolva o problema

Para resolver o problema, conclua as etapas a seguir:

1. Instale um armazenamento de bloco adequado, como o GlusterFS ou o Ceph, em seu cluster.
2. Assegure-se de que o armazenamento de bloco tenha uma classe de armazenamento.
3. Inclua a classe de armazenamento na configuração do gráfico Helm.
4. Implemente o gráfico de Helm.

O Minio no modo distribuído não é acessível ao fornecer um certificado TLS

Quando o servidor Minio é conectado por meio do Minio Client ou de qualquer cliente compatível com o S3, o erro "Servidor não inicializado" é exibido.

Reúna informações sobre o problema

1. Acesse o contêiner.

   ```
   kubectl exec -it mc2 sh
   ```

   O código a seguir é uma saída de amostra:

   ```
   / # mc config host add myminio https://minio-ibm-minio-objectstore:9000 admin ad
   min1234 S3v4 --insecure
   mc: <ERROR> Unable to initialize new config from the provided credentials. Server not
   initialized,
   please try again.
   / #
   ```

2. Verifique o status do pod.

   ```
   kubectl get po
   ```

   O código a seguir é uma saída de amostra:

```
NAME                                  READY     STATUS    RESTARTS   AGE
mc2                                   1/1       Running   52         2d
minio-ibm-minio-objectstore-0         1/1       Running   0          4m
minio-ibm-minio-objectstore-1         1/1       Running   0          4m
minio-ibm-minio-objectstore-2         1/1       Running   0          4m
minio-ibm-minio-objectstore-3         1/1       Running   0          4m
```

A saída indica que todos os pods estão em execução.

3. Verifique os logs do pod.

```
kubectl logs minio-ibm-minio-objectstore-0
```

O código a seguir é uma saída de amostra:

```
You are running an older version of Minio released 3 weeks ago
Update: https://docs.minio.io/docs/deploy-minio-on-kubernetes

Waiting for a minimum of 2 disks to come online (elapsed 0s)

Waiting for a minimum of 2 disks to come online (elapsed 1s)

Waiting for a minimum of 2 disks to come online (elapsed 2s)

Waiting for a minimum of 2 disks to come online (elapsed 7s)
...
```

A saída indica que as réplicas do servidor não são capazes de se comunicar entre si. O problema pode estar relacionado ao certificado TLS.

Resolva o problema

Assegure-se de gerar o certificado TLS para servidores Minio para o nome comum (CN) no formato a seguir:

```
"/CN=*.<chart deployment name>-ibm-minio-objectstore.<namespace>.svc.<cluster domain name>"
```

Esta etapa é um requisito para servidores Minio que são configurados com o certificado TLS.

O exemplo a seguir possui as etapas para gerar um certificado para implementação do Minio:

- Nome da liberação do Helm: minio
- Namespace para implementação: default
- Nome do domínio do cluster: cluster.local

```
openssl genrsa -out private.key 2048
openssl req -new -x509 -days 3650 -key private.key -out public.crt -subj  "/CN=*.minio-ibm-minio-objectstore.default.svc.cluster.local"
cp public.crt ca.crt
kubectl create secret generic tls-ssl-minio --from-file=./private.key --from-file=./public.crt --from-file=./ca.crt
```

Nota: o certificado é gerado para uma combinação especificada de nome da liberação, namespace e nome do domínio do cluster do Helm. O certificado não funcionará se qualquer um desses valores for diferente. Deve-se criar um certificado diferente para qualquer outra combinação de valores.

Os depósitos e objetos Minio estão intermitentemente inacessíveis

Quando o Minio é implementado no modo NAS e você se conecta a partir de um cliente, os depósitos e objetos criados ficam intermitentemente inacessíveis.

Sintomas

1. Crie um armazenamento de objeto Minio.

```
mc config host add myminio http://minio-nas-ibm-minio-objectstore:9000 admin admin1234 S3v4
```

O seguinte é uma saída de amostra:

```
`myminio` incluído com sucesso.
```

2. Crie um depósito.

   ```
   mc mb myminio/test
   ```

   O seguinte é uma saída de amostra:

   ```
   Depósito `myminio/test` criado com sucesso.
   ```

Ao executar o comando MinIO Client (mc) para listar os depósitos e objetos, talvez você veja o erro a seguir:

```
mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
/ # mc ls myminio/test
mc: <ERROR> Unable to stat `myminio/test`. Bucket `test` does not exist.
```

Para identificar a causa, descreva um pod Minio e verifique se ele está usando o volume persistente desejado.

1. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

2. Obtenha o nome do pod Minio.

   ```
   kubectl get pods | grep minio
   ```

   O seguinte é uma saída de amostra:

   ```
   minio-nas-ibm-minio-objectstore-974f85dc9-7mlrv   1/1       Running   0          4m
   ```

3. Obtenha as informações do pod Minio.

   ```
   kubectl describe po minio-nas-ibm-minio-objectstore-974f85dc9-7mlrv
   ```

   {codeblock}

   O seguinte é uma saída de amostra:

   ```
   Name:               minio-nas-ibm-minio-objectstore-974f85dc9-7mlrv
   Namespace:          default
   Priority:           0
   PriorityClassName:  <none>
   Node:               10.41.14.22/10.41.14.22
   Start Time:         Thu, 25 Apr 2019 22:13:12 -0700
   Labels:             app=ibm-minio-objectstore
                      chart=ibm-minio-objectstore-2.4.7
                      heritage=Tiller
                      pod-template-hash=974f85dc9
                      release=minio-nas
   Annotations:        kubernetes.io/psp: 00-rook-ceph-operator
                      productID: Minio_RELEASE.2019-04-09T01-22-30Z_free_00000
                      productName: Minio
                      productVersion: RELEASE.2019-04-09T01-22-30Z
                      scheduler.alpha.kubernetes.io/critical-pod:
   Status:             Running
   IP:                 10.1.106.199
   Controlled By:      ReplicaSet/minio-nas-ibm-minio-objectstore-974f85dc9
   Containers:
    ibm-minio-objectstore:
      Container ID:  docker://33a355387f1df4db9e932cf2921f095cac61baf174522eabf779db2d6c16a779
      Image:         minio/minio:RELEASE.2019-04-09T01-22-30Z
      Image ID:      docker-
   pullable://minio/minio@sha256:b363f54fc5a64d259d760106ad02c8725999c935f7aeae5348abfc0bed3fef0d
      Port:          9000/TCP
      Host Port:     0/TCP
      Command:
        /bin/sh
   ```

```
      -ce
      /usr/bin/docker-entrypoint.sh minio -C /root/.minio/ gateway nas /export
    State:          Running
      Started:      Thu, 25 Apr 2019 22:13:34 -0700
    Ready:          True
    Restart Count:  0
    Requests:
      cpu:      250m
      memory:   256Mi
    Liveness:   http-get http://:service/minio/health/live delay=5s timeout=1s period=30s
#success=1 #failure=3
    Readiness:  http-get http://:service/minio/health/ready delay=5s timeout=1s period=15s
#success=1 #failure=3
    Environment:
      MINIO_ACCESS_KEY:  <set to the key 'accesskey' in secret 'minio'>  Optional: false
      MINIO_SECRET_KEY:  <set to the key 'secretkey' in secret 'minio'>  Optional: false
      MINIO_BROWSER:     on
    Mounts:
      /root/.minio/ from minio-config-dir (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-vccnm (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  export:
    Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
  minio-user:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  minio
    Optional:    false
  minio-config-dir:
    Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
  default-token-vccnm:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-vccnm
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     CriticalAddonsOnly
                 node.kubernetes.io/memory-pressure:NoSchedule
                 node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age   From                 Message
  ----    ------     ----  ----                 -------
  Normal  Scheduled  13m   default-scheduler    Successfully assigned default/minio-nas-ibm-
minio-objectstore-974f85dc9-7mlrv to 10.41.14.22
  Normal  Pulling    13m   kubelet, 10.41.14.22  pulling image "minio/minio:RELEASE.2019-04-
09T01-22-30Z"
  Normal  Pulled     12m   kubelet, 10.41.14.22  Successfully pulled image
"minio/minio:RELEASE.2019-04-09T01-22-30Z"
  Normal  Created    12m   kubelet, 10.41.14.22  Created container
```

A saída de comando mostra que os pods Minio estão usando o seguinte volume:

```
Volumes:
  export:
    Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
```

Os depósitos e objetos ficam intermitentemente inacessíveis pelos seguintes motivos:

- O volume Persistente não corresponde ao armazenamento ReadWriteMany.
- O Persistente não está rotulado corretamente.

Resolvendo o problema

1. Desinstale a implementação com falha.

2. Use um volume persistente correspondente a uma tecnologia de armazenamento que suporte o volume ReadWriteMany.
3. Ligue a solicitação de volume persistente (PVC) a um volume persistente (PV) específico. Para assegurar que ocorra a ligação, rotule o PV que você precisa ligar como `\"pv: <pv name>\"`. Consulte o seguinte exemplo:

```
apiVersion: v1
kind: PersistentVolume
metadata:
 labels:
   pv: shared-pv
 name: shared-pv
```

Resolução de problemas do cluster do Rook Ceph

Revise frequentemente os problemas de cluster do Rook Ceph encontrados.

- A instalação do gráfico do Rook Operator usando a console de gerenciamento obtém o erro ESOCKETTIMEDOUT
- A exclusão do gráfico do Rook Operator não exclui o daemonset do Rook
- A instalação do gráfico de cluster do Rook (ibm-rook-rbd-cluster) usando a console de gerenciamento obtém o erro ESOCKETTIMEDOUT
- A instalação do gráfico de cluster do Rook (ibm-rook-rbd-cluster) obtém a falha da tarefa: Erro de BackoffLimitExceeded
  - O gráfico Helm do Rook Operator não está instalado em seu cluster
  - O gráfico de cluster do Rook já está instalado em seu namespace
- O gráfico de cluster do Rook inicia a implementação, mas o rook-ceph-mon obtém um erro CrashLoopBackOff
- A implementação é concluída para o gráfico ibm-rook-rbd-cluster, mas nenhum pod rook-mon, rook-ceph, manager ou api aparece
- Depois que um nó do trabalhador é reiniciado, o pod do agente do Rook permanece no status de erro

Antes de continuar com a resolução de problemas, assegure-se de que seu cluster atenda aos pré-requisitos e que você tenha permissões adequadas para executar operações relacionadas à instalação. Para obter mais informações, consulte Pré-requisitos e limitações.

Nota: é necessário configurar a CLI do kubectl para executar comandos de resolução de problemas. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

A instalação do gráfico do Rook Operator usando a console de gerenciamento obtém

o erro ESOCKETTIMEDOUT

É possível ver um erro `ESOCKETTIMEDOUT` enquanto instala o gráfico do Rook Operator usando a console de gerenciamento:

Resolva o problema

Verifique se os pods rook-agent e do operador estão em execução. Execute os comandos a seguir:

```
kubectl get nodes -o wide
```

```
kubectl -n default get po -o wide
```

Se você tiver tantos agentes quanto o número de nós do trabalhador e se um pod do operador estiver em execução, a instalação foi bem-sucedida. É possível ignorar o erro.

Se os pods do agente ou do operador não estiverem em execução, verifique os logs do Helm para identificar o erro:

```
kubectl -n kube-system get po | grep helm
```

O seguinte é uma saída de amostra:

```
helm-api-66b98d88bc-6psq6                          2/2       Running    0          1d
helm-repo-5495f5c48c-k9mkl                         1/1       Running    0          1d

kubectl -n kube-system log helm-api-66b98d88bc-6psq6 rudder
```

A exclusão do gráfico do Rook Operator não exclui o daemonset do Rook

1. Obtenha a lista de pods.

```
kubectl -n default get po
```

O seguinte é uma saída de amostra:

```
NAME               READY     STATUS    RESTARTS   AGE
rook-agent-ckxht   1/1       Running   0          23h
rook-agent-jnxh6   1/1       Running   0          23h
rook-agent-wkt26   1/1       Running   0          23h
```

2. Obtenha o daemonset.

```
kubectl -n default get ds
```

O seguinte é uma saída de amostra:

```
NAME         DESIRED   CURRENT   READY     UP-TO-DATE   AVAILABLE   NODE-SELECTOR   AGE
rook-agent   3         3         3         3            3           <none>          23h
```

Esse problema é conhecido na liberação alfa do Rook. Exclua manualmente o daemonset do agente. Execute os comandos a seguir:

1. Exclua o daemonset.

```
kubectl -n default delete ds rook-agent
```

O seguinte é uma saída de amostra:

```
daemonset "rook-agent" deleted
```

2. Obtenha uma lista de pods.

```
kubectl -n default get po
```

O seguinte é uma saída de amostra:

```
Nenhum recurso localizado.
```

A instalação do gráfico de cluster Rook (ibm-rook-rbd-cluster) usando a

console de gerenciamento obtém o erro ESOCKETTIMEDOUT

É possível ver um erro `ESOCKETTIMEDOUT` enquanto instala o gráfico de cluster Rook usando a console de gerenciamento.

1. Verifique os pods que estão em execução no namespace no qual você está instalando o gráfico. Procure por `rook-cluster-precheck-job` e seu InitContainer. Talvez você veja o erro a seguir:

```
kubectl -n default get po
```

O seguinte é uma saída de amostra:

```
NAME                               READY     STATUS              RESTARTS   AGE
rook-cluster-precheck-job-mqfv9    0/1       Init:ErrImagePull   0          28s
```

2. Verifique se o repositório do Docker que está especificado para a imagem do Hyperkube está correto.

A instalação do gráfico de cluster Rook (ibm-rook-rbd-cluster) obtém a falha da tarefa:

Erro de BackoffLimitExceeded

É possível ver um erro `BackoffLimitExceeded` enquanto instala o gráfico de cluster Rook.

As duas razões a seguir podem estar causando este erro:

- O gráfico Helm do Rook Operator não está instalado em seu cluster.
- O gráfico de cluster Rook já está instalado em seu namespace.

O gráfico Helm do Rook Operator não está instalado em seu cluster

Verifique se o gráfico Helm do Rook Operator está instalado em seu cluster:

```
kubectl get po --all-namespaces | grep rook-operator
```

Se o gráfico não estiver instalado, instale-o primeiramente.

Para obter mais informações sobre como instalar o gráfico Helm do Rook Operator, consulte Gráfico Helm do Ceph Operator.

O gráfico de cluster do Rook já está instalado em seu namespace

Verifique se o gráfico de cluster Rook está instalado em seu cluster:

```
kubectl -n default get cluster
```

O seguinte é uma saída de amostra:

```
NAME              KIND
default-cluster   Cluster.v1alpha1.rook.io
```

Não é possível instalar múltiplos gráficos de cluster Rook em um namespace.

O gráfico de cluster Rook inicia a implementação, mas o rook-ceph-mon obtém um erro

CrashLoopBackOff

1. Obtenha uma lista de pods.

   ```
   kubectl -n default get po
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME                              READY     STATUS             RESTARTS   AGE
   rook-agent-8tlqf                  1/1       Running            0          24m
   rook-agent-htjdl                  1/1       Running            0          24m
   rook-agent-q46vw                  1/1       Running            0          24m
   rook-ceph-mon0-f2bc6              0/1       CrashLoopBackOff   2          37s
   rook-operator-947bf78c6-8hjgj     1/1       Running            0          24m
   ```

2. Verifique o log `rook-ceph-mon`.

   ```
   kubectl -n default log rook-ceph-mon0-f2bc6
   ```

   O seguinte é uma saída de amostra:

   ```
   2018-05-18 10:51:39.932606 I | rook: starting Rook v0.7.1 with arguments '/usr/local/bin/rook
   mon --config-dir=/var/lib/rook --name=rook-ceph-mon0 --port=6790 --fsid=a01d92fb-8191-4343-
   8ec1-676abd0de780'
   2018-05-18 10:51:39.932749 I | rook: flag values: --admin-secret=*****, --ceph-config-
   override=/etc/rook/config/override.conf, --cluster-name=default, --config-dir=/var/lib/rook, --
   fsid=a01d92fb-8191-4343-8ec1-676abd0de780, --help=false, --log-level=INFO, --mon-
   endpoints=rook-ceph-mon0=10.0.0.185:6790, --mon-secret=*****, --name=rook-ceph-mon0, --
   port=6790, --private-ipv4=10.1.19.21, --public-ipv4=10.0.0.185
   The keyring does not match the existing keyring in /var/lib/rook/rook-ceph-mon0/data/keyring.
   Pode ser
   necessário excluir o conteúdo de dataDirHostPath no host a partir de uma implementação
   anterior.
   ```

Esse erro indica que você tinha uma implementação do Rook Ceph anterior.

Para corrigir esse problema, exclua o gráfico com falha e, em seguida, exclua o conteúdo do arquivo `dataDirHostPath` nos hosts que foram usados em uma implementação anterior. Ou especifique uma configuração `dataDirHostPath` diferente. Em seguida, reinstale o gráfico `ibm-rook-rbd-cluster`.

A implementação é concluída para o gráfico ibm-rook-rbd-cluster, mas nenhum pod de rook-ceph-mon,

rook-ceph, manager ou api aparece

Esse problema pode acontecer ao tentar reinstalar o `ibm-rook-rbd-cluster` sem excluir o conteúdo do `dataDirHostPath` nos hosts para limpeza dos discos de armazenamento.

Para resolver o problema, conclua as tarefas a seguir:

1. Excluir o gráfico `ibm-rook-rbd-cluster` com falha
2. Excluir o gráfico do Rook Operator

3. Excluir o conteúdo de `dataDirHostPath`.
4. Limpar o disco que você usou para armazenamento.
5. Reinstalar o gráfico do Rook Operator.
6. Reinstalar o gráfico `ibm-rook-rbd-cluster`.

Depois que um nó do trabalhador é reiniciado, o pod do agente do Rook permanece no status de erro

1. Obtenha as informações do pod.

   ```
   kubectl get po -o wide
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME                              READY     STATUS                RESTARTS   AGE       IP
   NODE
   rook-agent-5rst5                  1/1       Running               0          2d
   9.5.28.147    9.5.28.147
   rook-agent-bsrrx                  1/1       Running               0          2d
   9.5.28.143    9.5.28.143
   rook-agent-zq4bm                  0/1       CreateContainerError  1          2d
   9.5.28.146    9.5.28.146
   rook-api-86b5b8849c-fjqf8         1/1       Running               0          7m
   10.1.68.153   9.5.28.147
   rook-ceph-mgr0-9c56544c8-2mxqr    1/1       Running               0          2d
   10.1.19.31    9.5.28.143
   rook-ceph-mon0-g5t7m              1/1       Running               0          2d
   10.1.19.30    9.5.28.143
   rook-ceph-mon1-zl5px              1/1       Running               5          7m
   10.1.0.164    9.5.28.146
   rook-ceph-mon2-jjjht              1/1       Running               0          2d
   10.1.68.151   9.5.28.147
   rook-ceph-osd-9.5.28.143-2bpl6    1/1       Running               0          2d
   10.1.19.32    9.5.28.143
   rook-ceph-osd-9.5.28.146-8qwbx    1/1       Running               5          7m
   10.1.0.165    9.5.28.146
   rook-ceph-osd-9.5.28.147-mcksg    1/1       Running               0          2d
   10.1.68.152   9.5.28.147
   rook-operator-947bf78c6-58nng     1/1       Running               0          2d
   10.1.19.22    9.5.28.143
   ```

2. Obtenha informações sobre o pod do agente Rook.

   ```
   kubectl describe po rook-agent-zq4bm
   ```

   O seguinte é uma saída de amostra:

   ```
   Name:         rook-agent-zq4bm
   Namespace:    default
   Node:         9.5.28.146/9.5.28.146
   ...

   6m        6m        3    kubelet, 9.5.28.146    spec.containers{rook-agent}    Warning
   Failed    Error: Error response from daemon:
   Conflict. The container name "/k8s_rook-agent_rook-agent-zq4bm_default_5b2c4423-5a8e-11e8-a2b0-
   005056a7db67_2" is already in
   use by container ac71dc3e805f470d44afe6660f668e71832753505532625a9f30905c30f2063a. You have to
   remove (or rename) that
   container to be able to reuse that name.
   ```

3. Efetue login no nó no qual o pod está falhando. Encerre o contêiner em conflito conforme relatado no erro.

   ```
   docker kill ac71dc3e805f470d44afe6660f668e71832753505532625a9f30905c30f2063a
   ac71dc3e805f470d44afe6660f668e71832753505532625a9f30905c30f2063a
   ```

   O pod do agente inicia a execução normalmente.

   ```
   kubectl get po -o wide
   ```

   O seguinte é uma saída de amostra:

   ```
   NAME                              READY     STATUS    RESTARTS   AGE       IP             NODE
   rook-agent-5rst5                  1/1       Running   0          2d        9.5.28.147
   9.5.28.147
   ```

```
rook-agent-bsrrx                1/1       Running   0          2d        9.5.28.143
9.5.28.143
rook-agent-zq4bm                1/1       Running   2          2d        9.5.28.146
9.5.28.146
rook-api-86b5b8849c-fjqf8       1/1       Running   0          12m       10.1.68.153
9.5.28.147
rook-ceph-mgr0-9c56544c8-2mxqr  1/1       Running   0          2d        10.1.19.31
9.5.28.143
rook-ceph-mon0-g5t7m            1/1       Running   0          2d        10.1.19.30
9.5.28.143
rook-ceph-mon1-zl5px            1/1       Running   5          12m       10.1.0.164
9.5.28.146
rook-ceph-mon2-jjjht            1/1       Running   0          2d        10.1.68.151
9.5.28.147
rook-ceph-osd-9.5.28.143-2bpl6  1/1       Running   0          2d        10.1.19.32
9.5.28.143
rook-ceph-osd-9.5.28.146-8qwbx  1/1       Running   5          12m       10.1.0.165
9.5.28.146
rook-ceph-osd-9.5.28.147-mcksg  1/1       Running   0          2d        10.1.68.152
9.5.28.147
rook-operator-947bf78c6-58nng   1/1       Running   0          2d        10.1.19.22
9.5.28.143
```

Resolução de problemas do vSphere Cloud Provider

Revise problemas do vSphere Cloud Provider encontrados com frequência.

- Coletar logs
- O nó principal entra no estado "notReady" depois de configurar o provedor em nuvem do vSphere
- Falha no fornecimento de volume persistente

Coletar logs

Colete logs para identificar a causa raiz de um problema.

Para resolução de problemas do vSphere Cloud Provider, deve-se reunir os arquivos de log a seguir:

Antes de iniciar, assegure-se de que a CLI kubectl esteja configurada. Para obter informações adicionais, consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).

Log do gerenciador do controlador

1. Obtenha o nome do pod principal do Kubernetes.

   ```
   kubectl -n kube-system get pod | grep k8s-master
   ```

   O seguinte é uma saída de amostra:

   ```
   k8s-master-<nodename> 4/4 Running 3 1d
   ```

2. Obtenha o log do gerenciador do controlador.

   ```
   kubectl -n kube-system logs k8s-master-<nodename> -c controller-manager > controller-manager-
   <nodename>.log
   ```

Log do servidor da API

1. Obtenha o nome do pod principal do Kubernetes.

   ```
   kubectl -n kube-system get pod | grep k8s-master
   ```

   O seguinte é uma saída de amostra:

   ```
   k8s-master-<nodename> 4/4 Running 3 1d
   ```

2. Obtenha o log do servidor de API.

   ```
   kubectl -n kube-system logs k8s-master-<nodename> -c apiserver > apiserver-<nodename>.log
   ```

Log do Kubelet

Em cada nó, execute o comando a seguir para obter o log kubelet a partir desse nó.

```
journalctl -u kubelet > kubelet-< nodename> .log
```

O nó principal entra no estado "notReady" após a configuração do vSphere Cloud Provider

Depois de configurar um provedor em nuvem vSphere, seu cluster do IBM® Cloud Private fica inativo e os nós principais estão no estado "notReady".

Causas

Para configurar um provedor em nuvem, os parâmetros `-- cloud-provider` e `--cloud-config` precisam ser configurados para transmitir o tipo de provedor em nuvem e suas informações de configuração para o kubelet.

Se você fornecer informações de configuração incorretas, o processo `kubelet` não será iniciado.

Resolvendo o problema

- Verifique se você forneceu o nome do usuário e a senha corretos do vCenter.
- Verifique os logs kubelet e o journalctl para localizar a causa raiz do problema. Para obter mais informações, consulte Log do Kubelet .

Depois de identificar o erro, conclua as etapas a seguir:

1. Faça as correções necessárias no arquivo de configuração. Para obter detalhes de configuração, consulte Configurando um vSphere Cloud Provider após a instalação do IBM Cloud Private.
2. Recarregue o arquivo de unidade systemd do kubelet.

