IBM Hybrid Cloud

ICP平台日常维护与管理

刘品一

ICP二线技术支持

主要内容

- k8s在ICP中的实现及参数设置
- ICP平台日常管理与维护
- 日志和监控服务优化

k8s在ICP中的实现及参数设置

ICP中k8s的实现

- K8s的各个组件, 除了kubelet,在ICP中都进行了容器化, 以static pod的形式运行
- 只有kubelet是采用Linux系统服务的方式 /opt/kubernetes/hyperkube

```
[root@ha311master1 ~]# kubectl -n kube-system get po | grep k8s
k8s-etcd-9.110.87.203                                  1/1       Running                 5          98d
k8s-etcd-9.110.87.204                                  1/1       Running                 6          98d
k8s-etcd-9.110.87.205                                  1/1       Running                 8          98d
k8s-master-9.110.87.203                                4/4       Running                 25         36d
k8s-master-9.110.87.204                                4/4       Running                 23         36d
k8s-master-9.110.87.205                                4/4       Running                 30         36d
k8s-master-vip-script-9.110.87.203                     1/1       Running                 5          98d
k8s-master-vip-script-9.110.87.204                     1/1       Running                 6          98d
k8s-master-vip-script-9.110.87.205                     1/1       Running                 8          98d
k8s-proxy-9.110.87.203                                 1/1       Running                 5          98d
k8s-proxy-9.110.87.204                                 1/1       Running                 6          98d
k8s-proxy-9.110.87.205                                 1/1       Running                 8          98d
k8s-proxy-9.110.87.206                                 1/1       Running                 5          98d
k8s-proxy-9.110.87.207                                 1/1       Running                 4          98d
k8s-proxy-9.110.87.208                                 1/1       Running                 4          98d
k8s-proxy-9.110.87.209                                 1/1       Running                 4          98d
k8s-proxy-9.110.87.211                                 1/1       Running                 2          98d
k8s-proxy-9.110.87.215                                 1/1       Running                 3          98d
k8s-proxy-9.110.87.216                                 1/1       Running                 6          98d
k8s-proxy-9.110.87.217                                 1/1       Running                 18         98d
k8s-proxy-vip-script-9.110.87.209                      1/1       Running                 4          98d
k8s-proxy-vip-script-9.110.87.211                      1/1       Running                 2          98d
k8s-proxy-vip-script-9.110.87.215                      1/1       Running                 3          98d
```

K8s参数的设置和调整

- K8s master, etcd 和 kube-proxy
  - 配置文件在每个Master节点的/etc/cfc/pods/ 目录
  - ├── etcd.json
  - ├── kube-proxy.json
  - ├── master.json
  - └── master-vip-script.json

- Kubelet 服务
  - 配置文件在每个节点的/etc/cfc/kubelet
  - ├── kubelet-config
  - └── kubelet-service-config

ICP平台日常维护与管理

Healthcheck工具

- 在boot节点的 cluster 目录下运行以下命令:
  - For Linux® x86_64:

```
sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster
ibmcom/icp-inception-amd64:3.1.1-ee healthcheck -v
```

  - For Linux® on Power® (ppc64le):

```
sudo docker run --net=host -t -e LICENSE=accept -v "$(pwd)":/installer/cluster
ibmcom/icp-inception-ppc64le:3.1.1-ee healthcheck –v
```