   ```
   systemctl daemon-reload
   ```

3. Reinicie o serviço kubelet para trazer o nó de volta no estado ativo.

   ```
   Systemctl restart kubelet.service
   ```

   Nota: ao reiniciar o serviço kubelet, é possível que algum rótulo ou contaminação que você incluiu manualmente no nó seja perdido. Se sim, é possível incluí-lo novamente agora.

Falha no fornecimento de volume persistente

Falha no fornecimento de volume persistente (PV) com um erro "No VM found".

Causas

Esse problema ocorre quando o identificador do modo principal não é configurado ou é configurado incorretamente.

Talvez você veja as seguintes mensagens de log no arquivo de log do gerenciador do controlador:

```
[nodemanager.go:419] Error "No VM found" node info for node "master-node-01" not found
[vsphere_util.go:134] Error while obtaining Kubernetes node nodeVmDetail details. error : No VM
found
[vsphere.go:1160] Failed to get shared datastore: No VM found
```

Resolvendo o problema

1. Verifique o ID do provedor e o número de série do produto dos nós principais.
   a. Obtenha o número de série do produto do nó principal.

   ```
    sudo cat /sys/class/dmi/id/product_serial | sed -e 's/^VMware-//' -e 's/-/ /' | awk '{ print
   toupper($1$2$3$4 "-" $5$6 "-" $7$8 "-" $9$10 "-" $11$12$13$14$15$16) }'
   ```

O seguinte é uma saída de amostra:

```
4228B6D2-87BA-4094-2578-129A9085585C
```

b. Obtenha o ID do provedor das informações do nó do Kubernetes.

```
kubectl get node <master-node-01> -o json | jq '[.metadata.name, .spec.providerID]'
```

O seguinte é uma saída de amostra:

```
[
  "master-node-01",
  "vsphere://4228154b-efa3-51e0-80e2-53000dcdf383"
]
```

2. Instale o kubectl. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).

3. Configure o ID do provedor com o valor correto do número de série do produto.

```
kubectl patch node <master-node-01> -p '{"spec":{"providerID":"vsphere://4228B6D2-87BA-4094-
2578-129A9085585C"}}'
```

Nota: se não for possível modificar o objeto do nó, deve-se excluir as informações do nó do inventário do Kubernetes e reiniciar o kubelet para que o nó seja registrado novamente.

Uma interação lenta entre o kubelet e o Docker causa problemas do PLEG

O Docker atrasa sua resposta para o kubelet ao criar um pod.

Sintomas

Os pods podem falhar durante a criação, porque há um tempo limite para o kubelet para chamar o mecanismo Docker.

Causas

Há uma sobrecarga de pods no nó do host. Você poderá receber a seguinte mensagem de erro:

```
skipping pod synchronization - [PLEG is not healthy: pleg was last seen active 6m55.488150776s ago;
threshold is 3m0s]
```

Reduza a possibilidade de ocorrência

Atualize o arquivo de serviço kubelet.

1. Edite o arquivo de configuração de serviço kubelet. Execute o comando a seguir:

```
vi /etc/systemd/system/kubelet.service
```

2. Inclua a opção `--housekeeping-interval` para aumentar o tempo de intervalo. O valor padrão é 10s. Inclua a seguinte opção:

```
-- housekeeping-interval=30s
```

Seu arquivo de configuração de serviço kubelet pode ser semelhante à saída a seguir:

```
[Unit] Description=Kubelet Service Documentation=https://github.com/kubernetes/kubernetes

[Service]
EnvironmentFile=-/etc/environment
ExecStart=/opt/kubernetes/hyperkube kubelet \
 --cadvisor-port=0 \
 --docker-disable-shared-pid \
 --bootstrap-kubeconfig=/etc/cfc/kubelet/kubelet-bootstrap-config \
 --kubeconfig=/etc/cfc/kubelet/kubelet-config \
 --cert-dir=/etc/cfc/kubelet \
 --config=/etc/cfc/kubelet/kubelet-service-config \
 --dynamic-config-dir=/etc/cfc/kubelet/kubelet-dynamic-config \
 --network-plugin=cni \
 --hostname-override=9.111.255.33 \
```

```
 --node-ip=9.111.255.33 \
 --pod-infra-container-image=hyc-cloud-private-edge-docker-local.artifactory.swg-
devops.com/ibmcom-amd64/pause:3.1 \
 --node-labels=node-role.kubernetes.io/worker=true, \
 --register-with-taints= \
 --keep-terminated-pod-volumes=false \
 --housekeeping-interval=30s

Restart=always RestartSec=10

[Install] WantedBy=multi-user.target
```

3. Reinicie o serviço kubelet no nó do host com os comandos a seguir:

```
systemctl daemon-reload
systemctl restart kubelet
```

4. Deve-se especificar uma solicitação de recurso e limitar para seu aplicativo. Para obter mais detalhes, consulte Configurando a cota de recurso.

Eventos, Logs e Códigos de Erro

Revisar problemas encontrados frequentemente envolvendo eventos e logs.

- Eventos e logs (CLI)
- Eventos e logs (console de gerenciamento do cluster)
- Dados do log Elasticsearch não são limpos
- Códigos de erro
- Resolução de problemas de logs de auditoria

Eventos e logs (CLI)

Para ajudar na resolução de problemas de seu cluster, revise os logs do componente.

Logs do Docker

- Para verificar o status de contêineres, em quaisquer nós principais ou do trabalhador, execute:

  ```
  docker ps -a
  ```

  Isso retorna informações sobre quaisquer contêineres em execução em um nó.

- Para verificar o status de saída de qualquer contêiner em execução, execute o comando a seguir:

  ```
  docker logs container-id
  ```

Logs do trabalhador

- Para visualizar os logs de trabalhadores do Kubernetes, em um nó do trabalhador, execute o comando a seguir:

  ```
  systemctl status kubelet
  ```

- Para visualizar mais detalhes sobre os logs de trabalhadores do Kubernetes, em um nó do trabalhador, execute o comando a seguir:

  ```
  journalctl -u kubelet.service
  ```

- Para visualizar os logs Calico, em um nó do trabalhador, execute o comando a seguir:

  ```
  docker logs -f calico
  ```

Eventos e logs (console de gerenciamento do cluster)

Acesse os eventos e os logs de implementação usando a console de gerenciamento do cluster.

É possível usar os eventos e logs a seguir, acessíveis por meio da console de gerenciamento do cluster, como ferramentas ao solucionar problemas com uma implementação ou operação.

Eventos de implementação

Os eventos de implementação contêm detalhes da implementação e outros detalhes do evento para uma implementação.

Para revisar os eventos de implementação para uma implementação específica:

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Selecione a implementação que você deseja visualizar.
3. Clique na guia **Eventos**. É possível usar a função de procura para consultas rápidas baseadas na palavra-chave.

Logs de implementação

Antes de verificar o log de implementação, verifique se todos os pods em uma implementação estão no estado **Em execução**. Caso contrário, poderá ser necessário revisar os logs para descobrir por que os pods não estão funcionando.

Para revisar os logs de implementação para uma implementação específica:

1. Em sua página de cargas de trabalho, navegue para a página de detalhes do recurso.
2. Na tabela `Pods`, clique no menu de ações associado a seu pod e selecione `View logs`. Também é possível navegar para a página de detalhes do pod e clicar no menu Ações para `View logs`.
3. Se o Kibana estiver instalado, você será ativado para o Kibana. Se o Kibana não estiver instalado, você verá um pop-up contendo o comando kubectl correspondente. Use o comando kubectl para visualizar os logs para seu pod.
4. Para procurar no Kibana, navegue para a guia `Descoberta`. Digite suas consultas na caixa de texto. É possível usar os campos a seguir por conta própria ou combiná-los com outros campos usando `AND` e outros operadores. A documentação adicional está disponível nos sites Elastic Company e Apache Lucene.
    - `kubernetes.container_name` é o nome do contêiner que gerou o log.
    - `kubernetes.pod` é o nome do pod ao qual o contêiner pertence.
    - `kubernetes.namespace` é o namespace no qual o contêiner foi implementado.
    - `kubernetes.container_id` é o UUID designado ao contêiner pelo Docker.

Exemplos de consultas:

- `kubernetes.pod:mypod`
- `kubernetes.pod:mypod AND severity: NORMAL`
- `kubernetes.namespace:default AND FileNotFoundError`

Eventos ReplicaSet

ReplicaSets são usados para manter as réplicas ou pods que estão em execução em um cluster.

Para revisar os eventos ReplicaSet para uma implementação específica:

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Selecione a implementação que você deseja visualizar.
3. Revise os detalhes do ReplicaSet.

Eventos do pod

Os pods são as unidades implementáveis base que são criadas em seu cluster. Um único pod pode conter um ou mais contêineres.

Para revisar os eventos do pod para uma implementação específica:

1. No menu de navegação, clique em **Cargas de trabalho** > **Implementações**.
2. Selecione a implementação que você deseja visualizar.
3. Selecione o pod que você deseja visualizar.
4. Clique na guia **Eventos**. Uma caixa de procura que permite consultas rápidas baseadas em palavra-chave também está disponível.

Dados do log Elasticsearch não são limpos

Os dados do índice do Elasticsearch para os logs de cluster (Logstash) não são removidos do nó de gerenciamento.

Sintomas

Os dados de log de cluster que são armazenados na pasta `/opt/ibm/cfc/logging/elasticsearch` ocupam espaço em disco excessivo.

Causas

Por padrão, os logs são armazenados por 1 dia. Uma tarefa cron é executada a cada dia às 23h59min para limpar os logs. Se seus computadores estiverem suspensos nesse horário, os logs poderão se acumular.

Se seus nós de gerenciamento usarem o Red Hat Enterprise Linux (RHEL), um driver de armazenamento do Docker configurado incorretamente pode impedir que a tarefa cron seja executada automaticamente. Essa configuração pode fazer com que os logs se acumulem.

Além disso, se seus contêineres gerarem uma grande quantidade de dados de log ou de métrica, a capacidade de armazenamento de seus nós de gerenciamento poderá ser muito pequena ou os intervalos padrão de armazenamento de log e métrica poderão ser muito longos.

Resolvendo o problema

Se seus nós de gerenciamento usarem o RHEL, confirme se o driver de armazenamento do Docker está configurado corretamente. Os drivers de armazenamento devem ser configurados antes de instalar o IBM® Cloud Private.

- Para instalação manual do Docker, consulte Configurando seu mecanismo de Docker.
- Para instalação automática do Docker, consulte Configurações do Docker.

Se os contêineres gerarem uma grande quantidade de dados de log ou de métrica, aumente a capacidade de armazenamento de seus nós de gerenciamento ou modifique os critérios de curador de log e de métrica padrão seguindo as instruções na seção **Retenção de dados** da página Criação de log do IBM Cloud Private.

Códigos de Erros

Revise uma lista de códigos de erro que talvez você encontre ao trabalhar com a console de gerenciamento do IBM® Cloud Private.

Código de Erro	Significado	Notas
400	**Solicitação inválida** - a solicitação ou resposta é inválida.	
403	**Proibido** - o acesso solicitado não é permitido para este usuário.	
500	**Erro do servidor interno** - ocorreu um problema no servidor.	
502	**Serviço indisponível** - a resposta solicitada atingiu o tempo limite.	

{: caption="Tabela 1. IBM Cloud Private { } códigos de erro" caption-side="top"}

Resolução de problemas de logs de auditoria

Resolução de problemas de logs de auditoria

Não é possível ver logs de auditoria no Kibana

O problema pode ser por qualquer um dos seguintes motivos:

- A criação de log de auditoria está desativada por padrão Se for necessário gerar logs de auditoria para um serviço, você deve ativá-lo para esse serviço. Para obter informações adicionais, consulte Criação de log de auditoria no IBM Cloud Private.
- O parâmetro `AUDIT flag` é configurado como `true` no ConfigMap do serviço, mas ainda não é possível ver logs de auditoria no Kibana.

  Depois de configurar `AUDIT flag: true` no ConfigMap do serviço, verifique se os pods de serviço relacionados foram reiniciados. Para obter informações adicionais, consulte Criação de log de auditoria no IBM Cloud Private.

- O parâmetro `AUDIT flag` é configurado como `true` no ConfigMap do serviço, e os pods relacionados são reiniciados, mas ainda não é possível ver logs de auditoria no Kibana.
    - Verifique se a função de controle de acesso baseado na função (RBAC) tem acesso aos logs. Somente as funções de `auditor` e `administrador de cluster` têm privilégios para ver os logs de auditoria.
    - Verifique se o índice de `auditoria` foi criado no Kibana. Se ele não foi criado, é possível criá-lo seguindo estas etapas:

        1. Abra o painel do Kibana.
        2. Navegue para **Gerenciamento** > **Padrões de índice**.
        3. Clique em **Criar padrão de índice**.
        4. Inclua **padrão de índice** como `audit-*`.
        5. Selecione `@timestamp` no menu suspenso **Nome do campo de filtro de tempo**.
        6. Clique em **Criar padrão de índice**.

        É possível ver os logs de auditoria na seção **Descobrir** do painel. O índice `audit-*` deve ser exibido na seção **Campos selecionados**.

    - Se ainda não for possível ver os logs de auditoria, verifique o fluxo do log de auditoria e identifique o problema. O fluxo do log de auditoria do pod de serviço para o painel do Kibana é `Pods geram logs de auditoria` > `Journald` > `Fluentd` > `Elasticsearch` > `Kibanaa`.

        1. Instale o `kubectl`. Para obter mais informações, consulte Instalando a CLI do Kubernetes (kubectl).
        2. Localize o endereço IP do pod de serviço que tem o log de auditoria ativado.

            ```
            kubectl -n kube-system get pods -o wide | grep <service name or pod name of the service>
            ```

        3. Use o Shell Seguro (SSH) para se conectar a esse nó e verificar se os logs de auditoria estão acessando `journald`.

            ```
            journalctl -t 'icp-audit'
            journalctl -t 'icp-audit' -o json-pretty
            ```

        4. Se não localizar nenhum log, verifique se `journald` está funcionando. Em seguida, repita a etapa 4.

            ```
            systemd-cat -t icp-audit tail "Audit log testing message."
            ```

        5. Verifique se os pods `fluentd` e os pods de criação de log estão em execução.

            ```
            kubectl -n kube-system get pods
            ```

        6. Analise os logs do pod `fluentd` para verificar se o `fluentd` está conectado ao ELK.

            ```
            kubectl -n kube-system log <fluentd pod name>
            ```

            Nota: O nome do pod `fluentd` começa com `audit-logging-fluentd-ds-*`.

            Se não houver nenhum erro no log e for possível ver o seguinte texto nas primeiras linhas do log, isso indica que `fluentd` foi conectado com sucesso ao ELK.

            ```
             "Connection opened to Elasticsearch cluster => {:host="elasticsearch", :port=>9200, :scheme=>"https"}> "
            ```

            Se não for possível ver as linhas no log, verifique se o serviço de criação de log está instalado e em execução.

Ativar criação de log de auditoria mas não enviar logs para o ELK

Se desejar ativar a criação de log de auditoria, mas não desejar enviar os logs para o ELK, conclua as seguintes etapas:

1. Atualize o ConfigMap `audit-logging-fluentd-ds`.
    - Use a console de gerenciamento do IBM Cloud Private para editar o ConfigMap.

        1. Efetue logon no console de gerenciamento.
        2. Navegue para **Configuração** > **ConfigMaps**.
        3. Localize **audit-logging-fluentd-ds**.
        4. Clique em **...** > **Editar**.

5. Configure a sinalização `ENABLE_AUDIT_LOGGING_FORWARDING` como "`false`" e salve o arquivo ConfigMap.

- Use `kubectl` para editar o arquivo ConfigMap `audit-logging-fluentd-ds-config`.

  ```
  kubectl -n kube-system edit configmap audit-logging-fluentd-ds-config
  ```

  Configure a sinalização `ENABLE_AUDIT_LOGGING_FORWARDING` como "`false`" e salve o arquivo ConfigMap.

2. Recrie os pods `fluentd`.

   - Use a console de gerenciamento para recriar os pods.

     1. Efetue logon no console de gerenciamento.
     2. Navegue para **Cargas de trabalho** > **DaemonSets**.
     3. Localize o daemonset **audit-logging-fluentd-ds**. Clique no daemonset para ver todos os pods.
     4. Remova todos os pods `audit-logging-fluentd-ds-*`. Clique em **...** > **Remover** para remover um pod.

   - Use o `kubectl` para recriar os pods.

     ```
     kubectl -n kube-system get pod -o wide | grep audit-logging-fluentd-ds- | awk '{print
     $1}' | xargs kubectl delete pod -n kube-system
     ```

   O Kubernetes recria os pods `fluentd` com a configuração atualizada.

APIs

Acessar e modificar seu cluster do IBM® Cloud Private usando APIs.

- Preparando para executar comandos da API do componente ou de gerenciamento
- APIs do componente
- APIs IAM
- APIs do Helm
- API de gerenciamento de imagem
- API do Vulnerability Advisor
- APIs do Key Management Service

Preparando para executar comandos da API do componente ou de gerenciamento

Antes de executar comandos da API do componente, recupere o token de autenticação e faça download do certificado de CA para seu cluster.

1. Instale a linha de comandos do Kubernetes (kubectl). Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
2. (Opcional) Instale a interface da linha de comandos (CLI) do IBM Cloud Private e efetue login em seu cluster. Consulte Instalando a CLI do IBM Cloud Private.

3. Recupere os tokens de autenticação. É possível usar a CLI do IBM Cloud Private ou executar comandos curl.

   - Para usar a CLI do IBM Cloud Private, execute o comando a seguir:

     ```
     cloudctl tokens
     ```

     O token de acesso e o token de ID são exibidos:

     ```
     Access token:  Bearer
     jHVcaGjSuWEXGUPeH8WVnoqUyex5kJDlCl4DnZMQN2WseErIoVWDW5cY1Ikvoqdhxkou4bvHYJM77U2FZf2aKbo3h
     2FeH1OJpaEwj8rpGiDLusaXc54rJbnefbjAv4OECvk2tg9gIeJAx6DZlmlknDXB6zM0bkGtqayP94L8gSXtzPoW6m
     iuoR8dFD5Du63OjVBVZljw5ajOWBtvYzR1ttamH4SRMflEVWC3AxcBhtkadMCAsdWFi1KRGUoVM5k0TkuPFP7AAX8
     vryEpakeVqBQfbR2m0OlRBTRbkNJoacPp7AesyGLBjPVElQs01FCz3xhMNHkQJAFT0T0xhbhDj3DgenAjJd5eSNWx
     dTLSfhOaQClfx6BLq5Z0e8S0Uhp2NSWqNVkU49fHuTHW2UwxG84nvwpz4ZxYJaP3m6DZv5lTwKcbQkB3w9cLr8pgU
     aC94IZ9gJhYvprMRlNKz4etFgNPmhA4T1NS8NVVgMHBYOptv1qAdKSNzZ2l6V23h2BquFpVQ2MqE6Lcx0N0j1ZoRV
     uYo1qTrIkOpEBNbn94b3PHNJMVO2v1NqV85G6uNgPdvv85eneHItsIfUc5yXMeYLXZ017trlhxj43lLVsEyUaM7S5
     dfykayqVgsaJ9faHYz6F14oRKJCbwhg7y4ybxxxR5KtCKihXgf5QJJXYmsrimudD8KXMVsQKEyHFETzVR7eNMRZvc
     ohFKBPwZdIntkcpLygMmZK4Gaz6pD3t4PTkVXKAWQsPTepJ57FfP9kNkPU6BDxf9X7skBrsRFD3ldCdFUMi0lxOmk
     BakmXHLODRmmZl73bKuIKFT7lZAXY0Rn3d5NzXcyzVODRNXnMyRW9c2NLvhYQe0ltdR8dIDtRbsh1AQEC5P1eI9XG
     ```

```
hBLcAqhU0oBe7k5rMzQXL44TtYMo9NccsgMjNsfToyCqezocgfmLYjqZOUnRbBG7ymn5FJECzOd8X1KagaXqzlSmD
tD7cRzZms3iA8FE85cWaK
ID token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXNoIjoiN2p2MmFjeGtucHVpcXo1MXRna2giLCJyZWFs
bU5hbWUiOiJjdXN0b21SZWFsbSIsInVuaXF1ZVNlY3VyaXR5TmFtZSI6ImFkbWluIiwiaXNzIjoiaHR0cHM6Ly9te
WNsdXN0ZXIuaWNwOjk0NDMvb2lkYy9lbmRwb2ludC9PUCIsImF1ZCI6ImY4YjVjZGE1YTgzZjg4NjZhOTIxMTQ2MG
U5YTk4YzQ4IiwiZXhwIjoxNTA4MjYwODc4LCJpYXQiOjE1MDgyNjA4NzgsInN1YiI6ImFkbWluIn0.IrLm1R9a4GB
iTG0wYR1JhGqT4HSArN3gPHhPPTC4ZuS46LulRQCBksxh9I59uT4pYcqhd0qJ_xp9Ys1H8xLsq1zKSI0W2KAzuFkI
bXQiK9Q6_Z3oQOHE8XMG7Xfb0R8B4TgbTjQ3XWkEkXsyeliXk0l7mqlVIgTFbXx8nqcoFbXhmH7ZQukj73lMQ0AyK
KPpJktWtPCLpugtiTA0nkKUodncvHdSw43bmVQuGsQ_kRhhr8Ka8y_olYcBtYUSAKqdwiGPu6O0Qk-
57FCiUmX4W9pjLRAR9EmILY9RqJAsH5kE11kYHPTO2fu-B6omzw2eKxhjZYHMIPmxUciiBRB9Pw
```

Esses tokens são armazenados no arquivo `/<user_folder>/.cloudctl/config.json` enquanto você está com login efetuado na CLI, em que `<user_folder>` é o caminho para o diretório do usuário, como `/Users/my_username` no macOS.

- Para usar curl, execute o seguinte comando, em que `<Cluster Master Host>` e `<Cluster Master API Port>` são definidos em Terminais principais:

```
curl -k -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d
"grant_type=password&username=admin&password=admin&scope=openid" https://<Cluster Master
Host>:<Cluster Master API Port>/idprovider/v1/auth/identitytoken
```

O comando retorna um `access_token`, `refresh_token` e `id_token`, conforme mostrado no exemplo a seguir:
Token de acesso

```
{"access_token":"eb837eaf32459b711945a9d2259880119056e805ff0d2f36421cc171f94e58fef349f000
5d217e36889b62271d9f00fc7cb5b1fe5a86546d9dee8e22bca39b8f90d6cb61dae7fc383447823a09e380fee
efba5bea0c994408470a49db0df32ddc2b0cca9381519e60a63daae9b87ebfe9400b0c4af818b7f7d6c32e214
65909efc8aa02804808f23ff96ac342b3b1c35230ac8858dbcb7979995d7044c7b9cb05945c91b63a93870364
1e0fded339fb4c22e2383743a94a30c41892804193744e0c0f020909f9579555bf691b240fafb558f76877fe0
cb88ecbb3266fedbc7c541129270f67784d11ed658998b536841e0fdfc50a9ad056d2cabf717cb13326e4f620
a6ce172d8da4701b820c5ffe23223e7fb5725b244a1dd45538a0c7ca09a643759aaa2d8585a28689cae968ab3
328351e3c38a8b199040067ca5837169ce62a88282d1c8551d762fbdff77727cb51dd62213cef58dfb88e304a
bbc48063246b7e9f39650a0ac86f6c72973b702b79faf34b68afb9412c9e0e56b104e12bf1ea3764faeac258f
e1c3e896da412607a71ad8b4224efcfa0eafbc15f5e7af5b8baa41163c220419c7249e9652ca7b5692b42cbe4
c7d88c18d77440ec350582f51880e7354eed76ebd8ce760b27a6ca5808c6fc51ef843ee5d98e5bbafaeb40963
01853fa5fdc876275defd3ecabd0eb656c7cd2441e523e0c5468a1f261fb44","token_type":"Bearer","ex
pires_in":43199,"scope":"openid",
```

Token de atualização

```
"refresh_token":"6q4griAg9yCiGINQvF0Dp7N9hqXhcXZrAsqWWYgl6XQ80Uexsq",
```

Token de ID

```
"id_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXNoIjoiYWRmZDc4MmEwOTc1ZTNmMzc2
ZTkxZTI3YjJkNTYxZmQ0OTNiNTQzMSIsInJlYWxtTmFtZSI6ImN1c3RvbVJlYWxtIiwidW5pcXVlU2VjdXJpdHlOY
W1lIjoiYWRtaW4iLCJpc3MiOiJodHRwczovL215Y2x1c3Rlci5pY3A6OTQ0My9vaWRjL2VuZHBvaW50L09QIiwiYX
VkIjoiMGQzYzA3MTc5OTYxYmEzMWEyODY5NDU0NDQwM2E0NDYiLCJleHAiOjE1NTQ5MTQ2NTIsImlhdCI6MTU1NDg
4NTg1Miwic3ViIjoiYWRtaW4iLCJ0ZWFtUm9sZU1hcHBpbmdzIjpbXX0.CnT0qWECpJR9R16W-
IOqrXjSJR8DelRsDUXcX6hy_I0DPQ7hU55Bhcq6UChEg3qiWWRbKwrFIxikXPjEjw2B9oziEd8U8AEO-
4LEaXOpc5Lk1shvyxBQFDDgyUwgyGb-
erRbO_Sl1K4xotuTLg4nhoydwTXs7lZn7GC4UW8j1qkhlbFe5iLgKidCZsjyPo-
2GNYEQn0ufHH3KCR4DkHi6GX2RUxisNecwDzNl9P5JSyjlS-
r5QUZJ0b0DytKuY5HxpswpIFaO9U8JlYAFoOZ18eO_CzERHRQ_Ii1ePmagGAk-
eLJjmCNqY1zynfpEUuKlWUR5rVGHGzSbGA8J4CLvg"}
```

4. Armazene o token de autenticação em uma variável. É possível acessar APIs do IBM Cloud Private, incluindo Kubernetes e Heapster, especificando um token de autenticação no cabeçalho da solicitação. Execute o seguinte comando, em que `<ID token>` é o token de ID exibido:

```
export ID_TOKEN=<ID token>
```

5. Armazene o token de acesso em uma variável. Inclua o conteúdo completo do token de acesso, incluindo o valor de Acesso. Por exemplo, do token de acesso na saída de comando Curl na etapa 3, você deve incluir o valor do token de `"eb837e` em `"openid"`. É possível acessar APIs de gerenciamento de usuários do IBM Cloud Private especificando o token de acesso no cabeçalho da solicitação. Execute o seguinte comando, em que `<Access token>` é o seguinte token de acesso exibido:

```
export ACCESS_TOKEN=<Access token>
```

6. Obtenha uma cópia do certificado de CA para seu cluster.

- Se for possível acessar o nó de inicialização, o certificado de autoridade de certificação será `/<installation_directory>/cluster/cfc-certs/root-ca/ca.crt`.
- Para usar a CLI do IBM Cloud Private:

  1. Assegure-se de ter efetuado login com cloudctl, conforme necessário. Isso coloca os certificados de cluster em um diretório de configuração cloudctl.
  2. Confirme se o certificado de autenticação está disponível. Execute o seguinte comando, em que `<user_folder>` é o caminho para o diretório inicial do usuário, como `/Users/my_username` no macOS e `<cluster>` é o nome do cluster. Este caminho de arquivo é a variável `<certificate_path>` usada em uma etapa posterior:

     ```
     ls <user_folder>/.cloudctl/clusters/<cluster_name>
     ```

     O arquivo `ca.crt` é exibido, conforme mostrado na saída a seguir:

     ```
     ca.pem        cert.pem    key.pem        kube-config    kube-config.bat
     ```

APIs do componente

É possível acessar APIs para vários dos componentes que o IBM® Cloud Private usa.

- API do Kubernetes
- API do Docker Registry V2
- Prometheus API
- Helm API

API do Helm

É possível acessar APIs para a versão 2 ou para a versão 1 do Helm API.

Nota: é necessário usar a versão 2 das APIs, se possível. A versão 1 pode ser descontinuada em um futuro próximo.

- APIs de REST para o Helm API versão 2
- APIs de REST para o Helm API versão 1

API do Kubernetes

O IBM® Cloud Private Versão 3.2.0 usa o Kubernetes versão 1.13.5 para gerenciar serviços de longa execução.

É possível acessar os documentos da API do Kubernetes nos locais a seguir:

- Referência de API do Kubernetes no GitHub
- APIs do Kubernetes

Executando comandos da API do Kubernetes

Ao executar um comando da API do Kubernetes, deve-se obter e especificar o cabeçalho de autenticação. Consulte Preparando para executar comandos da API do componente ou de gerenciamento.

Depois de obter essas informações, é possível executar comandos da API do Kubernetes. Por exemplo, para listar os pods no namespace padrão, execute o comando a seguir:

```
curl -k -H "Authorization:Bearer $ID_TOKEN"  https://<Cluster Master Host>:<Kubernetes API Port>/api/v1/namespaces/default/pods
```

Neste comando:

- `$ID_TOKEN` é a variável que armazena o token de autenticação para o seu cluster
- `<Cluster Master Host>` e `<Kubernetes API Port>` estão definidos em Terminais principais

A saída se assemelha ao código a seguir:

```
{   "kind": "PodList",
"apiVersion": "v1",   "metadata": {     "selfLink": "/api/v1/namespaces
/default/pods", "resourceVersion": "414"   },   "items": [{...}, {...}] }
```

API do Docker Registry V2

É possível executar comandos da API do Docker. O IBM® Cloud Private Versão 3.2.0 usa uma distribuição do Docker que implementa a especificação de API do Docker Registry V2 para gerenciar o armazenamento de imagens do Docker.

O administrador de cluster e o administrador podem acessar a API do Docker Registry. No entanto, os administradores podem acessar apenas os recursos que pertencem ao seu namespace. Os administradores de cluster podem acessar todos os recursos no cluster.

Para obter detalhes sobre a API do Docker Registry V2, consulte a documentação oficial:

- Docker Registry HTTP API V2
- Autenticação do Docker Registry v2 via serviço central

Executando comandos da API do Docker

Ao executar um comando da API do Docker, deve-se obter um certificado de autenticação de um nó do cluster e especificar o token de autenticação.

Você deve copiar o arquivo `/etc/docker/certs.d/<Cluster Master Host>:8500/ca.crt`, o certificado de autenticação, de um nó do cluster para o nó que executa os comandos da API.

`<Cluster Master Host>:<Cluster Master API Port>` são usados para acessar as APIs. Os parâmetros são definidos nos Terminais mestres.

É possível armazenar o token de autenticação na variável `ID_TOKEN` executando os comandos a seguir:

```
export CMD=`curl --cacert /<certificate_path>/ca.crt -s -u admin:admin "https://<Cluster Master
Host>:<Cluster Master API Port>/image-manager/api/v1/auth/token?service=token-
service&scope=registry:catalog:*"`

export ID_TOKEN=$(echo $CMD | python -c 'import sys,json; print json.load(sys.stdin)["token"]')

echo $ID_TOKEN
```

Em seguida, é possível usar o token de autenticação em seus comandos REST. Por exemplo, para obter uma lista de imagens do Docker Registry, execute o comando a seguir:

```
curl --cacert /<certificate_path>/ca.crt -s -H "Authorization: Bearer $ID_TOKEN" "https://<Cluster
Master Host>:8500/v2/_catalog"
```

Nesse comando, `$ID_TOKEN` é a variável que armazena o cookie de autenticação para seu cluster.

A saída se assemelha ao código a seguir:

```
{"repositories":[]}
```

API do Prometheus

O IBM® Cloud Private Versão 3.2.0 usa o Prometheus Versão 2.0 para gerenciar as métricas de destinos de extração.

Tanto o administrador de cluster quanto o administrador da equipe podem acessar a API do Prometheus.

Deve-se acessar o Prometheus por meio do proxy de serviço da API do Prometheus.

Para visualizar os docs da API do Prometheus, consulte API Prometheus HTTP.

Executando comandos da API do Prometheus

Ao executar um comando da API do Prometheus, deve-se obter e especificar o cabeçalho de autenticação. Consulte Preparando para executar comandos da API do componente ou de gerenciamento.

Depois de obter essas informações, é possível executar comandos da API do Prometheus. Por exemplo, para obter o tempo de inicialização para todos os nós, execute o comando a seguir:

```
curl -k -s -X GET -H "Authorization:Bearer $ACCESS_TOKEN" https://<Cluster Master Host>:<Cluster
Master API Port>/prometheus/api/v1/query?query=node_boot_time_seconds
```

- `$ACCESS_TOKEN` é a variável que armazena o token de autenticação para seu cluster.
- `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

A saída assemelha-se ao conteúdo a seguir:

```
{"status":"success","data":{"resultType":"vector","result":[{"metric":
{"__name__":"node_boot_time","app":"monitoring-prometheus", \
"chart":"ibm-icpmonitoring-
1.1.0","component":"nodeexporter","heritage":"Tiller","instance":"9.42.135.189:9100", \
"job":"kubernetes-service-endpoints","kubernetes_name":"monitoring-prometheus-
nodeexporter","kubernetes_namespace":"kube-system", \
"release":"monitoring"},"value":[1523394278.231,"1521476293"]}, \
{"metric":{"__name__":"node_boot_time","app":"monitoring-prometheus","chart":"ibm-icpmonitoring-
1.1.0", \
"component":"nodeexporter","heritage":"Tiller","instance":"9.42.135.84:9100","job":"kubernetes-
service-endpoints", \
"kubernetes_name":"monitoring-prometheus-nodeexporter","kubernetes_namespace":"kube-
system","release":"monitoring"}, \
"value":[1523394278.231,"1521476285"]},{"metric":{"__name__":"node_boot_time","app":"monitoring-
prometheus", \
"chart":"ibm-icpmonitoring-
1.1.0","component":"nodeexporter","heritage":"Tiller","instance":"9.42.78.191:9100", \
"job":"kubernetes-service-endpoints","kubernetes_name":"monitoring-prometheus-
nodeexporter","kubernetes_namespace": \
"kube-system","release":"monitoring"},"value":[1523394278.231,"1521476254"]}]}}
```

APIs do IAM

APIs do Identity and Access management (IAM).

Para acessar APIs do IAM, detalhes de configuração podem ser necessários, como o endereço IP e o número da porta de seu cluster. Para obter informações sobre o cluster, consulte ConfigMap de configuração de cluster.

- API de gerenciamento e autenticação de usuário
- APIs de gerenciamento de serviço
- APIs de integração de serviço e RBAC
- Chaves API do usuário da plataforma
- APIs de conexão única
- APIs de verificação de funcionamento e de versão do serviço

Gerenciamento de usuário e APIs de autenticação

O gerenciamento de usuários e as APIs de autenticação gerenciam usuários e equipes.

Para usar essas APIs, deve-se incluir um cabeçalho de autorização em sua solicitação. É necessário um token de acesso para incluir no cabeçalho de autorização. Para obter o token de acesso, consulte Preparando para executar os comandos da API do componente ou de gerenciamento.

Dados de gerenciamento de usuários e da API de autenticação

Versão da API

1.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

`Porta da API Principal do Cluster`

Caminho Base

`idmgmt/identity/api/v1`

Formato de saída de comando

`application/json`

- Gerenciamento de conta
- Gerenciamento de diretório
- Gerenciamento de grupo de usuários
- Gerenciamento de usuários
- Gerenciamento de equipe

APIs de gerenciamento de conta

APIs para gerenciar contas.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/account`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Obter informações sobre todas as contas

Versão da API
1.0.0
Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/account`

Comando

`GET`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/account'
```

A resposta se assemelha ao código a seguir:

```
[{"id":"id-mycluster-account","name":"mycluster Account","description":"Description for mycluster
Account"}]
```

Obter informações sobre uma conta

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/account/{id}`

Comando

`GET`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/account/id-mycluster-account'
```

A resposta se assemelha ao código a seguir:

```
{"id":"id-mycluster-account","name":"mycluster Account","description":"Description for mycluster
Account","url":"/identity/api/v1/account/id-mycluster-account"}
```

APIs de gerenciamento de diretório

APIs para gerenciar o diretório LDAP.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/directory/ldap`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` são definidos em Terminais principais.

Conectar-se a um diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/directory/ldap/onboardDirectory`

Comando

`POST`

Formato de saída de comando

`application/json`

Nota: no comando curl, deve-se usar uma senha codificada em base64 no parâmetro `"LDAP_BINDPASSWORD"`. Para codificar a senha, use o comando a seguir:

```
echo -n < password> | base64
```

A seguir está uma saída de exemplo:

```
UGFzc3c3cwcmQ=
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header "Authorization: bearer $ACCESS_TOKEN" --header 'Content-Type:
application/json' -d '{"LDAP_ID": "Corp", "LDAP_URL": "ldap://corp.abc.com:389", "LDAP_BASEDN":
"o=ibm.com", "LDAP_BINDDN": "", "LDAP_BINDPASSWORD": "", "LDAP_TYPE": "IBM Tivoli Directory Server",
"LDAP_USERFILTER": "(&(emailAddress=%v)(objectclass=ePerson))", "LDAP_GROUPFILTER": "(&(cn=%v)
(objectclass=groupOfUniqueNames))", "LDAP_USERIDMAP": "*:emailAddress","LDAP_GROUPIDMAP":"*:cn",
"LDAP_GROUPMEMBERIDMAP": "groupOfUniqueNames:uniqueMember"}' 'https://<Cluster Master Host>:<Cluster
Master API Port>/idmgmt/identity/api/v1/directory/ldap/onboardDirectory'
```

Para obter mais informações sobre os parâmetros LDAP, consulte Definindo autenticação LDAP.

A resposta se assemelha ao código a seguir:

```
"8b019a10-daa0-11e7-8dba-bf3c83e12db5"
```

Obter informações sobre um diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/directory/ldap/{ID}`

Comando

`GET`

Formato de saída de comando

`application/json`

Nota: para obter informações sobre um diretório LDAP, deve-se designar o diretório como um recurso para uma equipe. Somente então os membros da equipe podem usar essa API para obter informações sobre o diretório LDAP. Para obter mais informações sobre como designar um recurso a uma equipe, consulte Designar recursos a uma equipe. O formato do CRN do recurso de diretório é `crn:v1:icp:private:<LDAP_ID>:::::Directory:<ID>`, em que `<LDAP_ID>` é o nome da conexão no campo `LDAP_ID` e `<ID>` é o identificador de GUID que é designado à conexão no campo `id`.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json' \
```

```
  "https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/directory/ldap/<LDAP_ID>"
```

A resposta se assemelha ao código a seguir:

```
{"id":"69452b20-bb3d-11e8-98b2-
970b1dcdf410","LDAP_ID":"openldap","LDAP_REALM":"REALM","LDAP_HOST":"corp.abc.com","LDAP_PORT":"389"
,"LDAP_IGNORECASE":"false","LDAP_BASEDN":"dc=ibm,dc=com","LDAP_BINDDN":"cn=admin,dc=ibm,dc=com","LDA
P_TYPE":"Custom","LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))","LDAP_GROUPFILTER":"(&(cn=%v)
(objectclass=groupOfUniqueNames))","LDAP_USERIDMAP":"*:uid","LDAP_GROUPIDMAP":"*:cn","LDAP_GROUPMEMB
ERIDMAP":"groupOfUniqueNames:uniquemember","LDAP_URL":"ldap://corp.abc.com:389","LDAP_PROTOCOL":"lda
p"}
```

O CRN para este recurso de diretório LDAP é `crn:v1:icp:private:openldap:::: Directory:69452b20-bb3d-11e8-98b2-970b1dcdf410`.

Atualizar um diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/directory/ldap/{ID}`

Comando

`PUT`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT \
    -H "Content-type: application/json" \
    -H "Authorization: Bearer ${ACCESS_TOKEN}" \
    "https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/directory/ldap/<LDAP_ID>" \
    -d '{
      "LDAP_ID": "openldap",
      "LDAP_URL": "ldap://corp.abc.com:389",
      "LDAP_BASEDN": "dc=ibm,dc=com",
      "LDAP_BINDDN": "cn=admin,dc=ibm,dc=com",
      "LDAP_BINDPASSWORD": "UGFzc3cwcmQ=",
      "LDAP_TYPE": "Custom",
      "LDAP_USERFILTER": "(&(uid=%v)(objectclass=person))",
      "LDAP_GROUPFILTER": "(&(cn=%v)(objectclass=groupOfUniqueNames))",
      "LDAP_USERIDMAP": "*:uid",
      "LDAP_GROUPIDMAP": "*:cn",
      "LDAP_GROUPMEMBERIDMAP": "groupOfUniqueNames:uniquemember"
    }'
```

A resposta se assemelha ao código a seguir:

```
{"id":"e02d78b0-72df-11e8-8d5e-
93a06ac1d3fc","LDAP_ID":"openldap","LDAP_REALM":"REALM","LDAP_HOST":"9.37.204.115","LDAP_PORT":"389"
,"LDAP_IGNORECASE":"false","LDAP_BASEDN":"dc=ibm,dc=com","LDAP_BINDDN":"cn=admin,dc=ibm,dc=com","LDA
P_TYPE":"Custom","LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))","LDAP_GROUPFILTER":"(&(cn=%v)
```

(objectclass=groupOfUniqueNames))","LDAP_USERIDMAP":"*:uid","LDAP_GROUPIDMAP":"*:cn","LDAP_GROUPMEMB
ERIDMAP":"groupOfUniqueNames:uniquemember","LDAP_URL":"ldap://corp.abc.com:389","LDAP_PROTOCOL":"lda
p"}

Listar conexões LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/list

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/directory/ldap/list'
```

A resposta se assemelha ao código a seguir:

[{"id":"8b019a10-daa0-11e7-8dba-
bf3c83e12db5","LDAP_ID":"Corp","LDAP_REALM":"REALM","LDAP_HOST":"corp.abc.com","LDAP_PORT":"389","LD
AP_BASEDN":"o=ibm.com","LDAP_BINDDN":"","LDAP_BINDPASSWORD":"","LDAP_TYPE":"IBM Tivoli Directory
Server","LDAP_USERFILTER":"(&(emailAddress=%v)(objectclass=ePerson))","LDAP_GROUPFILTER":"(&(cn=%v)
(objectclass=groupOfUniqueNames))","LDAP_USERIDMAP":"*:emailAddress","LDAP_GROUPIDMAP":"*:cn","LDAP_
GROUPMEMBERIDMAP":"groupOfUniqueNames:uniqueMember","LDAP_URL":"ldap://corp.abc.com:389","LDAP_PROTO
COL":"ldap"}]

Excluir um diretório LDAP

Nota: use esta API para excluir um diretório LDAP quando apenas um diretório LDAP está configurado.

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/offboardDirectory

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/directory/ldap/offboardDirectory'
```

A resposta se assemelha ao código a seguir:

```
"Count: 1"
```

Excluir diretório LDAP por ID

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/offboardDirectory?id={LDAP_ID}

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H 'Content-Type: application/json' \
  "https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/directory/ldap/offboardDirectory?id=<LDAP_ID>"
```

A resposta se assemelha ao código a seguir:

```
{"count":1}
```

Procurar por grupos de usuários no diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/{id}/fetchUserGroups

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" "https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/directory/ldap/{LDAP ID}/fetchUserGroups?
searchString=*sec*"
```

A resposta se assemelha ao código a seguir:

```
[{"cn":"security","dn":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com"},
{"cn":"cloudSecurity","dn":"cn=cloudSecurity,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com"}]
```

Procurar por usuários em seu diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/{id}/fetchUsers

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" "https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/directory/ldap/{LDAP ID}/fetchUsers?
searchString=*test*"
```

A resposta se assemelha ao código a seguir:

```
[{"cn":"TestUser","dn":"uid=testuser,ou=people,dc=ibm,dc=com"},
{"cn":"test1","dn":"uid=test1,ou=people,dc=ibm,dc=com"}]
```

Importar grupos de usuários de seu diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/{id}/importUserGroups

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header "Authorization: Bearer $ACCESS_TOKEN" --header 'Content-Type:
application/json' --header 'Accept: application/json' -d '{ "baseDN":
"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com" }' "https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/directory/ldap/fb01b1d0-1fa4-11e8-80d6-
15882dd657a0/importUserGroups"
```

A resposta se assemelha ao código a seguir:

```
{"name":"security","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com"}
```

Importar usuários de seu diretório LDAP

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/directory/ldap/{id}/importUser

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header "Authorization: Bearer $ACCESS_TOKEN" --header 'Content-Type:
application/json' --header 'Accept: application/json' -d '{ "baseDN":
```

```
"uid=testuser,ou=people,dc=ibm,dc=com" }' "https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/directory/ldap/fb01b1d0-1fa4-11e8-80d6-15882dd657a0/importUser"'
```

A resposta se assemelha ao código a seguir:

```
{"userId":"testuser","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","firstName":"TestUser","lastName":"","email":"testuser@ibm.com","lastLogin":"","userBa
seDN":"uid=testuser,ou=people,dc=ibm,dc=com","type":"LDAP","_id":"testuser","loopback__model__name":
"Users"}
```

APIs de gerenciamento de grupos de usuários

APIs para gerenciar grupos de usuários.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/usergroup`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Obter Todos os Grupos de Usuários

Versão da API
1.0.0
Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/usergroup`

Comando

`GET`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/usergroup'
```

A resposta se assemelha ao código a seguir:

```
[{"name":"security","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com"}]
```

Excluir um grupo de usuários

Versão da API
1.0.0
Componentes do URI da API

Esquema

`HTTPS`

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/usergroup/{id}

Comando

DELETE

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/usergroup/<usergroup name>'
```

A resposta se assemelha ao código a seguir:

```
{"count":1}
```

APIs de gerenciamento de usuário

APIs para gerenciar usuários.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/users`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Obter informações sobre todos os usuários

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/users

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/users'
```

A saída se assemelha ao código a seguir:

```
[{"userId":"aaa","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","firstName":"AAA","lastName":"","email":"aaa@ibm.com","lastLogin":"","userBaseDN":"uid
=aaa,ou=people,dc=ibm,dc=com","type":"LDAP"},{"userId":"testuser","directoryId":"fb01b1d0-1fa4-11e8-
80d6-
15882dd657a0","firstName":"TestUser","lastName":"","email":"testuser@ibm.com","lastLogin":"","userBa
seDN":"uid=testuser,ou=people,dc=ibm,dc=com","type":"LDAP"},{"userId":"bbb","directoryId":"fb01b1d0-
1fa4-11e8-80d6-
15882dd657a0","firstName":"BBB","lastName":"","email":"bbb@ibm.com","lastLogin":"","userBaseDN":"uid
=bbb,ou=people,dc=ibm,dc=com","type":"LDAP"}]
```


Obtenha informações sobre todos os membros da equipe à qual o usuário pertence

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/users?filter={\"{id}\":\"{name}\"}

Comando

GET

Formato de saída de comando

application/json

No exemplo a seguir, é possível obter todos os membros da equipe à qual Tom pertence.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/users?filter=\{\"id\":\"tom\"\}'
```

A saída se assemelha ao código a seguir:


```
[{"userId":"icpuser20","firstName":"Icpuser20","lastName":"","email":"icpuser20@ibm.com","directoryI
d":"d4b58be0-3426-11e9-b2dc-
0964dfea827a","userBaseDN":"uid=icpuser20,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Viewer"}]},
{"userId":"tom","firstName":"Tom","lastName":"Sen","email":"tom@ibm.com","directoryId": "e02d78b0-
72df-11e8-8d5e-
93a06ac1d3fc","lastLogin":"","deleted":false,"userBaseDN":"uid=tom,ou=people,dc=ibm,dc=com","type":"
LDAP",}]
```


Obtenha as equipes às quais um usuário é designado

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/equipes

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl --header 'Content-Type: application/json' --header 'Accept: application/json' -H
"Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams' --insecure
```

A saída se assemelha ao código a seguir:

```
[{"teamId":"testteam","name":"testteam","users":
[{"userId":"ibmuser1","lastLogin":"","deleted":false,"userBaseDN":"uid=ibmuser1,ou=people,dc=ibm,dc=
com","type":"LDAP","roles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"}]},
{"userId":"tom","firstName":"Tom","lastName":"","email":"tom@ibm.com","userBaseDN":"uid=tom,ou=peopl
e,dc=ibm,dc=com","roles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"}]}],"usergroups":
[],"resources":[],"serviceids":[],"directoryList":["9de4fb10-2868-11e9-97f4-8f832f9bc6f9"]}]
```

Obtenha o número de equipes às quais um usuário é designado

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/teams/count

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'Authorization: Bearer $ACCESS_TOKEN' 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams/count' --insecure
```

A saída se assemelha ao código a seguir:

```
1
```

Obtenha os diretórios aos quais um usuário tem acesso

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/teams/directories

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' --header
"Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams/directories' --insecure
```

A saída se assemelha ao código a seguir:

```
[{"id":"311d0730-65b6-11e9-bfc1-
979fb092ad2e","LDAP_ID":"customLdap","LDAP_REALM":"REALM","LDAP_HOST":"9.37.204.115","LDAP_PORT":"38
9","LDAP_IGNORECASE":"false","LDAP_BASEDN":"dc=ibm,dc=com","LDAP_BINDDN":"cn=admin,dc=ibm,dc=com","L
DAP_TYPE":"Custom","LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))","LDAP_GROUPFILTER":"(&(cn=%v)
(objectclass=groupOfUniqueNames))","LDAP_USERIDMAP":"*:uid","LDAP_GROUPIDMAP":"*:cn","LDAP_GROUPMEMB
ERIDMAP":"groupOfUniqueNames:uniqueMember","LDAP_URL":"ldap://9.37.204.115:389","LDAP_PROTOCOL":"lda
p"}]
```

Obter a maior função que é designada a um usuário nas equipes

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams/highestRole

Comando

GET

Formato de saída de comando

application/json

Nota: apenas o usuário com login efetuado pode visualizar o maior função que é designada ao usuário.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' --header
'Authorization: Bearer $ACCESS_TOKEN' 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams/highestRole' --insecure
```

A saída se assemelha ao código a seguir:

```
"Operador"
```

Obtenha a função mais alta designada a um usuário e CRN entre as equipes

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams/highestRole?crn={CRN}

Comando

GET

Formato de saída de comando

application/json

Nota: apenas o usuário com login efetuado pode visualizar o maior função que é designada ao usuário.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' --header
'Authorization: Bearer $ACCESS_TOKEN' 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams/highestRole?
crn=crn%3Av1%3Aicp%3Aprivate%3Ak8%3Amycluster%3An%2Fdefault%3A%3A%3A' --insecure
```

A saída se assemelha ao código a seguir:

```
"Operador"
```

Obter os recursos de equipe que são designados a um usuário

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

`/idmgmt/identity/api/v1/teams/resources`

Comando

`GET`

Formato de saída de comando

`application/json`

Nota: apenas o usuário com login efetuado pode visualizar os recursos que são designados ao usuário.