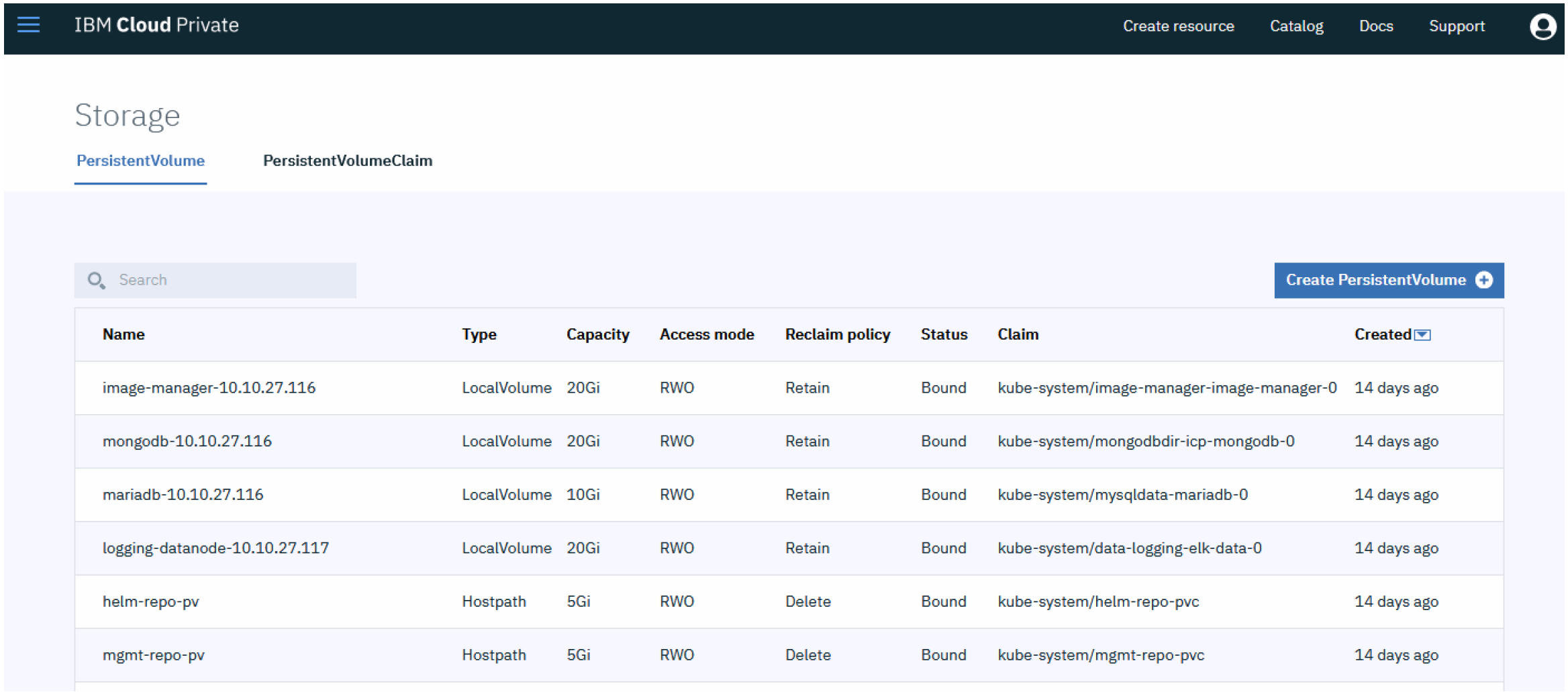
- 该命令会在cluster/logs目录下生成 healthcheck 目录, 存放ICP平台的详细信息
- ICP3.1.1, 您需要手工创建 cluster/logs/healthcheck/pods 目录

ICP组件

- IBM Cloud Private 两个主要组件：容器管理器 (Docker) 和容器编排器 (Kubernetes)
- IBM Cloud Private 集群的其他组件与这些主要组件配合工作，以提供认证、存储、网络、日志记录和监视等服务。 另外，还提供了集群 management console，它充当这些服务的集中式管理位置。
- 管理组件（例如，监视、测量和日志记录）在管理节点上运行。 如果集群中没有dedicated 管理节点，那么管理组件将在master主节点上运行。
- VA组件必须在专用的VA节点上运行
- ICP组件列表