O comando curl de amostra se assemelha ao código a seguir:

```
curl --header 'Content-Type: application/json' --header 'Accept: application/json' -H
"Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams/resources' --insecure
```

A saída se assemelha ao código a seguir:

```
[{"crn":"crn:v1:icp:private:k8:mycluster.icp:n/kube-
system:::","serviceName":"k8","region":"mycluster.icp","namespaceId":"kube-
system","scope":"namespace","actions":"CRUD"}]
```

Obtenha os recursos da equipe que estão designados a um usuário por tipo de recurso

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/idmgmt/identity/api/v1/teams/resources?resourceType={filter-type}`

Comando

`GET`

Formato de saída de comando

`application/json`

Nota: apenas o usuário com login efetuado pode visualizar os recursos que são designados ao usuário.

Os valores de tipo de filtro válidos são `namespace`, `helm-charts`, `repo`, `clusterserviceclass`, `clusterserviceplan` e `Directory`.

O comando curl de amostra se assemelha ao código a seguir:

```
curl --header 'Content-Type: application/json' --header 'Accept: application/json' -H
"Authorization: Bearer $ACCESS_TOKEN"
'https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/teams/resources?
resourceType=namespace' --insecure
```

A saída se assemelha ao código a seguir:

```
[{"crn":"crn:v1:icp:private:k8:mycluster.icp:n/kube-
system:::","serviceName":"k8","region":"mycluster.icp","namespaceId":"kube-
system","scope":"namespace","actions":"CRUD"}]
```

Obtenha os recursos da equipe designados a um tipo de recurso por tipo de recurso e tipo de ação

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/idmgmt/identity/api/v1/teams/resources?resourceType={filter-type}&actionType={action-type}`

Comando

`GET`

Formato de saída de comando

`application/json`

Nota: apenas o usuário com login efetuado pode visualizar os recursos que são designados ao usuário.

Os valores de tipo de filtro válidos são `namespace`, `helm-charts`, `repo`, `clusterserviceclass`, `clusterserviceplan` e `Directory`.
Os valores de tipo de ação válidos são `visualizar` e `implementar`.

O comando curl de amostra se assemelha ao código a seguir:

```
curl --header 'Content-Type: application/json' --header 'Accept: application/json' -H
"Authorization: Bearer $ACCESS_TOKEN"
'https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/teams/resources?
resourceType=repo&actionType=deploy' --insecure
```

A saída se assemelha ao código a seguir:

```
[{"crn":"crn:v1:icp:private:helm-catalog:mycluster:r/local-charts::helm-repos:","serviceName":"helm-
catalog","region":"mycluster","resourceType":"helm-repos","repoId":"local-
charts","scope":"repo","actions":"CRUD"}]
```

Obtenha mapeamentos de função de equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/idmgmt/identity/api/v1/teams/roleMappings`

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' --header
"Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/identity/api/v1/teams/roleMappings' --insecure
```

A saída se assemelha ao código a seguir:

```
["icp:testteam:operator","icp:default:member"]
```

Excluir um Usuário

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/users/{id}

Comando

EXCLUIR

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/users/aaa'
```

A saída se assemelha ao código a seguir:

```
"Count: 1"
```

Obtenha as informações de contas dos usuários

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/identity/api/v1/users/{user}/getAccounts

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/users/testuser/getAccounts'
```

A saída se assemelha ao código a seguir:

```
{"userId":"testuser","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","firstName":"TestUser","lastName":"","email":"testuser@ibm.com","lastLogin":"","userBa
seDN":"uid=testuser,ou=people,dc=ibm,dc=com","type":"LDAP","url":"/identity/api/v1/users/testuser"}
```

APIs de gerenciamento de equipe

APIs para gerenciar equipes.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/teams`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Criar uma Equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
 curl -k -X POST --header 'Content-Type: application/json' --header "Authorization: bearer
$ACCESS_TOKEN" -d '{"teamId":"test-team","name":"Test Team"}' https://<Cluster Master Host>:<Cluster
Master API Port>/idmgmt/identity/api/v1/teams
```

A saída se assemelha ao código a seguir:

```
{"teamId":"test-team","name":"Test Team","users":[],"usergroups":[]}
```

Designar usuários e grupos de usuários para uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams/{team-ID}

Comando

PUT

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT --header "Authorization: Bearer $ACCESS_TOKEN" --header 'Content-Type:
application/json' --header 'Accept: application/json' -d '{"teamId":"test-team","name":"Test
Team","users":[{"userId":"testuser","userBaseDN":"uid=testuser,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}],"usergroups":
[{"name":"security","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com",
"roles":[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}]}' "https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams/test-team"
```

A resposta se assemelha ao código a seguir:

```
4{"teamId":"test-team","name":"Test Team","users":
[{"userId":"testuser","userBaseDN":"uid=testuser,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}],"usergroups":
[{"name":"security","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com",
"roles":[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}]}
```

Designar recursos a uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams

Comando

```
POST
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header
"Authorization: bearer $ACCESS_TOKEN" -d '{"crn":
"crn:v1:icp:private:k8:mycluster.icp:n/default:::"}' 'https://<Cluster Master Host>:<Cluster Master
API Port>/idmgmt/identity/api/v1/teams/<team-ID>/resources' --insecure
```

O formato do recurso que você está designando à equipe é `"crn:v1:icp:private:k8:mycluster.icp:n/default:::"`, em que `mycluster.icp` é o valor `cluster_ca_domain` usado no arquivo `config.yaml` durante a instalação do IBM® Cloud Private. No comando de amostra, o namespace `default` é designado à equipe.

Nota: o namespace `default` não deve ser usado no ambiente de produção.

A resposta se assemelha ao código a seguir:

```
{"crn":"crn:v1:icp:private:k8:mycluster.icp:n/default:::","serviceName":"k8","region":"mycluster.icp
","namespaceId":"default"}
```

Incluir recursos do gráfico do Helm em uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
cluster_lb_address
```

Caminho

```
/helm-api/api/v2/releasesCRNs
```

```
/helm-api/api/v2/charts
```

```
/helm-api/api/v2/repos
```

Comando

```
POST
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header
"Authorization: bearer $ACCESS_TOKEN" -d '{"crn":"crn:v1:icp:private:helm-catalog:mycluster:r/local-
charts::helm-repos:"}' "https://mycluster.icp:8443/idmgmt/identity/api/v1/teams/team-$i/resources" -
-insecure
```

O formato do recurso que você está designando à equipe é `"crn:v1:icp:private:helm-catalog:mycluster:r/local-charts::helm-repos:"`. No comando de amostra, o repositório `local-charts` é designado à equipe.

A resposta se assemelha ao código a seguir:

```
{"crn":"crn:v1:icp:private:helm-catalog:mycluster.icp:r/local-
charts:::","serviceName":"k8","region":"mycluster.icp","repository":"local-charts","scope":"helm-
repos"}
```

Obter informações sobre uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams/{id}

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams/test-team'
```

A saída se assemelha ao código a seguir:

```
4{"teamId":"test-team","name":"Test Team","users":
[{"userId":"testuser","userBaseDN":"uid=testuser,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}],"usergroups":
[{"name":"security","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com",
"roles":[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}]}
```

Obter informações sobre todas as equipes

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams'
```

A saída se assemelha ao código a seguir:


```
[ { Team","users":[{"userId":"testuser","userBaseDN":"uid=testuser,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Operator"}]},
{"userId":"aaa","firstName":"AAA","lastName":"","email":"aaa@ibm.com","userBaseDN":"uid=aaa,ou=peopl
e,dc=ibm,dc=com","roles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"}]}],"usergroups":
[{"name":"security","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com","roles
":[{"id":"crn:v1:icp:private:iam::::role:Viewer"}]},{"name":"cloudSecurity","directoryId":"fb01b1d0-
1fa4-11e8-80d6-
15882dd657a0","userGroupDN":"cn=cloudSecurity,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Operator"}]}]},{"teamId":"f-122","name":"F122","users":
[{"userId":"aaa","roles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"}]},
{"userId":"bbb","roles":[{"id":"crn:v1:icp:private:iam::::role:Editor"}]},{"userId":"ccc","roles":
[{"id":"crn:v1:icp:private:iam::::role:Editor"}]},{"userId":"ddd","roles":
[{"id":"crn:v1:icp:private:iam::::role:Viewer"}]}],"usergroups":[]},{"teamId":"team-
1","name":"Team1","users":
[{"userId":"aaa","firstName":"AAA","lastName":"","email":"aaa@ibm.com","userBaseDN":"uid=aaa,ou=peop
le,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:ClusterAdministrator"}]}],"usergroups":
[{"name":"security","directoryId":"fb01b1d0-1fa4-11e8-80d6-
15882dd657a0","userGroupDN":"cn=security,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com","roles
":[{"id":"crn:v1:icp:private:iam::::role:Editor"}]}]},{"teamId":"team3","name":"Team3","users":
[],"usergroups":[]},{"teamId":"team2","name":"Team2","users":
[{"userId":"ppp","userBaseDN":"uid=ppp,ou=people,dc=ibm,dc=com","roles":
[{"id":"crn:v1:icp:private:iam::::role:Viewer"}]}],"usergroups":[]}]
```


Obter recursos que são designados a uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams/{id}/resources

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams/{id}/resources'
```

A saída se assemelha ao código a seguir:


```
[{"crn":"crn:v1:icp:private:k8:mycluster:n/default:::","serviceName":"k8","region":"mycluster","name
spaceId":"default","scope":"namespace"}]
```


Atualizar uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams

Comando

PUT

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT --header 'Content-Type: application/json' --header "Authorization: bearer
$ACCESS_TOKEN" -d '{"teamId":"test-team","name":"Test Team","users":[{"userId":"aaa","roles":
[{"id":"crn:v1:icp:private:iam::::role:Administrator"}]},{"userId":"bbb","roles":
[{"id":"crn:v1:icp:private:iam::::role:Editor"}]},{"userId":"ccc","roles":
[{"id":"crn:v1:icp:private:iam::::role:Editor"}]},{"userId":"ddd","roles":
[{"id":"crn:v1:icp:private:iam::::role:Viewer"}]}]}' 'https://<Cluster Master Host>:<Cluster Master
API Port>/idmgmt/identity/api/v1/teams/test-team'
```

A saída se assemelha ao código a seguir:

```
{"teamId":"test-team","name":"Test Team","users":[{"userId":"aaa","roles":
[{"id":"crn:v1:icp:private:iam::::role:Administrator"}]},{"userId":"bbb","roles":
[{"id":"crn:v1:icp:private:iam::::role:Editor"}]},{"userId":"ccc","roles":
[{"id":"crn:v1:icp:private:iam::::role:Editor"}]},{"userId":"ddd","roles":
[{"id":"crn:v1:icp:private:iam::::role:Viewer"}]}],"usergroups":[],"_rev":"2-
9238053d5bc6a27237a444e0a2e2cc5b","_id":"f-122","loopback__model__name":"Team"}
```

Excluir um recurso de uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

idmgmt/identity/api/v1/teams/{id}

Comando

EXCLUIR

Formato de saída de comando

```
application/json
```

Para excluir um recurso de uma equipe, deve-se primeiramente obter todos os recursos para a equipe (plataforma), para que seja possível recuperar o `CRN`. O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams/platform/resources'
```

A saída se assemelha ao código a seguir:

```
[
{"crn":"crn:v1:icp:private:k8:mycluster:n/kube-
system:::","serviceName":"k8","region":"mycluster","namespaceId":"kube-system"},
{"crn":"crn:v1:icp:private:k8:mycluster:n/default:::","serviceName":"k8","region":"mycluster","names
paceId":"default"}
]
```

Em seguida, você deve codificar o `CRN`. É possível usar o comando `urlencode` (no Ubuntu), conforme mostrado no código de amostra a seguir ou é possível usar um script Python.

```
urlencode 'crn:v1:icp:private:k8:mycluster:n/default:::'
```

A saída se assemelha ao código a seguir:

```
crn%3Av1%3Aicp%3Aprivate%3Ak8%3Amycluster%3An%2Fdefault%3A%3A%3A
```

Finalmente, é possível excluir o recurso da equipe (plataforma) usando o `CRN` codificado. O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type:
application/json" --header "Accept: application/json" 'https://<Cluster Master Host>:<Cluster Master
API
Port>/idmgmt/identity/api/v1/teams/platform/resources/rel/crn%3Av1%3Aicp%3Aprivate%3Ak8%3Amycluster%
3An%2Fdefault%3A%3A%3A'
```

Se necessário, é possível obter a lista de recursos para confirmar se o recurso foi removido. O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams/platform/resources'
```

A saída se assemelha ao código a seguir:

```
[
{"crn":"crn:v1:icp:private:k8:mycluster:n/kube-
system:::","serviceName":"k8","region":"mycluster","namespaceId":"kube-system"}
]
```

Excluir uma equipe

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`idmgmt/identity/api/v1/teams/{id}`

Comando

EXCLUIR

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" 'https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/identity/api/v1/teams/a-1'
```

A saída se assemelha ao código a seguir:

```
{"count":1}
```

APIs de gerenciamento de serviço

As APIs de gerenciamento de serviço gerenciam IDs de serviço, chaves API e políticas de serviço.

Para usar essas APIs, deve-se incluir um cabeçalho de autorização em sua solicitação. É necessário um token de acesso para incluir no cabeçalho de autorização. Para obter o token de acesso, consulte Preparando para executar os comandos da API do componente ou de gerenciamento.

- Gerenciamento de ID de serviço
- Gerenciamento de chave API
- Gerenciamento de política de serviço

Serviço APIs de gerenciamento de ID

APIs para gerenciar IDs de serviço.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/serviceids/`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` são definidos em Terminais principais.

Obter informações sobre todos os IDs de serviço

Versão da API
1.0.0
Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/serviceids/

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/serviceids/"
```

A resposta se assemelha ao código a seguir:

```
{1}, "'pageSize": 20, "items": [ {9}acf5d7a -d193-4453-be47-3619b7c25de8", "uuid": "ServiceId-
9acf5d7a-d193-4453-be47-3619b7c25de8"
,"crn":"crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-9acf5d7a-d193-4453-
be47-3619b7c25de8","createdAt":"2018-05-03T14:28+0000","modifiedAt":"2018-05-03T14:28
+0000"},"entity":{"boundTo":"crn:v1:icp:private:k8::n/kube-system:::","name":"pavan-s
erviceid"}},{"metadata":{"iam_id":"iam-ServiceId-d3948112-bded-4189-8537-bc8450a8725d
","uuid":"ServiceId-d3948112-bded-4189-8537-bc8450a8725d","crn":"crn:v1:icp:private:k
8::n/kube-system::serviceid:ServiceId-d3948112-bded-4189-8537-bc8450a8725d","createdA
t":"2018-05-03T14:28+0000","modifiedAt":"2018-05-03T14:28+0000"},"entity":{"boundTo":
"crn:v1:icp:private:k8::n/kube-system:::","name":"iam-pap-test","description":"iam-pa
p-test serviceid"}},{"metadata":{"iam_id":"iam-ServiceId-ee41749e-7be4-44ae-aa21-1b5e
342f9685","uuid":"ServiceId-ee41749e-7be4-44ae-aa21-1b5e342f9685","crn":"crn:v1:icp:p
rivate:k8::n/default::serviceid:ServiceId-ee41749e-7be4-44ae-aa21-1b5e342f9685","crea
tedAt":"2018-05-07T09:18+0000","modifiedAt":"2018-05-07T09:18+0000"},"entity":{"bound
To":"crn:v1:icp:private:k8::n/default:::","name":"pavan-default-serviceid"}}]}
```

Obter informações sobre todos os IDs de serviço que estão ligados a CRNs

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/serviceids/?boundTo={CRN}

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/serviceids/?
boundTo=crn:v1:icp:private:k8::n/kube-system:::"
```

A resposta se assemelha ao código a seguir:

```
{1}, "'pageSize": 20, "items": [ {9}acf5d7a -d193-4453-be47-3619b7c25de8", "uuid": "ServiceId-
9acf5d7a-d193-4453-be47-3619b7c25de8"
,"crn":"crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-9acf5d7a-d193-4453-
be47-3619b7c25de8","createdAt":"2018-05-03T14:28+0000","modifiedAt":"2018-05-03T14:28
+0000"},"entity":{"boundTo":"crn:v1:icp:private:k8::n/kube-system:::","name":"pavan-s
erviceid"}},{"metadata":{"iam_id":"iam-ServiceId-d3948112-bded-4189-8537-bc8450a8725d
","uuid":"ServiceId-d3948112-bded-4189-8537-bc8450a8725d","crn":"crn:v1:icp:private:k
8::n/kube-system::serviceid:ServiceId-d3948112-bded-4189-8537-bc8450a8725d","createdA
t":"2018-05-03T14:28+0000","modifiedAt":"2018-05-03T14:28+0000"},"entity":{"boundTo":
"crn:v1:icp:private:k8::n/kube-system:::","name":"iam-pap-test","description":"iam-pa
p-test serviceid"}}]}
```

Crie um ID de serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/serviceids/

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization:
Bearer $ACCESS_TOKEN" -d '{"boundTo": "crn:v1:icp:private:k8::n/kube-system:::","name":
"test_serviceid","description": "Description for test_serviceid"}' "https://<Cluster Master Host>:
<Cluster Master API Port>/iam-token/serviceids/"
```

A resposta se assemelha ao código a seguir:

```
{"metadata":{"iam_id":"iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f","uuid":"Se
rviceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f","crn":"crn:v1:icp:private:k8::n/kube-sy
stem::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f","createdAt":"2018-05-
08T07:47+0000","modifiedAt":"2018-05-08T07:47+0000"},"entity":{"boundTo":"crn:v1:icp:
private:k8::n/kube-system:::","name":"test_serviceid","description":"Description for
test_serviceid"}}
```

Obter informações sobre um ID de serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/serviceids/{service ID}

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/serviceids/ServiceId-9acf5d7a-
d193-4453-be47-3619b7c25de8"
```

A resposta se assemelha ao código a seguir:

```
{"metadata":{"iam_id":"iam-ServiceId-9acf5d7a-d193-4453-be47-3619b7c25de8","uuid":"Se
rviceId-9acf5d7a-d193-4453-be47-3619b7c25de8","crn":"crn:v1:icp:private:k8::n/kube-sy
stem::serviceid:ServiceId-9acf5d7a-d193-4453-be47-3619b7c25de8","createdAt":"2018-05-
03T14:28+0000","modifiedAt":"2018-05-03T14:28+0000"},"entity":{"boundTo":"crn:v1:icp:
private:k8::n/kube-system:::","name":"pavan-serviceid"}}
```

Excluir um ID de serviço e a chave API associada

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/serviceids/{service ID}

Comando

DELETE

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/serviceids/ServiceId-63f6b26f-
1568-4e3e-b88c-77809cea8c8f"
```

A resposta se assemelha ao código a seguir:

```
Não Código de Resposta conteúdo: 204
```

Atualizar um ID de serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

```
/iam-token/serviceids/{service ID}
```

Comando

```
PUT
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization:
Bearer $ACCESS_TOKEN" -d '{"name": "test_serviceid","description": "Updated Description for
test_serviceid"}' "https://<Cluster Master Host>:<Cluster Master API Port>/iam-
token/serviceids/ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f"
```

A resposta se assemelha ao código a seguir:

```
{"metadata":{"iam_id":"iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f","uuid":"Se
rviceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f","crn":"crn:v1:icp:private:k8::n/kube-sy
stem::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f","createdAt":"2018-05-
08T07:47+0000","modifiedAt":"2018-05-08T07:48+0000"},"entity":{"boundTo":"crn:v1:icp:
private:k8::n/kube-system:::","name":"test_serviceid","description":"Updated Descript
ion for test_serviceid"}}
```

Ligando um ID de serviço a uma equipe

A ligação de um ID de serviço a uma equipe permite que administradores e operadores gerenciem o ID de serviço. Eles podem criar e remover novas Chaves API e acessar políticas.

É possível implementar permissões de função específicas nas suas equipes para IDs de serviço. Para obter mais informações sobre como incluir o RBAC em equipes para IDs de serviço, consulte Incluindo o RBAC em equipes para IDs de serviço

Nota: sua equipe deve ter acesso ao mesmo namespace que está ligado ao ID de serviço.

Os administradores de cluster podem ligar um ID de serviço a uma equipe usando a guia Equipes ou usando a guia IDs de Serviço.

Ligar um ID de serviço usando a guia Equipes

Siga as tarefas para ligar um ID de serviço a sua equipe a partir da guia Equipes:

1. No menu de navegação, clique em **Gerenciar** > **Identidade e Acesso** > **IDs de Serviço**.
2. Selecione o nome do seu ID de serviço e clique na guia **Equipes**.
3. Na guia Equipes, clique em **Incluir equipe**.
4. Selecione a equipe que precisa de acesso para incluí-la no namespace que está ligado ao ID de serviço.
5. Clique em **Incluir** para incluir o ID de serviço nas equipes selecionadas.

   Nota: o console de gerenciamento requer que você inclua mais equipes, caso todas as equipes disponíveis tenham sido incluídas no ID de serviço ou não haja mais equipes que atendam aos requisitos.
6. Para remover equipes que estão ligadas ao ID de serviço a partir da guia Equipes, conclua as etapas a seguir:
   1. Na guia Equipes, passe o cursor do mouse sobre o nome da equipe.
   2. Clique no ícone **Abrir e fechar lista de opções** e selecione **Remover**.

      Seu ID de serviço foi removido da equipe que foi vinculada.

Ligar um ID de serviço usando a guia IDs de Serviço

Conclua as etapas a seguir para ligar um ID de serviço a sua equipe a partir da guia IDs de Serviço:

1. No menu de navegação, clique em **Gerenciar** > **Identidade e Acesso** > **Equipes**.
2. Selecione o nome de sua equipe e clique na guia **ID de Serviço**.
3. Na guia IDs de Serviço, clique em **Incluir IDs de Serviço**.
4. Selecione o ID de serviço a ser incluído na equipe.
5. Clique em **Incluir** para incluir a equipe no ID de serviço.
6. Para remover equipes que estão ligadas ao ID de serviço a partir da guia IDs de Serviço, conclua as etapas a seguir:
    1. Na guia IDs de Serviço, passe o cursor do mouse sobre seu nome de ID de serviço.
    2. Clique no ícone **Abrir e fechar lista de opções** e selecione **Remover**.

       Seu ID de serviço foi removido da equipe que foi vinculada.

API APIs de gerenciamento de chave

APIs para gerenciar chaves.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/apikeys/`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Criar uma chave de API

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/iam-token/apikeys/`

Comando

`POST`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization:
Bearer $ACCESS_TOKEN" -d '{"name": "test_serviceid_apikey","description": "Description for
test_serviceid_apikey ","boundTo": "crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-
63f6b26f-1568-4e3e-b88c-77809cea8c8f"}' "https://<Cluster Master Host>:<Cluster Master API
Port>/iam-token/apikeys/"
```

A resposta se assemelha ao código a seguir:

```
{"metadata":{"uuid":"ApiKey-1c40ff8e-5b33-441e-b06f-d5cb89cd1a88","createdAt":"2018-05-
08T08:13+0000","modifiedAt":"2018-05-08T08:13+0000"},"entity":
{"boundTo":"crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-
```

```
77809cea8c8f","name":"test_serviceid_apikey","description":"Description for
test_serviceid_apikey","format":"APIKEY","apiKey":"YZouCtoSz6zTdg9c7tfNNBl5kvo8Fgz1C__8IrsWtieA"}}
```

Obter a chave API que está ligada a um CRN

Versão da API

1.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/iam-token/apikeys/?boundTo={CRN}::{service ID}
```

Comando

```
GET
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/apikeys/?
boundTo=crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-
77809cea8c8f"
```

A resposta se assemelha ao código a seguir:

```
{"currentPage":1,"pageSize":20,"items":[{"metadata":{"uuid":"ApiKey-b730fe51-66b5-4c60-83a4-
e3e0416d4d86","createdAt":"2018-05-08T08:11+0000","modifiedAt":"2018-05-08T08:11+0000"},"entity":
{"boundTo":"crn:v1:icp:private:k8::n/kube-system::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-
77809cea8c8f","name":"test_serviceid_apikey","description":"Description for test_serviceid_apikey
","format":"APIKEY","apiKey":"OKfUUmsbiK5QF8Yq1_2oiKkDzqAVGaOp504Bnvjn0nxs"}}]}
```

Obter informações sobre uma chave de API

Versão da API

1.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/iam-token/apikeys/{API key}
```

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/apikeys/ApiKey-1c40ff8e-5b33-
441e-b06f-d5cb89cd1a88"
```

A resposta se assemelha ao código a seguir:

```
{"metadata":{"uuid":"ApiKey-1c40ff8e-5b33-441e-b06f-d5cb89cd1a88","createdAt":"2018-0
5-08T08:13+0000","modifiedAt":"2018-05-08T08:13+0000"},"entity":{"boundTo":"crn:v1:ic
p:private:k8::n/kube-system::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f
","name":"test_serviceid_apikey","description":"Description for test_serviceid_apikey
","format":"APIKEY","apiKey":"YZouCtoSz6zTdg9c7tfNNBl5kvo8Fgz1C__8IrsWtieA"}}
```

Atualizar uma chave de API

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/apikeys/{API key}

Comando

PUT

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization:
Bearer $ACCESS_TOKEN" -d '{"name": "test_serviceid_apikey","description": "Updated Description for
test_serviceid_apikey"}' "https://<Cluster Master Host>:<Cluster Master API Port>/iam-
token/apikeys/ApiKey-1c40ff8e-5b33-441e-b06f-d5cb89cd1a88"
```

A resposta se assemelha ao código a seguir:

```
{"metadata":{"uuid":"ApiKey-1c40ff8e-5b33-441e-b06f-d5cb89cd1a88","createdAt":"2018-0
5-08T08:13+0000","modifiedAt":"2018-05-08T08:24+0000"},"entity":{"boundTo":"crn:v1:ic
p:private:k8::n/kube-system::serviceid:ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f
","name":"test_serviceid_apikey","description":"Updated Description for test_servicei
d_apikey","format":"APIKEY","apiKey":"YZouCtoSz6zTdg9c7tfNNBl5kvo8Fgz1C__8IrsWtieA"}}
```

Excluir uma chave de API

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/apikeys/{API key}

Comando

EXCLUIR

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/apikeys/ApiKey-1c40ff8e-5b33-
441e-b06f-d5cb89cd1a88"
```

A resposta se assemelha ao código a seguir:

```
Não Código de Resposta conteúdo: 204
```

Gere um token OpenID Connect (OIDC)

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/apikeys/{API key}

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded" -H "Accept: application/json" -
d 'grant_type=urn:ibm:params:oauth:grant-
type:apikey&apikey=EiAUSgnt32h800nur9BCPnHlIsOK6G95pH6fNR2T7peZ&response_type=cloud_iam'
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/oidc/token"
```

A resposta se assemelha ao código a seguir:

```
{"access_token":"eyJraWQiOiIyMDE3MDUxNS0wMDowMDowMCIsImFsZyI6IlJTMjU2In0.eyJpYW1faWQi
OiJpYW0tU2VydmljZUlkLTYzZjZiMjZmLTE1NjgtNGUzZS1iODhjLTc3ODA5Y2VhOGM4ZiIsImlkIjoiaWFtL
VNlcnZpY2VJZC02M2Y2YjI2Zi0xNTY4LTRlM2UtYjg4Yy03NzgwOWNlYThjOGYiLCJyZWFsbWlkIjoiaWFtIi
```

wic3ViX3R5cGUiOiJTZXJ2aWNlSWQiLCJpYXQiOjE1MjU3NzMyNzQsImV4cCI6MTUyNTg1OTY3NCwiaXNzIjo
iaHR0cHM6Ly9sb2NhbGhvc3Q6MTA0NDMvb2lkYy90b2tlbiIsImdyYW50X3R5cGUiOiJ1cm46aWJtOnBhcmFt
czpvYXV0aDpncmFudC10eXBlOmFwaWtleSIsInNjb3BlIjoib3BlbmlkIiwiY2xpZW50X2lkIjoiZGVmYXVsd
CJ9.Lq68chxg3R7pNbq9Tqd7AUfGYK3Xl8Ec-g6E5oFxgDRKGy3Q-coFNf2X5IDFBqkoX3RqIPdB1gx8oNt2P
cf8vrxnfZCymo949kRtz1fgwUqs7JAtC6etvT54i-Cd_i-M8_2zXI8j0jXeRc58i30bSg34gEHZFikgGuGa3m
J_hxLYUUhtHzN4E_OS578WpQ7GfDkNCSv5aaYSA1fQIwMfF76cTcB8Cb_9d9oUAFyVe5CRGicLuGtULnckVy2
aWLdjM_txu8p7NF_Y5EbD-0plHfrsqpE877jhgTkbtP_p3aBR028q8j8X-a_S5NnIOKsAbAAX_8jIMCBYNRJQ
nXU5Uw","token_type":"Bearer","expires_in":86400,"expiration":1525859674}

Introspecção de um token OIDC

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/oidc/introspect

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header 'Content-Type: application/x-www-form-urlencoded' --header 'Accept:
application/json' -d
'token=eyJraWQiOiIyMDE3MDUxNS0wMDowMDowMCIsImFsZyI6IlJTMjU2In0.eyJyZWFsbWlkIjoiaWFtIiwic3ViX3R5cGUiO
iJTZXJ2aWNlSWQiLCJpYXQiOjE1MjA1ODAyMjcsImV4cCI6MTUyMDY2NjYyNywiaXNzIjoiaHR0cHM6Ly9sb2NhbGhvc3Q6NDQzM
C9vaWRjL3Rva2VuIiwiZ3JhbnRfdHlwZSI6InVybjppYm06cGFyYW1zOm9hdXRoOmdyYW50LXR5cGU6YXBpa2V5Iiwic2NvcGUiO
iJvcGVuaWQiLCJjbGllbnRfaWQiOiJkZWZhdWx0In0.cJs3O2KFMQj7jM2b3p0ieem0qCeL1wxC5WAFPWBVe7jpicvjmLkww7LJy
aT45o_ickH3ehoGCDVyaZZdtYmiMKr2CFdAZvCEbpVKeq2KHqsZVWae_ezjUp2aHyPh9MUjyQKmNaI2dinxqQSHZkXH4nLMrDhsL
3VUYhTI786m6crhESuhndZnCJq3otKhy6xFg1woCIxp9L3gWPth2f4srS9z1d-ZXP02mtyGZJUUZJeQA84dP6OC5QjJVlE-
clIlq_-xDOk4M16vHX8KSPjKhwv2F5gCV32EhZGUUc-PvsRl5SW5xKpyaRP5VGxHEbunu0aPCsA-JtL6ELbi77sagw'
'https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/oidc/introspect'
```

A resposta se assemelha ao código a seguir:

{"active":true,"iss":"https://$MASTER_NODE_IP:10443/oidc/token","realmId":"iam","sub_type":"ServiceI
d","account":
{},"scope":"openid","client_id":"default","iat":1529391495,"exp":1529477895,"grant_type":"urn:ibm:pa
rams:oauth:grant-type:apikey","iam_id":"iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f"}

APIs de gerenciamento de política de serviço

APIs para gerenciar políticas de serviço.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Obter informações sobre todas as funções do sistema definido

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/roles

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/roles"
```

A resposta se assemelha ao código a seguir:

```
{"systemDefinedRoles":[{"crn":"crn:v1:icp:private:iam::::role:Viewer","displayName":"
Viewer","description":"Viewers can take actions that do not change state (i.e. read o
nly)."},{"crn":"crn:v1:icp:private:iam::::role:ClusterAdministrator","displayName":"C
lusterAdministrator","description":"ClusterAdministrators can take all actions includ
ing the ability to manage access control."},{"crn":"crn:v1:icp:private:iam::::role:Ad
ministrator","displayName":"Administrator","description":"Administrators can take all
actions including the ability to manage access control."},{"crn":"crn:v1:icp:private
:iam::::role:Editor","displayName":"Editor","description":"Editors can take actions t
hat can modify the state and create/delete sub-resources."},{"crn":"crn:v1:icp:privat
e:iam::::role:Operator","displayName":"Operator","description":"Operators can take ac
tions required to configure and operate resources."}]}
```

Obter informações sobre todos os serviços registrados

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/services?fields={field name}

Comando

GET

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/services?fields=name"
```

A resposta se assemelha ao código a seguir:


```
{"services":[{"_id":"5aeac66bdc2c4100c82016ba","name":"helm-api-service","enabled":tr
ue,"platformExtenstions":{"supportedAttributes":[{"key":"accountId"},{"key":"serviceN
ame"}],"supportedRoles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"},{"id":
"crn:v1:icp:private:iam::::role:ClusterAdministrator"},{"id":"crn:v1:icp:private:iam:
:::role:Operator"},{"id":"crn:v1:icp:private:iam::::role:Viewer"},{"id":"crn:v1:icp:p
rivate:iam::::role:Editor"}]},"actions":[],"supportedRoles":[]},{"_id":"5aeac677dc2c4
100c82016bb","name":"elasticsearch-service","enabled":true,"platformExtenstions":{"su
pportedAttributes":[{"key":"accountId"},{"key":"serviceName"}],"supportedRoles":[{"id
":"crn:v1:icp:private:iam::::role:Administrator"},{"id":"crn:v1:icp:private:iam::::ro
le:ClusterAdministrator"},{"id":"crn:v1:icp:private:iam::::role:Operator"},{"id":"crn
:v1:icp:private:iam::::role:Viewer"},{"id":"crn:v1:icp:private:iam::::role:Editor"}]}
,"actions":[],"supportedRoles":[]},{"_id":"5aeac678dc2c4100c82016bc","name":"servicemonitoring-
service","enabled":true,"platformExtenstions":{"supportedAttributes":[{"ke
y":"accountId"},{"key":"serviceName"}],"supportedRoles":[{"id":"crn:v1:icp:private:ia
m::::role:Administrator"},{"id":"crn:v1:icp:private:iam::::role:ClusterAdministrator"
},{"id":"crn:v1:icp:private:iam::::role:Operator"},{"id":"crn:v1:icp:private:iam::::r
ole:Viewer"},{"id":"crn:v1:icp:private:iam::::role:Editor"}]},"actions":[],"supported
Roles":[]},{"_id":"5af03e6b078693000abde634","name":"idmgmt","enabled":true,"platform
Extenstions":{"supportedAttributes":[{"key":"accountId"},{"key":"serviceName"}],"supp
ortedRoles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"},{"id":"crn:v1:icp:
private:iam::::role:ClusterAdministrator"},{"id":"crn:v1:icp:private:iam::::role:Oper
ator"},{"id":"crn:v1:icp:private:iam::::role:Viewer"},{"id":"crn:v1:icp:private:iam::
::role:Editor"}]},"actions":[],"supportedRoles":[]},{"_id":"5af03e6b078693000abde635"
,"name":"idprovider","enabled":true,"platformExtenstions":{"supportedAttributes":[{"k
ey":"accountId"},{"key":"serviceName"}],"supportedRoles":[{"id":"crn:v1:icp:private:i
am::::role:Administrator"},{"id":"crn:v1:icp:private:iam::::role:ClusterAdministrator
"},{"id":"crn:v1:icp:private:iam::::role:Operator"},{"id":"crn:v1:icp:private:iam::::
role:Viewer"},{"id":"crn:v1:icp:private:iam::::role:Editor"}]},"actions":[],"supporte
dRoles":[]},{"_id":"5af03e6b078693000abde636","name":"idauth","enabled":true,"platfor
mExtenstions":{"supportedAttributes":[{"key":"accountId"},{"key":"serviceName"}],"sup
portedRoles":[{"id":"crn:v1:icp:private:iam::::role:Administrator"},{"id":"crn:v1:icp
:private:iam::::role:ClusterAdministrator"},{"id":"crn:v1:icp:private:iam::::role:Ope
rator"},{"id":"crn:v1:icp:private:iam::::role:Viewer"},{"id":"crn:v1:icp:private:iam:
:::role:Editor"}]},"actions":[],"supportedRoles":[]},{"_id":"5af03e6b078693000abde637
","name":"identity","enabled":true,"platformExtenstions":{"supportedAttributes":[{"ke
y":"accountId"},{"key":"serviceName"}],"supportedRoles":[{"id":"crn:v1:icp:private:ia
m::::role:Administrator"},{"id":"crn:v1:icp:private:iam::::role:ClusterAdministrator"
},{"id":"crn:v1:icp:private:iam::::role:Operator"},{"id":"crn:v1:icp:private:iam::::r
ole:Viewer"},{"id":"crn:v1:icp:private:iam::::role:Editor"}]},"actions":[],"supported
Roles":[]}]}
```


Obter informações sobre políticas que são designadas a um ID de serviço e escopo

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/scopes/{scope}/service_ids/{service ID}/policies

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/scopes/n%252Fkube-
system/service_ids/iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f/policies"
```

A resposta se assemelha ao código a seguir:

```
{"policies":[{"id":"14cbbc3f-077d-439d-9a24-6fe39263e0ab","roles":[{"id":"crn:v1:icp:
private:iam::::role:Administrator","displayName":"Administrator","description":"Admin
istrators can take all actions including the ability to manage access control."},{"id
":"crn:v1:icp:private:iam::::role:Operator","displayName":"Operator","description":"O
perators can take actions required to configure and operate resources."}],"resources"
:[{"namespaceId":"kube-system","serviceName":"idmgmt"}],"links":{"href":"https://9.37
.239.107:8443/acms/v1/scopes/n%252Fkube-system/service_ids/iam-ServiceId-63f6b26f-156
8-4e3e-b88c-77809cea8c8f/policies/14cbbc3f-077d-439d-9a24-6fe39263e0ab","link":"self"
}}]}
```

Criar uma política de acesso para um ID de serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/scopes/{scope}/service_ids/{service ID}/policies

Comando

POST

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization:
Bearer $ACCESS_TOKEN" -d '{"resources": [{"namespaceId": "kube-system","serviceName":
"idmgmt"}],"roles": [{"id": "crn:v1:icp:private:iam::::role:Administrator"},{"id":
"crn:v1:icp:private:iam::::role:Editor"
}]}' "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/scopes/n%252Fkube-
system/service_ids/iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f/policies"
```

A resposta se assemelha ao código a seguir:

```
{"id":"14cbbc3f-077d-439d-9a24-6fe39263e0ab","roles":[{"id":"crn:v1:icp:private:iam::
::role:Administrator","displayName":"Administrator","description":"Os administradores podem tomar
todas as ações, incluindo a capacidade de gerenciar o controle de acesso."},{"id":"crn:v1:icp
:private:iam::::role:Editor","displayName":"Editor","description":"Editors can take a
ctions that can modify the state and create/delete sub-resources."}],"resources":[{"n
amespaceId":"kube-system","serviceName":"idmgmt"}],"links":{"href":"https://9.37.239.
107:8443/acms/v1/scopes/n%252Fkube-system/service_ids/iam-ServiceId-63f6b26f-1568-4e3
e-b88c-77809cea8c8f/policies/14cbbc3f-077d-439d-9a24-6fe39263e0ab","link":"self"}}
```

Obter informações sobre a política de acesso que é designada a um serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/scopes/{scope}/service_ids/{service ID}/policies/{policy ID}

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/scopes/n%252Fkube-
system/service_ids/iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f/policies/14cbbc3f-077d-439d-
9a24-6fe39263e0ab"
```

A resposta se assemelha ao código a seguir:

```
{"id":"14cbbc3f-077d-439d-9a24-6fe39263e0ab","roles":[{"id":"crn:v1:icp:private:iam::
::role:Administrator","displayName":"Administrator","description":"Os administradores podem tomar
todas as ações, incluindo a capacidade de gerenciar o controle de acesso."},{"id":"crn:v1:icp
:private:iam::::role:Operator","displayName":"Operator","description":"Os operadores podem tomar as
ações necessárias para configurar e operar recursos."}],"resources":[{"namespace
Id":"kube-system","serviceName":"idmgmt"}],"links":{"href":"https://9.37.239.107:8443
/acms/v1/scopes/n%252Fkube-system/service_ids/iam-ServiceId-63f6b26f-1568-4e3e-b88c-7
7809cea8c8f/policies/14cbbc3f-077d-439d-9a24-6fe39263e0ab","link":"self"}}
```

Atualizar uma política de acesso que é designada a um serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/scopes/{scope}/service_ids/{service ID}/policies/{policy ID}

Comando

PUT

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT -H "Content-Type: application/json" -H "Accept: application/json" -H " Authorization:
Bearer $ACCESS_TOKEN" -d '{"resources": [{"namespaceId": "kube-system","serviceName":
"idmgmt"}],"roles": [{"id": "crn:v1:icp:private:iam::::role:Administrator"
},{"id": "crn:v1:icp:private:iam::::role:Operator"}]}' "https://<Cluster Master Host>:<Cluster
Master API Port>/iam-pap/acms/v1/scopes/n%252Fkube-system/service_ids/iam-ServiceId-63f6b26f-1568-
4e3e-b88c-77809cea8c8f/policies/14cbbc3f-077d-439d-9a24-6fe39263e0ab"
```

A resposta se assemelha ao código a seguir:

```
{"id":"14cbbc3f-077d-439d-9a24-6fe39263e0ab","roles":[{"id":"crn:v1:icp:private:iam::
::role:Administrator","displayName":"Administrator","description":"Os administradores podem tomar
todas as ações, incluindo a capacidade de gerenciar o controle de acesso."},{"id":"crn:v1:icp
:private:iam::::role:Operator","displayName":"Operator","description":"Os operadores podem tomar as
ações necessárias para configurar e operar recursos."}],"resources":[{"namespace
Id":"kube-system","serviceName":"idmgmt"}],"links":{"href":"https://9.37.239.107:8443
/acms/v1/scopes/n%252Fkube-system/service_ids/iam-ServiceId-63f6b26f-1568-4e3e-b88c-7
7809cea8c8f/policies/14cbbc3f-077d-439d-9a24-6fe39263e0ab/14cbbc3f-077d-439d-9a24-6fe
39263e0ab","link":"self"}}
```

Excluir uma política de acesso que é designada a um serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/scopes/{scope}/service_ids/{service ID}/policies/{policy ID}

Comando

DELETE

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE -H "Accept: application/json" -H "Authorization: Bearer $ACCESS_TOKEN"
"https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/acms/v1/scopes/n%252Fkube-
system/service_ids/iam-ServiceId-63f6b26f-1568-4e3e-b88c-77809cea8c8f/policies/14cbbc3f-077d-439d-
9a24-6fe39263e0ab"
```

A resposta se assemelha ao código a seguir:

```
Não Código de Resposta conteúdo: 204
```

APIs de Integração de Serviço e RBAC

APIs para integrar um serviço e implementar o controle de ação baseado em função (RBAC).

Para usar essas APIs, deve-se incluir um cabeçalho de autorização em sua solicitação. É necessário um token de acesso para incluir no cabeçalho de autorização. Para obter o token de acesso, consulte Preparando para executar os comandos da API do componente ou de gerenciamento.

`<Cluster Master Host>:<Cluster Master API Port>` são usados para acessar as APIs. Os parâmetros são definidos nos Terminais mestres.

Definir um serviço

Para integrar um serviço, deve-se primeiramente definir o serviço em um arquivo JSON. A definição de serviço também deve incluir as ações que podem ser executadas e as funções que têm permissão para executar a ação.

A seguir estão os arquivos JSON de exemplo.

Exemplo Um

Nome do arquivo: action_role_service1.json

```
{
    "chartName": "chartname1",
    "displayName":
    {
        "padrão": "Ações para Service1"
    }, "actions": [ {
        "id": "POST /service1/api",
        "displayName":
        {
            "padrão": "Ação de serviço criar"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator" ]
    },
    {
        "id": "GET /service1/api",
        "displayName":
        {
            "padrão": "Ação de serviço lida"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor", "crn:v1:icp:private:iam::::role:Viewer" ]
    },
    {
        "id": "PUT /service1/api",
        "displayName":
        {
            "padrão": "Atualização da ação de serviço"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor" ]
    },
    {
        "id": "DELETE /service1/api",
        "displayName":
        {
            "padrão": "Excluir ação de serviço"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", ]
    }], "enabled": true, "supportedAttributes": [ {
        "chave": "cadeia"
    }], "supportedRoles": [ {
        "id": "crn:v1 :icp:private:iam ::::role :ClusterAdministrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Administrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Operator"
    },
    {
```

```
        "id": "crn:v1 :icp:private:iam ::::função :Editor"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Viewer"
    }]
}
```

Exemplo dois

Nome do arquivo: action_role_service2.json

```
{
    "chartName": "chartname2",
    "displayName":
    {
        "padrão": "Ações para Service2"
    }, "actions": [ {
        "id": "action.subaction.create",
        "displayName":
        {
            "padrão": "Ação de serviço criar"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator" ]
    },
    {
        "id": "action.subaction.read",
        "displayName":
        {
            "padrão": "Ação de serviço lida"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor", "crn:v1:icp:private:iam::::role:Viewer" ]
    },
    {
        "id": "action.subaction.update",
        "displayName":
        {
            "padrão": "Atualização da ação de serviço"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", "crn:v1:icp:private:iam::::role:Operator",
"crn:v1:icp:private:iam::::role:Editor" ]
    },
    {
        "id": "action.subaction.delete",
        "displayName":
        {
            "padrão": "Excluir ação de serviço"
        }, "roles": [
            "crn:v1:icp:private:iam::::role:ClusterAdministrator",
"crn:v1:icp:private:iam::::role:Administrator", ]
    }], "enabled": true, "supportedAttributes": [ {
        "chave": "cadeia"
    }], "supportedRoles": [ {
        "id": "crn:v1 :icp:private:iam ::::role :ClusterAdministrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Administrator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Operator"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::função :Editor"
    },
    {
        "id": "crn:v1 :icp:private:iam ::::role :Viewer"
    }]
}
```

Criar o Serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/iam-pap/acms/v1/services/{SERVICE_NAME}`

Comando

`PUT`

Formato de saída de comando

`application/json`

Use o comando a seguir para criar o serviço:

```
curl -k -X PUT -H 'Content-Type: application/json' -H 'Accept: application/json' -H "Authorization:
Bearer ${ACCESS_TOKEN}" -d @<API_action_roles_JSON_file> "https://<Cluster Master Host>:<Cluster
Master API Port>/iam-pap/acms/v1/services/<service_name>"
```

Substitua os valores de parâmetros a seguir no comando:

- `API_action_roles_JSON_file`: o nome do arquivo de definição de serviço. Consulte Definir um Serviço .
- `<Cluster Master Host>:<Cluster Master API Port>`: Os parâmetros estão definidos em Terminais principais.
- `service_name`: nome do serviço que você está criando.

Ativar RBAC pelo controlador de ingresso

Se o seu serviço for autorizado internamente, não será necessário ativar o RBAC no controlador de ingresso.

Se o seu serviço for acessado por meio do controlador de ingresso, será possível ativar o RBAC pelo controlador de ingresso. Conclua estas etapas:

1. Efetue login no nó principal como um usuário com permissões raiz.
2. Configure a CLI do `kubectl`. Consulte Acessando seu cluster a partir da CLI do Kubernetes (kubectl).
3. Obtenha informações sobre os controladores de ingresso.

   ```
   kubectl -n kube-system get ingresso
   ```

4. Atualize o controlador de ingresso de seu serviço.

   a. Edite o recurso.

   ```
   kubectl -n kube-system edit ingress <service1>
   ```

   b. Remova a anotação `auth-type` . A anotação assemelha-se ao código a seguir:

   ```
   icp.management.ibm.com/auth-type: id-token

   icp.management.ibm.com/auth-type: access-token
   ```

   c. Inclua a anotação `authz-type` .

   ```
   icp.management.ibm.com/authz-type: rbac
   ```

   d. Salve as alterações.

5. Reinicie o pod `icp-management-ingresso`.

   i. Obtenha o ID do pod `icp-management-ingress`.

   ```
   kubectl -n kube-system get pods | grep icp-management-ingress
   ```

   ii. Exclua o pod `icp-management-ingresso`.

   ```
   kubectl -n kube-system delete pod <icp-management-ingress-pod-id>
   ```

   Aqui está um comando de exemplo.

   ```
   kubectl -n kube-system delete pod icp-management-ingresso-2kt6d
   ```

   Aguarde um minuto para que o pod seja reiniciado.

   iii. Verifique o status da cápsula. O pod é exibido como `1/1 Em execução`.

   ```
   kubectl -n kube-system get pods | grep icp-management-ingress
   ```

Atualizar um serviço

Para atualizar um serviço, deve-se definir e criar um serviço.

- Definir um Serviço
- Criar o Serviço

Obtenha informações sobre os serviços que são integrados

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/iam-pap/acms/v1/services`

Comando

`GET`

Formato de saída de comando

`application/json`

Para obter informações sobre os serviços que são integrados, execute o seguinte comando:

```
curl -k -X GET -H 'Content-Type: application/json' -H 'Accept: application/json' -H "Authorization:
Bearer $ACCESS_TOKEN" "https://<Cluster Master Host>:<Cluster Master API Port>/iam-
pap/acms/v1/services"
```

Obter informações sobre um serviço

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/services/{service_name}

Comando

GET

Formato de saída de comando

application/json

Para obter informações sobre um serviço, execute o comando a seguir:

```
curl -k -X GET -H 'Content-Type: application/json' -H 'Accept: application/json' -H "Authorization:
Bearer ${ACCESS_TOKEN}" "https://<Cluster Master Host>:<Cluster Master API Port>/iam-
pap/acms/v1/services/<service_name>"
```

Excluir um serviço

Versão da API
1.0.0
Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-pap/acms/v1/services/{service_name}

Comando

EXCLUIR

Formato de saída de comando

application/json

Para excluir um serviço, execute o comando a seguir:

```
curl -k -X DELETE -H 'Content-Type: application/json' -H 'Accept: application/json' -H
"Authorization: Bearer ${ACCESS_TOKEN}" "https://<Cluster Master Host>:<Cluster Master API
Port>/iam-pap/acms/v1/services/<service_name>"
```

Chaves API do usuário da plataforma

Uma chave API é um código exclusivo que é transmitido a uma API para identificar o aplicativo ou usuário de chamada. Eles são usados para rastrear e controlar como a API está sendo usada, por exemplo, para evitar o uso malicioso ou abuso da API. As chaves API geralmente agem como identificadores exclusivos e tokens para autenticação. As chaves API possuem um conjunto de direitos de acesso específicos da identidade associada a eles. As chaves API que são específicas de `user` são conhecidas como chaves API de plataforma.

Para criar e listar chaves API de plataforma, forneça o `User access token` no cabeçalho de autorização e configure `boundTo` como `self`.

`<Cluster Master Host>:<Cluster Master API Port>` são usados para acessar as APIs. Os parâmetros são definidos nos Terminais mestres.

Criar chave API do usuário da plataforma

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --
header "Authorization: Bearer $ACCESS_TOKEN" -d '{"name": "test_platform_apikey", "description":
"Description for test platform apikey ","boundTo": "self"}' 'https://<Cluster Master Host>:<Cluster
Master API Port>/iam-token/apikeys/'
```

A saída se assemelha ao código a seguir:

```
Resposta:
{"metadata":{"uuid":"ApiKey-6050fa05-e591-427f-beea-565c5377b79d","crn":"crn:v1:icp:private:iam-
identity::::apikey:ApiKey-6050fa05-e591-427f-beea-565c5377b79d","createdAt":"2018-08-
13T07:38+0000","modifiedAt":"2018-08-13T07:38+0000"},"entity":
{"name":"test_platform_apikey","description":"Description for test platform apikey
","boundTo":"crn:v1:icp:private:iam-
identity:::IBMid:user:ravi","format":"APIKEY","apiKey":"Oy2nk5QS93sXxccWjt8o38nUrO383_PZ7AnOSZOb26gY
"}}
```

Listar chave API de plataforma que está ligada a um CRN

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET --header 'Accept: application/json' --header "Authorization: Bearer $ACCESS_TOKEN"
'https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/apikeys/?boundTo=self'
```

A saída se assemelha ao código a seguir:

```
Resposta:
{"currentPage":1,"pageSize":20,"items":[{"metadata":{"uuid":"ApiKey-6050fa05-e591-427f-beea-
565c5377b79d","crn":"crn:v1:icp:private:iam-identity::::apikey:ApiKey-6050fa05-e591-427f-beea-
565c5377b79d","createdAt":"2018-08-13T07:38+0000","modifiedAt":"2018-08-13T07:38+0000"},"entity":
{"name":"test_platform_apikey","description":"Description for test platform apikey
","boundTo":"crn:v1:icp:private:iam-
identity:::IBMid:user:ravi","format":"APIKEY","apiKey":"Oy2nk5QS93sXxccWjt8o38nUrO383_PZ7AnOSZOb26gY
"}}]}
```

Atualizar chave API da Plataforma

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X PUT -H 'Content-Type: application/json' -H 'Accept: application/json' -H "Authorization:
Bearer $ACCESS_TOKEN" -d '{"name": "test_platform_apikey","description": "Updated description for
test_platform_apikey"}' 'https://<Cluster Master Host>:<Cluster Master API Port>/iam-
token/apikeys/ApiKey-6050fa05-e591-427f-beea-565c5377b79d'
```

A saída se assemelha ao código a seguir:

```
Resposta:
{"metadata":{"uuid":"ApiKey-6050fa05-e591-427f-beea-565c5377b79d","crn":"crn:v1:icp:private:iam-
identity::::apikey:ApiKey-6050fa05-e591-427f-beea-565c5377b79d","createdAt":"2018-08-
13T07:38+0000","modifiedAt":"2018-08-13T08:55+0000"},"entity":
{"name":"test_platform_apikey","description":"Updated Descrição para
test_platform_apikey","boundTo":"crn:v1:icp:private:iam-
identity:::IBMid:user:ravi","format":"APIKEY","apiKey":"Oy2nk5QS93sXxccWjt8o38nUrO383_PZ7AnOSZOb26gY
"}}
```

Excluir chave API da plataforma

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X DELETE -H 'Accept: application/json' -H "Authorization: Bearer $ACCESS_TOKEN"
'https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/apikeys/ApiKey-1c40ff8e-5b33-
441e-b06f-d5cb89cd1a88'
```

A saída se assemelha ao código a seguir:

```
Resposta: nenhum Código de Resposta de conteúdo: 204
```

API da API da Plataforma Introspect

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header 'Content-Type: application/x-www-form-urlencoded' --header 'Accept:
application/json' -d 'apikey=Oy2nk5QS93sXxccWjt8o38nUrO383_PZ7AnOSZOb26gY' 'https://<Cluster Master
Host>:<Cluster Master API Port>/iam-token/oidc/introspect'
```

A saída se assemelha ao código a seguir:

```
Resposta:
{"active":true,"iss":"https://<cluster_ca_domain>:10443/oidc/token","realmId":"IBMid","sub":"ravi","
account":{},"scope":"openid","client_id":"default","grant_type":"urn:ibm:params:oauth:grant-
type:apikey"}
```

Recuperar token da chave API da plataforma

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded" -H "Accept: application/json" -
d 'grant_type=urn:ibm:params:oauth:grant-
type:apikey&apikey=Oy2nk5QS93sXxccWjt8o38nUrO383_PZ7AnOSZOb26gY&response_type=cloud_iam'
'https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/oidc/token'
```

A saída se assemelha ao código a seguir:

```
Resposta:
{"access_token":"eyJraWQiOiIyMDE3MDUxNS0wMDowMDowMCIsImFsZyI6IlJTMjU2In0.eyJyZWFsbWlkIjoiSUJNaWQiLCJ
zdWIiOiJyYXZpIiwiaWF0IjoxNTM0MTUwOTU3LCJleHAiOjE1MzQyMzczNTcsImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0OjEwNDQ
zL29pZGMvdG9rZW4iLCJncmFudF90eXBlIjoidXJuOmlibTpwYXJhbXM6b2F1dGg6Z3JhbnQtdHlwZTphcGlrZXkiLCJzY29wZSI
6Im9wZW5pZCIsImNsaWVudF9pZCI6ImRlZmF1bHQifQ.eQGgLevdKu7ZTpKKO4bJP2MHDJDTxNmpDNTYLYBepN-48xZmk0l-
3CTd3YC71y9Ve-UNUn5qL9m-B6G25sBc5KmyW2ZU7DS6MKeLrE9-bI-
HVp9GhNJ5DC06TSxCM6zcS26YqM1GXpqxOSTINgm7L9rKN10n9uYwszccha12bVR5ctLtrD8_5er1OaugKewgfXOWR8g1gwljwmm
s2FMnp72wE3DynwG16Bzon7U8FzC4YBmh4mFAHMc7C10553jLoOBMf7iq_3Yj4dNRegZhVpYQBC97A1vatODoNkrXbcuBtA9ujJ4
dzLj76Lam5c_iLTGbjTnwkgcALEUXjCv7kQ","refresh_token":"f-3Mv3cPfhQzMVsBNlzjANu-ltpbVAhMbEGG_5Rwb-
wnl0oEeuHhVT5XpfermzmJILSejwUIx0ncMO5ollt2uI-
ZPG7dm_Lu3GAnlyPUMQKV72BMxUZsaLNDTTsbKRR_1vDwl9Uwix9EmX58on6WYiZ-hswb0_gz6wt1-
OgXVYp8P4zoDIkovrl2DgANkv3k1Q_MW8lul3eJrlzQxZmm61lJ74ixxPdl0yTNndsq3bRWJPafbIs60ikitTt4Cu2BumUbHFqjK
Gq8l7rih9342HIHWf31jrpALKco5w-
K4uWymG3VOT2SzdIj8R_ICofH2AFxwxbTYEXXPo7kzSxQx0ZI1R7rqYSW5x_hJ1_E40_XjjWyF29N-
z4TIxuuG1NaARnoxrk3FopO2H4zEQOJYGMGIutCTHVQVLLWZeQK1AwoV8CyotluqT1maKk48P6ms1tTOhGLIqw64cgJUSRuL393t
kqR3GvsgQgtPCQ5FPQBo2AkVfZzgZvaQLP6yHspku0zludQ1S3PE9aNVryf3_BnnZO0_6AcI-
4cFiu2SiUNKQ==","token_type":"Bearer","expires_in":86400,"expiration":1534237357}
```

Token de acesso Introspect IAM

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X POST --header 'Content-Type: application/x-www-form-urlencoded' --header 'Accept:
application/json' -d
'token=eyJraWQiOiIyMDE3MDUxNS0wMDowMDowMCIsImFsZyI6IlJTMjU2In0.eyJyZWFsbWlkIjoiSUJNaWQiLCJzdWIiOiJyY
XZpIiwiaWF0IjoxNTM0MTUwOTU3LCJleHAiOjE1MzQyMzczNTcsImlzcyI6Imh0dHBzOi8vbG9jYWxob3N0OjEwNDQzL29pZGMvd
G9rZW4iLCJncmFudF90eXBlIjoidXJuOmlibTpwYXJhbXM6b2F1dGg6Z3JhbnQtdHlwZTphcGlrZXkiLCJzY29wZSI6Im9wZW5pZ
CIsImNsaWVudF9pZCI6ImRlZmF1bHQifQ.eQGgLevdKu7ZTpKKO4bJP2MHDJDTxNmpDNTYLYBepN-48xZmk0l-3CTd3YC71y9Ve-
UNUn5qL9m-B6G25sBc5KmyW2ZU7DS6MKeLrE9-bI-
HVp9GhNJ5DC06TSxCM6zcS26YqM1GXpqxOSTINgm7L9rKN10n9uYwszccha12bVR5ctLtrD8_5er1OaugKewgfXOWR8g1gwljwmm
s2FMnp72wE3DynwG16Bzon7U8FzC4YBmh4mFAHMc7C10553jLoOBMf7iq_3Yj4dNRegZhVpYQBC97A1vatODoNkrXbcuBtA9ujJ4
dzLj76Lam5c_iLTGbjTnwkgcALEUXjCv7kQ' 'https://<Cluster Master Host>:<Cluster Master API Port>/iam-
token/oidc/introspect'
```

A saída se assemelha ao código a seguir:

```
Resposta:
{"active":true,"iss":"https://<cluster_ca_domain>:10443/oidc/token","realmId":"IBMid","sub":"ravi","
account":
{},"scope":"openid","client_id":"default","iat":1531117013,"exp":1531203413,"grant_type":"urn:ibm:pa
rams:oauth:grant-type:apikey"}
```

APIs de conexão única

Use essas APIs para configurar a conexão única (SSO) no cluster do IBM® Cloud Private.

O administrador de cluster e o Administrador da equipe podem acessar todas as APIs de SSO.

Para usar essas APIs, deve-se incluir um cabeçalho de autorização em sua solicitação. É necessário um token de acesso para incluir no cabeçalho de autorização. Para obter o token de acesso, consulte Preparando para executar os comandos da API do componente ou de gerenciamento.

`<Cluster Master Host>:<Cluster Master API Port>` são usados para acessar as APIs. Os parâmetros são definidos nos Terminais mestres.

Ativar SAML

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/idmgmt/v1/saml/management`

Comando

`PUT`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -v -k -X PUT --header 'Authorization: Bearer $ACCESS_TOKEN' \
--header 'Content-Type: application/json' -d '{"enable": true}' https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/v1/saml/management
```

A resposta se assemelha ao código a seguir:

```
* Tentando 1.1.1.1 ...
* Conectado a 1.1.1.1 (1.1.1.1) porta 8443 (#0)
* localizados 148 certificados em /etc/ssl/certs/ca-certificates.crt * localizados 594 certificados
em /etc/ssl/certs * ALPN, oferta http/1.1 * conexão SSL usando TLS1.2 /
ECDHE_RSA_AES_256_GCM_SHA384.
.
* ALPN, server accepted to use http/1.1
> PUT /idmgmt/v1/saml/management HTTP/1.1
> Host: 1.1.1.1:8443
> User-Agent: curl/7.47.0
> Accept: */*
> Autorização: Bearer.
.
> Content-Type: application/json
> Content-Length: 16
>
* upload completely sent off: 16 out of 16 bytes
.
.
* Connection #0 to host 1.1.1.1 left intact
```

```
Response:
---------
Configuration successful

root@rise1: ~ #
```

Exportar metadados

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idauth/ibm/saml20/defaultSP/samlmetadata

Comando

GET

Formato de saída de comando

application/json

Nota: esta chamada API faz download de um arquivo de metadados do IBM Cloud Private.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -v -X GET --header 'Authorization: Bearer $ACCESS_TOKEN'\
https://<Cluster Master Host>:<Cluster Master API Port>/idauth/ibm/saml20/defaultSP/samlmetadata
```

A resposta de amostra é semelhante ao código a seguir:

```
* Tentando 2.2.2.2 ...
* Conectada a 2.2.2.2 (2.2.2.2) porta 8443 (#0)
* localizados 148 certificados em /etc/ssl/certs/ca-certificates.crt * localizados 594 certificados
em /etc/ssl/certs * ALPN, oferta http/1.1 * conexão SSL usando TLS1.2 /
ECDHE_RSA_AES_256_GCM_SHA384.
.
< HTTP/1.1 200 OK.
.
< X-XSS-Protection: 1; mode=block
<
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://2.2.2.2:8443/ibm/saml20/defaultSP"><md:SPSSODescriptor AuthnRequestsSigned="true"
WantAssertionsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>MIID9zCCAd8CCQCB/mzlEf0kfzANBgkqhkiG9w0BAQsFADBjMQswCQYDVQQGEwJVUzERMA8GA1UE
.
.
daqKQMlMr3xN9BAqmrUuFwKsrhz1uuJ/4v3iePDG5Qy4k4UVuOFiz1e5Tsakw72fGjk=</ds:X509Certificate>
</ds:X509Data>\
</ds:KeyInfo></md:KeyDescriptor><md:KeyDescriptor use="encryption"><ds:KeyInfo
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:X509Data>
<ds:X509Certificate>MIID9zCCAd8CCQCB/mzlEf0kfzANBgkqhkiG9w0BAQsFADBjMQswCQYDVQQGEwJVUzERMA8GA1UE
.
.
daqKQMlMr3xN9BAqmrUuFwKsrhz1uuJ/4v3iePDG5Qy4k4UVuOFiz1e5Tsakw72fGjk=</ds:X509Certificate>
```

```
</ds:X509Data>\
</ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://2.2.2.2:8443/ibm/saml20/defaultSP/slo"/><md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://2.2.2.2:8443/ibm/saml20/defaultSP/acs" index="0" isDefault="true"/>
</md:SPSSODescriptor></md:EntityDescriptor>root@rise1:~#
```

Importar metadados

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/v1/saml/upload

Comando

POST

Formato de saída de comando

application/json

Nota: essa chamada API é usada para fazer upload do arquivo de metadados recebido do servidor SAML corporativo. No exemplo, o nome do arquivo é `samlidp2_IBM_metadata_CIS_STAGE.xml`.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -v -k -X POST --header 'Authorization: Bearer $ACCESS_TOKEN' \
-F 'data=@samlidp2_IBM_metadata_CIS_STAGE.xml' https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/v1/saml/upload
```

A resposta de amostra é semelhante ao código a seguir:

```
* Tentando 1.1.1.1 ...
* TCP_NODELAY configurado.
.
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS header, Certificate Status (22):
} [ 5 bytes de dados ]
.
.
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* ALPN, server accepted to use http/1.1
* Server certificate:
.
.
> POST /idmgmt/v1/saml/upload HTTP/1.1
> Host: 1.1.1.1:8443
> User-Agent: curl/7.53.0
> Accept: */*
> Autorização: Bearer.
.
> Content-Length: 5313
```

```
> Expect: 100-continue
> Content-Type: multipart/form-data; boundary=------------------------9ecb5200d6d7e5e6
.
.
< HTTP/1.1 200 OK.
.
{ [31 bytes data]
100  5344  100    31  100  5313      1    275  0:00:31  0:00:19  0:00:12     0Metadata uploaded
successfully.
```

Verificar status de configuração de SSO

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/v1/saml/status

Comando

GET

Formato de saída de comando

application/json

Qualquer uma das respostas a seguir é válida:

- {"enable":true}
  O SAML está ativado e o arquivo de metadados foi transferido por upload com êxito.
- {"status":false,"description":["SAML Feature not enabled"]}
  SAML não está ativado.
- {"status":false,"description":["IDP Metadata not uploaded"]}
  O SAML está ativado, mas o arquivo de metadados recebido do servidor SAML corporativo não foi transferido por upload.
- {"status":false,"description":["SAML Feature not enabled", "IDP Metadata not uploaded"]}
  O SAML não está ativado ou não foi feito upload do arquivo de metadados recebido do servidor SAML corporativo.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -v -X GET --header 'Authorization: Bearer $ACCESS_TOKEN' https://<Cluster Master Host>:
<Cluster Master API Port>/idmgmt/v1/saml/status
```

A resposta de amostra é semelhante ao código a seguir:

```
* Tentando 2.2.2.2 ...
* Conectada a 2.2.2.2 (2.2.2.2) porta 8443 (#0)
* localizados 148 certificados em /etc/ssl/certs/ca-certificates.crt * localizados 594 certificados
em /etc/ssl/certs * ALPN, oferta http/1.1 * conexão SSL usando TLS1.2 /
ECDHE_RSA_AES_256_GCM_SHA384.
.
* ALPN, server accepted to use http/1.1
> GET /idmgmt/v1/saml/status HTTP/1.1
> Host: 2.2.2.2:8443
> User-Agent: curl/7.47.0
> Accept: */*
```

```
> Autorização: Bearer.
.
< HTTP/1.1 200 OK
< Server: openresty/1.11.2.4
< Date: Tue, 24 Jul 2018 08:43:00 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 16
< Connection: keep-alive
< Vary: Origin, Accept-Encoding
< Access-Control-Allow-Credentials: true
< X-XSS-Protection: 1; mode=block
< X-Frame-Options: SAMEORIGIN
< Strict-Transport-Security: max-age=15552000; includeSubDomains
< X-Download-Options: noopen
< X-Content-Type-Options: nosniff
< X-DNS-Prefetch-Control: off
< ETag: W/"10-H+tYev0PkZ4BjVuzAcmKAO7d1nc"
< X-Frame-Options: SAMEORIGIN
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
<
* Connection #0 to host 2.2.2.2 left intact

_Resposta: _

{"status":false,"description":["SAML Feature not enabled"]}
```

Desativar SAML

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/idmgmt/v1/saml/management

Comando

PUT

Formato de saída de comando

application/json

Essa API desativa o SAML e exclui o arquivo de metadados que foi enviado pelo servidor SAML corporativo.

O comando curl de amostra se assemelha ao código a seguir:

```
curl -v -k -X PUT --header 'Authorization: Bearer $ACCESS_TOKEN' --header 'Content-Type:
application/json' \
-d '{"enable": false}'  https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/v1/saml/management
```

A resposta de amostra é semelhante ao código a seguir:

```
* Tentando 1.1.1.1 ...
* Conectado a 1.1.1.1 (1.1.1.1) porta 8443 (#0)
* localizados 148 certificados em /etc/ssl/certs/ca-certificates.crt * localizados 594 certificados
em /etc/ssl/certs * ALPN, oferta http/1.1 * conexão SSL usando TLS1.2 /
ECDHE_RSA_AES_256_GCM_SHA384.
```

```
.
* ALPN, server accepted to use http/1.1
> PUT /idmgmt/v1/saml/management HTTP/1.1
> Host: 1.1.1.1:8443
> User-Agent: curl/7.47.0
> Accept: */*
> Autorização: Bearer.
.
> Content-Type: application/json
> Content-Length: 17
>
* upload completely sent off: 17 out of 17 bytes
< HTTP/1.1 200 OK
< Server: openresty/1.11.2.4
< Date: Sat, 04 Aug 2018 09:54:14 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 24
< Connection: keep-alive
< Vary: Origin, Accept-Encoding
< Access-Control-Allow-Credentials: true
< X-XSS-Protection: 1; mode=block
< X-Frame-Options: SAMEORIGIN
< Strict-Transport-Security: max-age=15552000; includeSubDomains
< X-Download-Options: noopen
< X-Content-Type-Options: nosniff
< X-DNS-Prefetch-Control: off
< ETag: W/"18-Akebq37pOvmw0KQ/7FxMJOPs24g"
< X-Frame-Options: SAMEORIGIN
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
<
* Connection #0 to host 1.1.1.1 left intact
Configuration successfulroot@rise1:~#
```

APIs de verificação de funcionamento e de versão do serviço

APIs para verificar o funcionamento do serviço e a versão da API.

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Serviço do gerenciador de identidade

APIs do serviço`icp-identity-manager`.

JSON swagger

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/idmgmt/explorer/swagger.json`

Comando

`GET`

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET "https://<Cluster Master Host>:<Cluster Master API
Port>/idmgmt/explorer/swagger.json"
```

A resposta se assemelha ao código a seguir:

```
Response:
{"swagger":"2.0","info":{"version":"1.0.0","title":"platform-identity-mgmt","description":"platform-
identity-mgmt description"},\
"basePath":"/idmgmt/identity/api/v1","paths":{"/directory/ldap/{id}":{"get":{"tags":
["Directory"],"summary"
...
...
{"type":"string","format":"date-time"},"expiry":{"type":"string","format":"date-time"}},"required":
["id"],\
"additionalProperties":false},"ObjectID":{"type":"string","pattern":"^[a-fA-F\\d]{24}$"}}}
```

Verificação da Versão

Versão da API

1.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/idmgmt/identity/api/v1/
```

Comando

```
GET
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET "https://<Cluster Master Host>:<Cluster Master API Port>/idmgmt/identity/api/v1/"  -I
```

A resposta se assemelha ao código a seguir:

```
Response:
HTTP/1.1 204 No Content
Server: openresty/1.13.6.2
Date: Fri, 05 Apr 2019 07:14:06 GMT
Connection: keep-alive
Vary: Origin
Access-Control-Allow-Credentials: true
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Download-Options: noopen
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
```

Serviço de administração de política do IAM

APIs do serviço `iam-policy-administration`.

JSON swagger

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/iam-pap/v1/api-docs`

Comando

`GET`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/v1/api-docs"
```

A resposta se assemelha ao código a seguir:

```
Resposta: {
  "swagger": "2.0",
  "info": {
  ...
  ...
}
```

Verificação de funcionamento e análise de prontidão

Versão da API

1.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/iam-pap/v1/health`

Comando

`GET`

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/v1/health"
```

A resposta se assemelha ao código a seguir:

```
Response:
OK
```

Verificação da Versão

Versão da API

1.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/iam-pap/v1
```

Comando

```
GET
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET "https://<Cluster Master Host>:<Cluster Master API Port>/iam-pap/v1"  -I
```

A resposta se assemelha ao código a seguir:

```
Response:
HTTP/1.1 204 No Content
Server: openresty/1.13.6.2
Date: Fri, 05 Apr 2019 07:24:53 GMT
Connection: keep-alive
transaction-id: e4574842-52e0-4734-a7e4-5e13c2099ad3
X-Response-Time: 0.708ms
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
```

Serviço de token do IAM

APIs do serviço`iam-token-service`.

Verificação da Versão

Versão da API

1.0.0

Componentes do URI da API

Esquema

HTTPS

IP do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho

/iam-token/v1

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -X GET "https://<Cluster Master Host>:<Cluster Master API Port>/iam-token/v1" -I
```

A resposta se assemelha ao código a seguir:

```
Response:
HTTP/1.1 204 No Content
Date: Wed, 08 May 2019 06:59:50 GMT
Content-Type: text/plain
Content-Length: 0
Connection: keep-alive
X-Powered-By: Servlet/3.1
Content-Language: en-US
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
```

API de gerenciamento de imagem

Os comandos da API de gerenciamento de imagem listam, atualizam e excluem imagens no registro de imagem.

O Administrador do cluster e o Administrador podem acessar todos os comandos das APIs de gerenciamento de imagem para todos os recursos.

Para usar essas APIs, deve-se incluir um cabeçalho de autorização em sua solicitação. Consulte Preparando para executar comandos da API do componente ou de gerenciamento.

Por exemplo:

```
curl --cacert /<certificate_path>/ca.crt -s -H "Authorization: Bearer $ID_TOKEN" https://<Cluster
Master Host>:<Cluster Master API Port>/image-manager/api/v1
```

Neste comando:

- `<certificate_path>` é `<installation_directory>/cluster/cfc-certs/router/icp-router.crt` em seu nó de inicialização
- `$ID_TOKEN` é a variável que armazena o token de identidade para seu cluster
- `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais

Se você não configurou um domínio de CA, será possível continuar a solicitação com uma conexão insegura.

Por exemplo:

```
curl -k -H "Authorization: Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster Master API
Port>/image-manager/api/v1
```

Dados da API de gerenciamento de imagem

Versão da API

3.2.0

Componentes do URI da API

Esquema

`HTTPS`

Nome do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho Base

`/image-manager/api/v1`

Formato de saída de comando

`application/json`

Excluir repositórios

Remover repositórios específicos do registro de imagem privado.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1/repositories/<repo>`

- Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.
- `repo` é o nome completo da imagem do repositório, incluindo o namespace. Por exemplo, `default/alpine`. O registro de imagem privado cria repositórios individuais para imagens que são designadas com o mesmo nome. Ao remover um repositório de imagem, você remove todas as imagens desse repositório. Não é possível remover uma imagem específica no repositório.

  Por exemplo, `mycluster.icp:8500/default/nginx:1.9.1` é uma imagem no repositório `mycluster.icp:8500/default/nginx`.

Depois de remover o repositório, é possível remover os arquivos relacionados do armazenamento de registro. Veja Removendo arquivos de imagem do armazenamento de registro privado.

Versão da API

3.2.0

Componentes do URI da API

Esquema

`HTTPS`

Nome do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho Base

`/image-manager/api/v1/repositories/{namespace}/{repo}`

Comando

`DELETE`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X DELETE -H "Authorization:Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1/repositories/<repo> --insecure
```

O comando não retorna nenhuma mensagem.

```
root@master:~# curl -X DELETE -H "Authorization: Bearer $ID_TOKEN" https://mycluster.icp:8443/image-manager/api/v1/repositories/default/alpine --insecure
```

Parâmetros

Tabela 1. Os parâmetros que você usa para excluir repositórios

Tipo	Nome	Descrição	Esquema	Padrão	Obrigatório
Caminho	Namespace	Namespace no qual o repositório está	sequência		SIM
Caminho	repositório:.*	Nome do repositório	sequência		SIM

Respostas

Tabela 2. As respostas que são retornadas quando você exclui os repositórios

Código HTTP	Descrição	Esquema
200	OK	Sem conteúdo

Obter tokens JWT

Obter o token JWT que deve ser usado com o cliente Docker.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1/auth/token`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Versão da API

3.2.0

Componentes do URI da API

Esquema

`HTTPS`

Nome do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho Base

`/image-manager/api/v1/auth/token`

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -H GET --header "Authorization: Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster
Master API Port>/image-manager/api/v1/auth/token
```

A resposta se assemelha ao código a seguir:

```
root@master:~# curl -k -H GET --header "Authorization: Bearer $ID_TOKEN"
https://mycluster.icp:8443/image-manager/api/v1/auth/token
{"expires_in":1800,"issued_at":"2017-11-30T05:24:21Z","token":\
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjZMUDY6SFFJUjozVkg0OlZIUUk6TTRNMjpMSEFWOldGS0M6Mk9aQTp
STjRMOlQ3UUs6RzVLUTpKNkg1In0.eyJpc3MiOiJyZWdpc3RyeS10b2tlbi1pc3N1ZXIiLCJzdWIiOiIiLCJhdWQiOiIiLCJleHA
iOjE1MTIwMjEyNjEsIm5iZiI6MTUxMjAxOTQ2MSwiaWF0IjoxNTEyMDE5NDYxLCJqdGkiOiJsYzVkRzVPVTI0anhjemVLIiwiYWN
jZXNzIjpudWxsfQ.F2D6VjwhQRlRvU-
da2zHDzDbck8avh1lxc8B4hAVrIhEY8orcqwYhocjxnFk3kXdNm3yihMwi17lYySEnet3_p7jWOJ0XGTF6_m7DeUsNd-
YqtGGv7FTycTG_10Xnm7zuknBhpsbMx_Eq4gKRBM1ndwdkhhOLG135r97BbTcT_GlHcnmyKfFLXfmhVgiAhBeTn1_phmSO-
Olys0bhbKl2M_jIiCGLwleKQpa3dFsJ3JsYGSQcB7dEVZuSAmzc3OmlYXmvl8oCnjyM-RIkQk8-
uGkI_7cjhXOY8rORGMgp0r6f8LXky5K_XH1mIG7BARKr1E-9SKyyIGwRWMUIs3fwA"}
```

Respostas

Tabela 1. As respostas que são retornadas quando você obtém tokens JWT

Código HTTP	Descrição	Esquema
200	OK	Sem conteúdo

Listar terminais

Listar terminais disponíveis.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Versão da API

3.2.0

Componentes do URI da API

Esquema

HTTPS

Nome do Host

Host Principal do Cluster

Número da porta

Porta da API Principal do Cluster

Caminho Base

/image-manager/api/v1

Comando

GET

Formato de saída de comando

application/json

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -H GET --header "Authorization: Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster
Master API Port>/image-manager/api/v1
```

A resposta se assemelha ao código a seguir:

```
root@master:~# curl -k -X GET --header "Authorization: Bearer $ID_TOKEN"
https://mycluster.icp:8443/image-manager/api/v1
IBM Cloud Private Image Management API.
The listener is working. Now try a valid endpoint
Available APIs:
GET /
GET /image-manager/api/v1
GET /image-manager/api/v1/auth/token
GET /image-manager/api/v1/repositories
GET /image-manager/api/v1/repositories/{repo:.*}
PUT /image-manager/api/v1/repositories/{repo:.*}
DELETE /image-manager/api/v1/repositories/{repo:.*}
```

Respostas

Tabela 1. As respostas que são retornadas quando você lista terminais

Código HTTP	Descrição	Esquema
200	OK	Sem conteúdo

Listar repositórios

Listar todos os repositórios no registro de imagem privado.

O registro de imagem privado cria repositórios individuais para imagens que são designadas com o mesmo nome.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1/repositories`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Versão da API

3.2.0

Componentes do URI da API

Esquema

`HTTPS`

Nome do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho Base

`/image-manager/api/v1/repositories`

Comando

`GET`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -H GET --header "Authorization: Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster
Master API Port>/image-manager/api/v1/repositories --insecure
```

A resposta de amostra é semelhante ao código a seguir:

```
root@master:~# curl -k -H GET --header "Authorization: Bearer $ID_TOKEN"
https://mycluster.icp:8443/image-manager/api/v1/repositories
{"repositories":[{"name":"default/alpine","owner":"default","scope":"global","tags":\
[{"name":"3.6","manifest":{"schemaVersion":2,"mediaType":"application/
vnd.docker.distribution.manifest.v2+json",\
"config":{"mediaType":"application/vnd.docker.container.image.v1\
+json","size":1512,"digest":\
"sha256:053cde6e8953ebd834df8f6382e68be83adb39bfc063e40b0fc61b4b333938f1"},"layers":
[{"mediaType":"application/\
vnd.docker.image.rootfs.diff.tar.gzip","size":1991435,"digest":\
"sha256:b56ae66c29370df48e7377c8f9baa744a3958058a766793f821dadcb144a4647"}]}}]},
{"name":"default/alpine-ppc64le","owner":"default","scope":"namespace","tags":
[{"name":"3.6","manifest":\
{"schemaVersion":2,"mediaType":"application/vnd.docker.distribution.manifest.v2+json",\
"config":{"mediaType":"application/vnd.docker.container.image.v1
+json","size":1760,"digest":\
"sha256:daa414b19dcffa0ba7b80abb50a6b31156f1efd7bb1b8cdaeeb848da367afa38"},"layers":
[{"mediaType":"application/\
vnd.docker.image.rootfs.diff.tar.gzip","size":2008578,\
"digest":"sha256:1e52418956f7d2a8ea35e8e6e3318fd08e005b27457d77868c225e7433bbfa02"},
{"mediaType":"application/\
vnd.docker.image.rootfs.diff.tar.gzip","size":176,\
"digest":"sha256:acf472f4e5bb7956ac20bb343b304e1d3de1f79160c0d158cccbe25980022d50"}]}]},
{"name":"default/alpine-s390x","owner":"default","scope":"namespace","tags":
[{"name":"3.6","manifest":
{"schemaVersion":2,"mediaType":"application/vnd.docker.distribution.manifest.v2+json","config":
{"mediaType":\
"application/vnd.docker.container.image.v1
+json","size":1758,"digest":\
"sha256:f7d8089567d12668e7f0d27e18b87abb38013f1b221b37ff7158fce2cbc3d792"},"layers":
[{"mediaType":"application/\
vnd.docker.image.rootfs.diff.tar.gzip","size":2109387,\
"digest":"sha256:22a16c518b849c1025d3475ee9310b25eaf5ec70f10d657ac7a322323f14873a"},
{"mediaType":"application/vnd.docker.image.rootfs.diff.tar.gzip","size":176,\
"digest":"sha256:0e5978b6b34b3e943e0fd25dfb50991c0bad82a986cfdaa91c4de756431ba679"}]}}]}]}root@maste
r:~#
```

Respostas

Tabela 1. As respostas que são retornadas quando você lista todos os repositórios

Código HTTP	Descrição	Esquema
200	OK	Sem conteúdo

Listar repositório especificado

Obter detalhes sobre um repositório específico.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1/repositories/<repo>`

Em que `repo` é o nome de uma imagem, incluindo tags. Por exemplo, se você enviar por push a imagem `mycluster.icp:8500/default/tomcat` para o registro de imagem, então, `default/tomcat` será o nome do repositório. Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Versão da API

3.2.0

Componentes do URI da API

Esquema

`HTTPS`

Nome do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho Base

`/image-manager/api/v1/repositories/{namespace}/{repo}`

Comando

`GET`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -H GET --header "Authorization: Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster
Master API Port>/image-manager/api/v1/repositories/<repo> --insecure
```

A resposta se assemelha ao código a seguir:

```
root@master:~# curl -k -H GET --header "Authorization: Bearer $ID_TOKEN"
https://mycluster.icp:8443/image-manager/api/v1/repositories/default/alpine --insecure

{"name":"default/alpine","owner":"default","scope":"global","tags":\
[{"name":"3.6","manifest":{"schemaVersion":2,"mediaType":"application/\
vnd.docker.distribution.manifest.v2+json","config":{"mediaType":"application/\
vnd.docker.container.image.v1+json","size":1512,"digest":\
"sha256:053cde6e8953ebd834df8f6382e68be83adb39bfc063e40b0fc61b4b333938f1"},"layers":
[{"mediaType":"application/vnd.docker.image.rootfs.diff.tar.gzip","size":1991435,\
"digest":"sha256:b56ae66c29370df48e7377c8f9baa744a3958058a766793f821dadcb144a4647"}]}}]}root@master:
~#
```

Parâmetros

Tabela 1. Os parâmetros que você usa para listar repositórios específicos

Tipo	Nome	Descrição	Esquema	Padrão	Obrigatório
Caminho	namespace	Namespace no qual o repositório está	sequência		SIM
Conteúdo	repositório:.*	Nome do repositório	sequência		SIM

Respostas

Tabela 2. As respostas que são retornadas quando você lista repositórios específicos

Código HTTP	Descrição	Esquema
200	Criar sucesso	Sem conteúdo

Atualizar metadados do repositório

Atualizar os metadados de um repositório.

Caminho base: `https://<Cluster Master Host>:<Cluster Master API Port>/image-manager/api/v1/repositories/<repo>`

Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Em que `repo` é o nome de uma imagem, incluindo tags. Por exemplo, se você enviar por push a imagem `mycluster.icp:8500/default/tomcat` para o registro de imagem, então, `default/tomcat` será o nome do repositório.

Versão da API

3.2.0

Componentes do URI da API

Esquema

```
HTTPS
```

Nome do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho Base

```
/image-manager/api/v1/repositories/{namespace}/{repo}
```

Comando

```
PUT
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X PUT -d '{"scope": "global"}' -H "Authorization: Bearer $ID_TOKEN" https://<Cluster Master
Host>:<Cluster Master API Port>/image-manager/api/v1/repositories/<repo> --insecure
```

A resposta se assemelha ao código a seguir:

```
curl -X PUT -d '{"scope": "global"}' -H "Authorization:Bearer $ID_TOKEN"
https://mycluster.icp:8443/image-manager/api/v1/repositories/default/cam-broker --insecure

OK
```

Parâmetros

Tabela 1. Os parâmetros usados para atualizar um repositório

Tipo	Nome	Descrição	Esquema	Padrão	Obrigatório
Conteúdo	"escopo":	Atualize a imagem do repositório em um nível global ou de namespace. Valores: "global" ou "namespace"	sequência		SIM
Conteúdo	namespace	Namespace no qual o repositório está	sequência		SIM
Conteúdo	repositório:*	Nome do repositório	sequência		SIM

Respostas

Tabela 2. As respostas que são retornadas quando você atualiza metadados de um repositório

Código HTTP	Descrição	Esquema
200	OK	Sem conteúdo

API do Consultor de Vulnerabilidade

Os comandos da API do Vulnerability Advisor podem ser usados para gerenciar relatórios de segurança.

O Administrador de Cluster e o Administrador podem acessar todas as APIs do Vulnerability Advisor para todos os recursos.

Também é possível acessar os docs da API swagger para os componentes do Vulnerability Advisor a partir do `https://<Cluster Master Host>:<Cluster Master API Port>/va/ui/api-docs/index.html` da console de gerenciamento do IBM Cloud Private. Os parâmetros `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Para acessar essas APIs na linha de comandos, deve-se incluir um cabeçalho de autorização em sua solicitação. Consulte Preparando para executar comandos da API do componente ou de gerenciamento.

Por exemplo:

```
curl -k -s -XGET -H  "Authorization: Bearer $ID_TOKEN" https://<Cluster Master Host>:<Cluster Master API Port>/va/api
```

O parâmetro `$ID_TOKEN` é a variável que armazena o token de identidade para seu cluster e `<Cluster Master Host>` e `<Cluster Master API Port>` estão definidos em Terminais principais.

Consultor de API de dados

Versão da API

3.2.0

Componentes do URI da API

Esquema

`HTTPS`

Nome do Host

`<Host Mestre do Cluster>`

Número da porta

`<Cluster Master API Port>`

Caminho Base

`/va/api/v1`

Formato de saída de comando

`application/json`

Obter namespaces

O comando curl de amostra assemelha-se ao seguinte comando:

```
curl -k -s -XGET -H "Authorization: Bearer $ID_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API Port>/va/api/get-namespaces?access_group=kube-system&max=3' | jq .
```

A resposta é semelhante ao comando a seguir:

```
{
  "result": [
    {
      "namespace": "kube-system/va-annotator-74c4c9bb96-8pbg9/va-
annotator/e84dbbbca7af7c6ca3555620ca4ffc80d23c668221321098968cc759741ebaea",
      "source_type": "container",
      "timestamp": "2018-04-27T10:59:59+0000"
    },
    {
      "namespace": "kube-system/filebeat-ds-amd64-
9s4zv/POD/6bcc7bb988f449586eaa5c45289e1f1f67af2b2ad68f4c2bc60e1944aca93e47",
      "source_type": "container",
      "timestamp": "2018-04-27T06:53:09+0000"
   }
}
```

Obtenha uma captura instantânea dos registros de data e hora

O comando curl de amostra assemelha-se ao seguinte comando:

```
curl -k -s -XGET -H "Authorization: Bearer $ID_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API Port>/va/api/get-snapshot-timestamps?access_group=kube-system&namespace=kube-system/elasticsearch-client-6c9fc8b5b6-dvztg/POD/d228119a35ab6ff158b3903cf24b3c014ad3809748485b1c6008d00baf1d6487' | jq ..
```

A resposta é semelhante ao conteúdo a seguir:

```
{
  "result": [
    "2018-04-27T06:52:23+0000",
    "2018-04-26T06:52:32+0000",
    "2018-04-25T06:54:12+0000",
    "2018-04-24T06:53:22+0000",
  ],
  "request_id": "66c4e381-0cc9-4f63-8ca5-aac81400ac8d"
}
```

Obtenha relatório

O comando curl de amostra assemelha-se ao seguinte comando:

```
curl -k -s -XGET -H "Authorization: Bearer $ID_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API Port>/va/api/get-report?access_group=kube-system&namespace=kube-system/elasticsearch-client-6c9fc8b5b6-dvztg/POD/d228119a35ab6ff158b3903cf24b3c014ad3809748485b1c6008d00baf1d6487&timestamp=2018-04-10T06%3A53%3A27%2B0000&report_type=compliance' | jq .
```

A resposta se assemelha ao código a seguir:

```
{
  "result": {
    "Conformidade": {
      "statusCode": 200,
      "body": {
        "total": 27,
        "compliant": 23,
        "non_compliant": 4,
        "custom": 0,
        "crawled_time": "2018-04-10T06:53:27+0000",
        "details": [
          {
            "compliance_id": "Linux.1-1-a",
            "compliant": "false",
            "compliance_check_time": "2018-04-10T06:53:30.639238Z",
            "reason": "File /etc/passwd not found",
            "description": "Each UID must be used only once.",
            "rule_type": "default"
          },
          {
          ..............
```

Obtenha uma avaliação

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -s -XGET -H "Authorization: Bearer $ID_TOKEN" 'https://<Cluster Master Host>:<Cluster Master API Port>/va/api/get-verdict?access_group=kube-system&namespace=kube-system%2Fk8s-proxy-10.91.0.130%2Fproxy%2Fa7a93bba0f57de8055b3b5c880c340501a5d2158fa36326fbc76392e243b55f4&policy_name=package_vulnerability_found&source_type=container' | jq .
```

A resposta se assemelha ao código a seguir:

```
{
  "result": {
    "status": "violation",
    "detail":
    {
      .....
    }
  },
  "request_id": "30252cbf-5519-4bd3-837e-d0549795eef3"
}
```

Obtenha recursos

O comando curl de amostra se assemelha ao código a seguir:

```
curl -k -s -XGET -H "Authorization: Bearer $ID_TOKEN" 'https://<Cluster Master Host>:<Cluster Master
API Port>/va/api/get-features?access_group=kube-system&namespace=kube-system%2Fk8s-proxy-
10.91.0.130%2Fproxy%2Fa7a93bba0f57de8055b3b5c880c340501a5d2158fa36326fbc76392e243b55f4&source_type=c
ontainer&timestamp=2018-11-11T03%3A56%3A26%2B0000&data_type=config' | jq .
```

A resposta se assemelha ao código a seguir:

```
{
    "result": {
        "total": 9,
        "count": 9,
        "crawled_time": "2018-11-11T03:56:26+0000",
        "values": [
        {
            "key": "/etc/hostname",
            "value":
            {
                "name": "hostname",
                "content": "ra3icp2\n",
                "path": "/etc/hostname"
            }
        },
        ....
        }]
    },
    "request_id": "889032f0-65aa-462f-b174-2d82f7a6bbf9"
}
```

APIs do Key Management Service

Use essas APIs para gerenciar chaves para o Key Management Service (KMS).

O acesso aos terminais de API é controlado pelo nível de acesso do ID do serviço que faz a chamada.

Para usar essas APIs, deve-se ter um ID de serviço e também incluir um cabeçalho de autorização em sua solicitação. É necessário incluir um token de acesso OIDC no cabeçalho de autorização. Para obter o token de acesso, consulte Gerar um token do OpenID Connect (OIDC).

`<Cluster Master Host>:<Cluster Master API Port>` são usados para acessar as APIs. Os parâmetros são definidos nos Terminais mestres.

Gerar uma chave

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor

Versão da API

2.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/kms/api/v2/keys`

Comando

```
POST
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -H 'correlation-id: <correlation_ID>' \
   -d '{
  "metadata": {
    "collectionType": "application/vnd.ibm.kms.key+json", "collectionTotal": 1
  }, "resources": [ {
    "type": "application/vnd.ibm.kms.key+json",
    "name": "<key_alias>",
    "description": "<key_description>",
    "expirationDate": "<YYYY-MM-DDTHH:MM:SS.SSZ>",
    "extractable": <key_type>
    }
  ]
 }'
```

Variável	Descrição
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.
correlation_ID	O identificador exclusivo que é usado para rastrear e correlacionar transações.
key_alias	Um nome exclusivo, legível, para facilitar a identificação da sua chave. **Importante:** Para proteger sua privacidade, não armazene seus dados pessoais como metadados para sua chave.
key_description	**Opcional:** uma descrição estendida de sua chave. **Importante:** Para proteger sua privacidade, não armazene seus dados pessoais como metadados para sua chave.
YYYY-MM-DD HH:MM:SS.SS	**Opcional:** a data e hora em que a chave expira no sistema, no

formato RFC 3339. Se você não especificar o atributo `expirationDate`, a chave não expirará. | | key_type | Um valor booleano que determina se o material chave pode deixar o serviço. Ao configurar o atributo `extractable` como `false`, o serviço cria uma chave raiz que pode ser usada para operações `wrap` ou `unwrap`. Configure o atributo `extractable` como `true` para gerar uma chave padrão.|

Uma resposta bem-sucedida retorna o valor de ID da sua chave, juntamente com outros metadados. O ID é um identificador exclusivo que é designado à sua chave e é usado para chamadas subsequentes à API do Key Protect.

Importar uma chave

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor

Versão da API

2.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/kms/api/v2/keys`

Comando

`POST`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST \
  https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys \
  -H 'authorization: Bearer $ACCESS_TOKEN' \
  -H 'icp-instance: <instance_ID>' \
  -H 'content-type: application/vnd.ibm.kms.key+json' \
  -H 'correlation-id: <correlation_ID>' \
  -d '{
 "metadata": {
   "collectionType": "application/vnd.ibm.kms.key+json", "collectionTotal": 1
 }, "resources": [ {
   "type": "application/vnd.ibm.kms.key+json",
   "name": "<key_alias>",
   "description": "<key_description>",
   "expirationDate": "<YYYY-MM-DDTHH:MM:SS.SSZ>",
   "payload": "<key_material>",
   "extractable": <key_type>
   }
 ]
}'
```

Variável	Descrição
IAM_token	O token de acesso do IBM Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.
correlation_ID	O identificador exclusivo que é usado para rastrear e correlacionar transações.
key_alias	Um nome exclusivo, legível, para facilitar a identificação da sua chave. **Importante:** Para proteger sua privacidade, não armazene seus dados pessoais como metadados para sua chave.
key_description	**Opcional:** uma descrição estendida de sua chave. **Importante:** Para proteger sua privacidade, não armazene seus dados pessoais como metadados para sua chave.
YYYY-MM-DD HH:MM:SS.SS	**Opcional:** a data e hora em que a chave expira no sistema, no

formato RFC 3339. Se você não especificar o atributo `expirationDate`, a chave não expirará. | | key_material | O material de chave codificado em base64, como uma chave de quebra de chave existente, que você deseja armazenar e gerenciar no serviço.

Assegure-se de que o material chave atenda aos seguintes requisitos:

1 . A chave deve ser 256, 384 ou 512 bits.
2. Os bytes de dados, por exemplo, 32 bytes para 256 bits, devem ser codificados usando a codificação base64. | | key_type | Um valor booleano que determina se o material chave pode deixar o serviço. Ao configurar o atributo `extractable` como `false`, o serviço importa uma chave raiz que pode ser usada para operações `wrap` ou `unwrap`. Configure o atributo `extractable` como `true` para importar uma chave padrão.|

Uma resposta bem-sucedida retorna o valor de ID da sua chave, juntamente com outros metadados. O ID é um identificador exclusivo que é designado à sua chave e é usado para chamadas subsequentes à API do Key Protect.

Recuperar uma lista de chaves

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor

Versão da API
2.0.0
Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/kms/api/v2/keys
```

Comando

```
GET
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.collection+json' \
```

Variável	Descrição
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.

Uma resposta bem-sucedida retorna o número de chaves e os nomes de chaves. Ele não retorna o material chave.

Recuperar uma série de chaves

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor

Versão da API
2.0.0
Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/kms/api/v2/keys
```

Comando

```
HEAD
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X HEAD \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.collection+json' \
```

Variável	Descrição
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.

Uma resposta bem-sucedida retorna o número de chaves.

Recuperar uma chave por ID

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor

Versão da API
2.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/kms/api/v2/keys/{ID}
```

Comando

```
GET
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X GET \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID} \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.collection+json' \
```

Variável	Descrição
ID	O UUID v4 da chave.
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.

Uma resposta bem-sucedida retorna os detalhes da chave solicitada. Ele não retorna o material chave.

Agrupar uma chave

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor ou visualizador

Versão da API

2.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/kms/api/v2/keys/{ID}
```

Comando

```
POST
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID}?action=wrap \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.kms.key_action+json' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -d '{
     'plaintext': '<data_key>',
     'aad': ['<additional_data>', '<additional_data>']
   }'
```

Variável	Descrição
ID	A chave raiz que é usada como a chave de quebra. Deve ser um UUID v4 para uma chave ativa.
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.
correlation_ID	O identificador exclusivo que é usado para rastrear e correlacionar transações.
data_key	A chave de criptografia de dados (DEK). Forneça um texto simples codificado em base64 durante uma

ação de agrupamento. Para gerar um novo DEK, omita a propriedade de texto sem formatação. O KMS gera um texto sem formatação aleatório de 32 bytes que é enraizado em um dispositivo HSM e, em seguida, agrupa esse valor. O comprimento da chave deve ser menor que ou igual a 4.096 bytes. | | additional_data | Os dados de autenticação adicionais (AAD) que são usados para proteger a chave. Se você usar o AAD quando fizer uma chamada de diagnóstico, deverá usar o mesmo AAD durante uma chamada de desagrupamento. É possível especificar até 126 AADs. O comprimento do AAD deve estar no intervalo de 0 a 255.|

Desagrupar uma chave

Tipo de usuário ou nível de acesso requerido: administrador do cluster, administrador, editor ou visualizador

Versão da API

2.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

`/kms/api/v2/keys/{ID}`

Comando

`POST`

Formato de saída de comando

`application/json`

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID}?action=unwrap \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.kms.key_action+json' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -d '{
     'ciphertext': '<data_key>',
     'aad': ['<additional_data>', '<additional_data>']
   }'
```

Variável	Descrição
ID	A chave raiz que é usada como a chave de quebra. Deve ser um UUID v4 para uma chave ativa.
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.
correlation_ID	O identificador exclusivo que é usado para rastrear e correlacionar transações.
data_key	O DEK agrupado (WDEK) que é usado em ações `unwrap`. Forneça um texto cifrado

codificado em base64 durante uma ação de desagrupamento. A resposta é um texto sem formatação com codificação base64 no corpo da resposta. | | additional_data | O AAD que é usado para proteger a chave. Se você usou AADs ao fazer uma chamada de agrupamento, deve-se usar os mesmos AADs durante a chamada de desagrupamento.|

Girar uma chave

Tipo de usuário ou nível de acesso necessários: administrador de cluster ou administrador

Versão da API

2.0.0

Componentes do URI da API

Esquema

`HTTPS`

IP do Host

`Host Principal do Cluster`

Número da porta

`Porta da API Principal do Cluster`

Caminho

```
/kms/api/v2/keys/{ID}
```

Comando

```
POST
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X POST \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID}?action=rotate \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.kms.key_action+json' \
   -H 'content-type: application/vnd.ibm.kms.key+json' \
   -d '{
     'payload': '< data_key>'
   }'
```

Variável	Descrição
ID	A chave raiz que é usada como a chave de quebra. Deve ser um UUID v4 para uma chave ativa.
ACCESS_TOKEN	O token de acesso do IBM Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.
correlation_ID	O identificador exclusivo que é usado para rastrear e correlacionar transações.
data_key	O material de chave codificado em base64, como uma chave de quebra de chave existente, que você deseja armazenar e gerenciar no serviço. Para girar uma chave que foi gerada inicialmente pelo KMS, omita o atributo de carga útil e transmita um corpo de entidade de solicitação vazio. Para girar uma chave importada, forneça um material de chave que atenda aos seguintes requisitos: 1. A chave deve ser 256, 384 ou 512 bits. Os bytes de dados, por exemplo, 32 bytes para 256 bits, devem ser codificados usando a codificação base64.

Excluir uma chave por ID

Tipo de usuário ou nível de acesso necessários: administrador de cluster ou administrador

Versão da API

2.0.0

Componentes do URI da API

Esquema

```
HTTPS
```

IP do Host

```
Host Principal do Cluster
```

Número da porta

```
Porta da API Principal do Cluster
```

Caminho

```
/kms/api/v2/keys/{ID}
```

Comando

```
EXCLUIR
```

Formato de saída de comando

```
application/json
```

O comando curl de amostra se assemelha ao código a seguir:

```
curl -X DELETE \
   https://<Cluster Master Host>:<Cluster Master API Port>/kms/api/v2/keys/{ID} \
   -H 'authorization: Bearer $ACCESS_TOKEN' \
   -H 'icp-instance: <instance_ID>' \
   -H 'accept: application/vnd.ibm.collection+json' \
```

Variável	Descrição
ID	O UUID v4 da chave.
ACCESS_TOKEN	O token de acesso do IBM® Cloud Private. Inclua o conteúdo completo do token de acesso, incluindo o valor do portador, na solicitação Curl.
instance_ID	O identificador exclusivo que é designado à sua instância do KMS.

Ao excluir uma chave, o conteúdo da chave e os dados associados são removidos permanentemente. Não é possível reverter a ação.

Glossário

Última atualização: 27 de maio de 2019

Este glossário fornece termos e definições para o IBM® Cloud Private.

As referências cruzadas a seguir são utilizadas neste glossário:

- *Consulte* o encaminha de um termo não preferencial para um termo preferencial ou de uma abreviação para a sua forma por extenso.
- *Consulte também* o encaminha para um termo relacionado ou contrastante.

Para obter outros termos e definições, consulte o Website de terminologia da IBM.

A B C D E F G H I K L M N O P R S T V W

A

alta disponibilidade (HA)

A habilidade de serviços de TI para suportar todas as indisponibilidades e continuar a fornecer recursos de processamento, de acordo com algum nível de serviço predefinido. As interrupções cobertas incluem eventos planejados, como manutenção e backups, e eventos não planejados, como falhas de software, falhas de hardware, quedas de energia e desastres. Veja também tolerância a falhas.

ambiente de airgap

Um ambiente de rede que não tem acesso à Internet.

Aplicativo

Um ou mais programas de computador ou componentes de software que fornecem funcionalidade no suporte direto de um processo ou processos de negócios específicos.

applog

Consulte log do aplicativo.

B

broker de serviço

Um componente de um serviço que implementa um ampère de ofertas e planos de serviços, e interpreta chamadas para provisionamento e desprovisionamento, vinculação e desvinculação.

buildpack

Uma coleção de scripts que fornecem suporte de estrutura e tempo de execução para aplicativos.

C

camada

Uma versão mudada de uma imagem pai. As imagens consistem em camadas, em que a versão é disposta em camadas na parte superior da imagem pai para criar a nova imagem. Consulte também contêiner, imagem.

catálogo

Um local centralizado que pode ser usado para procurar e instalar pacotes em um cluster.

Célula Diego

Uma instância da máquina virtual no Cloud Foundry.

Chave API

Um código exclusivo que é transmitido para uma API para identificar o aplicativo de chamada ou o usuário. Uma chave de API é usada para rastrear e controlar como a API está sendo usada, por exemplo, para evitar o uso malicioso ou abuso da API.

Cloud Pak

Um software conteinerizado de classificação corporativa que é construído com padrões abertos e integrado com serviços de plataforma para operações de gerenciamento e de ciclo de vida.

cluster

Um conjunto de recursos, nós do trabalhador, redes e dispositivos de armazenamento que mantêm os aplicativos altamente disponíveis e prontos para implementação em contêineres.

conjunto

Um grupo de contêineres que estão em execução em um cluster do Kubernetes. Um pod é uma unidade de trabalho executável, que pode ser um aplicativo independente ou um microsserviço.

console de gerenciamento

A interface gráfica com o usuário para o IBM Cloud Private.

Container

Uma construção do sistema que permite aos usuários executar instâncias do sistema operacional lógico separadas simultaneamente. Os contêineres usam camadas de sistemas de arquivos para minimizar os tamanhos de imagem e promover a reutilização. Veja também imagem, camada, registro.

contêiner de concepção

Consulte contêiner do instalador.

contêiner do instalador

O contêiner do Docker que executa o gerenciador de configuração e a ferramenta de implementação do Cloud Foundry.

controle de acesso baseado na função (RBAC)

O processo de restringir os componentes integrais de um sistema baseado na autenticação do usuário, funções e permissões.

corrupção

Marcar uma determinada entrada, como uma variável, como sendo insegura para submetê-la à verificação de segurança.

D

deployment

Um processo que recupera a saída de uma construção, compacta a saída com propriedades de configuração e instala o pacote em um local predefinido para que ele possa ser testado ou executado.

DevOps

Uma metodologia de software que integra desenvolvimento de aplicativo e operações de TI para que as equipes possam entregar código mais rápido para produção e iterar continuamente com base no feedback do mercado.

Diego

A arquitetura que é usada no Cloud Foundry para gerenciar contêineres de app.

Docker

Uma plataforma aberta que os desenvolvedores e administradores de sistema podem usar para construir, enviar e executar aplicativos distribuídos.

E

equilibrador de carga

Um software ou hardware que distribui a carga de trabalho em um conjunto de servidores para garantir que estes não sejam sobrecarregados. O balanceador de carga também direciona os usuários a outro servidor se o servidor inicial falhar.

extension

Um pacote que contém um processo de implementação e seus scripts e arquivos necessários.

F

Ferramenta de implementação do Cloud Foundry

A interface com o usuário que é usada para gerenciar a implementação do Cloud Foundry.

G

gerenciador de imagem

Um local centralizado para gerenciar imagens dentro de um cluster.

Grafana

Uma plataforma de análise de dados e de visualização de software livre para monitorar, procurar, analisar e visualizar métricas.

Gráfico do Helm

Um pacote do Helm que contém informações para instalar um conjunto de recursos do Kubernetes em um cluster do Kubernetes.

H

HA

Veja alta disponibilidade.

I

imagem

Um sistema de arquivos e seus parâmetros de execução usados dentro de um tempo de execução do contêiner para criar um contêiner. O sistema de arquivos consiste em uma série de camadas, combinadas no tempo de execução, que são criadas à medida que a imagem é construída por atualizações sucessivas. A imagem não retém o estado à medida que o contêiner é executado. Consulte também contêiner, camada, registro.

Imagem de contêiner

No Docker, software independente, executável, incluindo ferramentas de código e do sistema, que pode ser usado para executar um aplicativo.

ingress

Uma coleção de regras para permitir conexões de entrada com os serviços de cluster do Kubernetes.

isolamento

O processo de confinamento de implementações de carga de trabalho para recursos virtuais e físicos dedicados para obter suporte de ocupação variada.

Istio

Tecnologia aberta que permite que os desenvolvedores conectem, gerenciem e protejam redes de diferentes microsserviços ininterruptamente, não importa a plataforma, a origem ou o fornecedor.

K

Klusterlet

No IBM Multicloud Manager, o agente que é responsável por um único cluster do Kubernetes.

Kubernetes

Uma ferramenta de orquestração de software livre para contêineres.

L

Liberação do Helm

Uma instância de um gráfico do Helm que é executado em um cluster do Kubernetes.

log de auditoria

Um arquivo de log que contém um registro de eventos e respostas do sistema.

log do aplicativo (applog)

Um log que é produzido a partir de aplicativos que são implementados no ambiente do Cloud Foundry.

log do sistema (syslog)

Um log que é produzido por componentes do Cloud Foundry.

M

malha

Uma topologia de rede na qual os dispositivos são conectados com muitas interconexões redundantes entre nós de rede. Cada nó tem uma conexão com todos os outros nós na rede.

malha de serviço

No Istio, uma camada de infraestrutura que permite interação e comunicação de microsserviços.

marketplace

Uma lista de serviços ativados a partir dos quais os usuários podem provisionar recursos.

microclimate

Uma solução de ponta a ponta, nativa de nuvem para criação, construção, teste e implementação de aplicativos.

microsserviço

Um conjunto de componentes arquitetônicos pequeno e independente, cada um com um único propósito, que comunica-se por uma API leve comum.

Minio

Um servidor de armazenamento de objetos compatível com o Amazon S3 que pode ser usado para armazenar dados não estruturados, como fotos, vídeos, arquivos de log, backups, VMs e imagens de contêiner.

MT

Consulte tipo de máquina.

multicloud

Um modelo de computação em nuvem no qual uma empresa usa uma combinação de arquitetura local, de nuvem privada e de nuvem pública.

N

namespace

Um cluster virtual dentro de um cluster do Kubernetes que pode ser usado para organizar e dividir recursos entre múltiplos usuários.

Network File System (NFS)

Um protocolo que permite que um computador acesse arquivos por meio de uma rede como se eles estivessem em seus discos locais.

NFS

Consulte Network File System .

nó de armazenamento

Um nó que é usado para fornecer o armazenamento de backend e o sistema de arquivos para armazenar os dados em um sistema.

nó de gerenciamento

Um nó opcional que hospeda somente serviços de gerenciamento, como monitoramento, medição e criação de log, e pode ser usado para evitar que o nó principal fique sobrecarregado.

nó de inicialização

Um nó que é usado para executar instalação, configuração, ajuste de escala do nó e atualizações de cluster.

nó do proxy

Um nó que transmite solicitações externas para os serviços que são criados dentro de um cluster.

nó do trabalhador

Em um cluster, uma máquina física ou virtual que contém as implementações e serviços que formam um aplicativo.

nó mestre

Um nó que fornece serviços de gerenciamento e controla os nós do trabalhador em um cluster. Os nós principais hospedam processos que são responsáveis pela alocação de recursos, manutenção do estado, planejamento e monitoramento.

O

org

Consulte organização.

organização (org)

No Cloud Foundry, o principal metaobjeto na infraestrutura que é gerenciado por uma conta com privilégios administrativos.

orquestração de contêiner

O processo de gerenciamento do ciclo de vida de contêineres, incluindo fornecimento, implementação e disponibilidade.

P

Pilha ELK

Os três produtos, Elasticsearch, Logstash e Kibana, que formam uma pilha de ferramentas que transmitem, armazenam, procuram e monitoram dados, incluindo logs.

política de localização

Uma política que define onde os componentes do aplicativo devem ser implementados e quantas réplicas devem existir.

política de segurança do pod

Uma política que é usada para configurar o controle de nível de cluster sobre o que um pod pode fazer ou o que ele pode acessar.

Prometheus

Um kit de ferramentas de monitoramento e alerta de sistemas de software livre.

R

RBAC

Consulte Controle de Acesso Baseado na Função.

rede da área de armazenamento virtual (VSAN)

Uma malha na rede de área de armazenamento (SAN).

registro

Um serviço de armazenamento e distribuição de imagem de contêiner público ou privado. Consulte também contêiner, imagem.

repo

Consulte repositório.

Repositório do Helm

Uma coleção de gráficos.

repositório (repo)

Uma área de armazenamento persistente para dados e outros recursos de aplicativos.

resource

Um componente físico ou lógico que pode ser fornecido ou reservado para um aplicativo ou instância de serviço. Exemplos de recursos incluem banco de dados, contas e processador, memória e limites de armazenamento.

S

segmento de isolamento

Uma divisão que pode ser usada para separar aplicativos como se estivessem em implementações diferentes, sem a necessidade de gerenciamento redundante e complexidade de rede.

serviço de criação de log de gerenciamento

Uma pilha ELK que é usada para coletar e armazenar todos os logs capturados pelo Docker.

solicitação de volume persistente

Uma solicitação para armazenamento em cluster.

Solution Pak

Consulte Cloud Pak.

syslog

Consulte log do sistema.

T

TA

Consulte Transformation Advisor.

team

Uma entidade que agrupa usuários e recursos.

terminal

Um endereço de destino de rede que é exposto pelos recursos do Kubernetes, como serviços e ingressos.

tipo de máquina (MT)

Uma configuração que é usada para instanciar uma máquina virtual.

tolerância a falhas

A capacidade de um sistema de continuar a funcionar eficazmente após a falha de uma parte componente. Consultar também alta disponibilidade.

Transformation Advisor (TA)

Uma ferramenta do desenvolvedor que é usada para avaliar apps Java EE no local para implementação na nuvem.

V

Virtual Machine File System (VMFS)

Um sistema de arquivo em cluster que permite a virtualização escalar além de um único nó para vários servidores ESX VMware.

VMFS

Consulte Virtual Machine File System.

volume persistente

Armazenamento de rede em um cluster que é provisionado por um administrador.

VSAN

Consulte rede da área de armazenamento virtual.

W

workload (carga de trabalho)

Uma coleção de servidores virtuais que desempenham um propósito coletivo definido pelo cliente. Uma carga de trabalho geralmente pode ser visualizada como um aplicativo com multicamadas. Cada carga de trabalho é associada a um conjunto de políticas que definem os objetivos de desempenho e de consumo de energia.

zona de disponibilidade

Um segmento de infraestrutura de rede designado pelo operador, funcionalmente independente.

Avisos

Estes avisos legais pertencem ao produto IBM® Cloud Private e à sua documentação.

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos EUA. Este material pode estar disponível na IBM em outros idiomas. Entretanto, você pode ser obrigado a ter uma cópia do produto ou da versão do produto naquele idioma para ter acesso a ele.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre os produtos e serviços disponíveis atualmente em sua área. Qualquer referência a um produto, programa ou serviço da IBM não pretende declarar ou inferir que somente aquele produto, programa ou serviço da IBM pode ser utilizado. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento dessa publicação não concede ao Cliente nenhuma licença para essas patentes. Consultas sobre licenças devem ser enviadas, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP: 22296-903

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Licenciamento de Propriedade Intelectual
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japão

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA" SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO LIMITADO ÀS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Algumas jurisdições não permitem a renúncia de garantias explícitas ou implícitas em determinadas transações, portanto, esta declaração pode não se aplicar a você.

Estas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas mudanças periódicas nas informações aqui contidas; tais mudanças serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar produtos ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais deste produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode usar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre ele com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações que foram trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP: 22296-903

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Os dados de desempenho discutidos aqui são apresentados como derivados sob as condições operacionais específicas. Os resultados reais podem variar.

As informações relativas a produtos não-IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre a capacidade de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

As instruções relacionadas à direção ou ao intento futuro da IBM estão sujeitas à mudança ou retirada sem aviso e representam somente metas e objetivos.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem os nomes de indivíduos, empresas, marcas e produtos. Todos esses nomes são fictícios e qualquer semelhança com pessoas ou empresas reais é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. É possível copiar, modificar e distribuir estes programas de amostra de qualquer forma sem pagamento à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativos para a plataforma operacional para a qual os programas de amostra são gravados. Esses exemplos não foram totalmente testados sob todas as condições. Portanto, a IBM não pode garantir ou confirmar a excelência em confiabilidade, desempenho ou função de tais programas. Os programas de amostra são fornecidos "NO ESTADO EM QUE SE ENCONTRAM", sem garantia de qualquer tipo. A IBM não deve ser responsabilizada por quaisquer danos decorrentes do uso dos programas de amostra.

Marcas registradas

IBM, o logotipo IBM e ibm.com são marcas ou marcas registradas da International Business Machines Corp., registradas em várias jurisdições em todo o mundo. Nomes de outros produtos e serviços podem ser marcas registradas da IBM ou de terceiros. Uma lista atual de marcas registradas da IBM está disponível na web em Copyright and trademark information .

Linux é marca registrada da Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada da The Open Group nos Estados Unidos e em outros países.

VMware, o logotipo do VMware, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server e VMware vSphere são marcas registradas ou marcas comerciais da VMware, Inc. ou de suas subsidiárias nos Estados Unidos e/ou em outras jurisdições.

Termos e Condições

Uso Pessoal

O Cliente pode reproduzir estas publicações para uso pessoal não comercial contanto que todos os avisos proprietários sejam preservados. O Cliente não poderá distribuir, exibir ou criar trabalhos derivativos destas publicações ou de qualquer parte das mesmas sem a autorização expressa, por escrito, da IBM.

uso comercial

É possível reproduzir, distribuir e exibir estas publicações unicamente dentro da empresa, desde que todos os avisos do proprietário sejam preservados. Não é possível fazer trabalhos derivativos dessas publicações ou reproduzir, distribuir ou exibir essas publicações ou qualquer parte delas fora de sua empresa, sem o consentimento expresso da IBM.

Direitos

Exceto conforme o que é expressamente concedido nessa permissão, nenhuma outra permissão, licença ou direito é concedido, expresso ou implícito, com relação às publicações ou quaisquer informações, dados, software ou outra propriedade intelectual contida aqui.

A IBM se reserva o direito de retirar as permissões concedidas neste instrumento sempre que, a seu critério, o uso destas publicações for prejudicial a seu interesse ou, como determinado pela IBM, as instruções acima não estiverem sendo seguidas apropriadamente.

Não é possível fazer o download, exportar ou reexportar estas informações, exceto em conformidade total com todas as leis e regulamentos aplicáveis, incluindo todas as leis e regulamentos de exportação dos Estados Unidos.

A IBM NÃO GARANTE O CONTEÚDO DESSAS PUBLICAÇÕES. ESTAS PUBLICAÇÕES SÃO FORNECIDAS "NO ESTADO EM QUE SE ENCONTRAM" E SEM QUAISQUER GARANTIAS DE QUALQUER TIPO, EXPRESSAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO SE LIMITANDO A, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, NÃO INFRAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO SÃO EXCLUÍDAS.

Licença

A auditoria de licença no IBM® Cloud Private é ativada por padrão para determinar se o uso atual está dentro dos níveis de autorização de licença e evitar potenciais violações de licença.

Para propósitos de licenciamento, o IBM Cloud Private é precificado por Núcleo de Processamento Virtual (VPC).

O IBM® License Metric Tool (ILMT) é usado para ajudá-lo a avaliar se você está em conformidade com os requisitos de licenciamento. O ILMT fornece recursos úteis para gerenciar ambientes virtualizados e medir o uso de licença. O ILMT descobre o software que está instalado em sua infraestrutura, ajuda você a analisar os dados de uso e permite gerar relatórios de auditoria. Cada relatório fornece a você informações diferentes sobre sua infraestrutura, por exemplo os grupos de computadores, instalações de software e o conteúdo de seu catálogo do software.

É possível usar o ILMT para determinar seu uso de VPC. Veja Determinando o uso de Virtual Processor Cores (VPC).

Por padrão, cada relatório de auditoria do ILMT apresenta dados dos 90 dias anteriores. É possível customizar o tipo e a quantidade de informações exibidas em um relatório usando filtros e salvar suas configurações pessoais para uso futuro. Também é possível exportar os relatórios para o formato .csv ou .pdf e planejar e-mails de relatório para que os destinatários especificados sejam notificados quando ocorrerem eventos importantes. Para obter mais informações, consulte o IBM License Metric Tool no IBM Knowledge Center.

Também é possível revisar o uso de licença usando o serviço de medição do IBM Cloud Private. Consulte Serviço de medição do IBM Cloud Private.