ICP平台服务持续化存储

- ICP使用PV的方式为多个平台服务提供持续化存储



Name	Type	Capacity	Access mode	Reclaim policy	Status	Claim	Created
image-manager-10.10.27.116	LocalVolume	20Gi	RWO	Retain	Bound	kube-system/image-manager-image-manager-0	14 days ago
mongodb-10.10.27.116	LocalVolume	20Gi	RWO	Retain	Bound	kube-system/mongodbdir-icp-mongodb-0	14 days ago
mariadb-10.10.27.116	LocalVolume	10Gi	RWO	Retain	Bound	kube-system/mysqldata-mariadb-0	14 days ago
logging-datanode-10.10.27.117	LocalVolume	20Gi	RWO	Retain	Bound	kube-system/data-logging-elk-data-0	14 days ago
helm-repo-pv	Hostpath	5Gi	RWO	Delete	Bound	kube-system/helm-repo-pvc	14 days ago
mgmt-repo-pv	Hostpath	5Gi	RWO	Delete	Bound	kube-system/mgmt-repo-pvc	14 days ago

ICP命令行快捷登录

- cloudctl login -a https://9.110.87.50:8443 --skip-ssl-validation -u admin -p admin -c id-mycluster-account -n cert-manager

```
[root@ha311master1 ~]# cloudctl login --help
NAME:
   login - Log user in.

USAGE:
   cloudctl login [-a CLUSTER_URL] [-u USERNAME] [-p PASSWORD] [-c ACCOUNT_ID or ACCOUNT_NAME] [-n namespace]

      WARNING:   Providing your password as a command line option is not recommended.
                 Your password might be visible to others and might be recorded in your shell history.

      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


PARAMETERS:
   -a value                The URL that you use to access the management console, such as https://<ip_address>:
   -u value                Username
   -p value                Password
   -c value                Account ID or name
   -n value                Namespace
   --skip-ssl-validation   Bypass SSL validation of HTTP requests. This option is not recommended.
```

cloudctl登录启用trace

- 如果登陆失败, 可以用如下命令启用trace, 已查看更多信息

```
[root@ha311master1 ~]# cloudctl config --trace=true
[root@ha311master1 ~]# ./cloudctllogin.sh
preparing to make request to import config: &{GET https://9.110.87.50:8443/api/v1/config map[Accept-Language:[] Accept:[application/json]] map[
] map[] map[] <nil>}
Authenticating...
preparing to make request to get token: &{POST https://9.110.87.50:8443/idprovider/v1/auth/identitytoken map[Accept:[application/json] Content-
Type:[application/x-www-form-urlencoded;charset=UTF-8] Accept-Language:[]] map[] map[scope:[openid] username:[admin] password:[PRIVATE DATA HID
DEN] grant_type:[password]] map[] <nil>}
OK

preparing to make request to get user info: &{POST https://9.110.87.50:8443/idprovider/v1/auth/userInfo map[Accept:[application/json] Content-T
ype:[application/x-www-form-urlencoded;charset=UTF-8] Accept-Language:[]] map[] map[access_token:[PRIVATE DATA HIDDEN]] map[] <nil>}
getting a list of accounts...
found accounts: [{id-mycluster-account mycluster Account }]
Targeted account mycluster Account (id-mycluster-account)

getting list of namespaces...
getting namespaces with url: https://9.110.87.50:8443/idmgmt/identity/api/v1/users/admin/getTeamResources?resourceType=namespace
found namespaces: [cert-manager default ibmcom istio-system kube-public kube-system nsicpl2 platform services state-mongo]
Targeted namespace cert-manager

Downloading binary from /cli/config to /root/.cloudctl/clusters/kubeConfig-3227.zip
Located unzipped directory: /root/.cloudctl/clusters/kubeConfig357722066
Current OS version detected: linux
Configuring kubectl ...
Property "clusters.mycluster" unset.
Property "users.mycluster-user" unset.
Property "contexts.mycluster-context" unset.
Cluster "mycluster" set.
User "mycluster-user" set.
Context "mycluster-context" created.
Switched to context "mycluster-context".
OK
Configuring Helm certificates
HELM_HOME variable is not set.
HELM_HOME is set to /root/.helm

Configuring helm: /root/.helm
OK
```

当kubectl命令无法使用时

- 如果无法登录进ICP 系统, Kubectl命令由于没有配置context 信息无法使用

```
kubectl --kubeconfig=/var/lib/kubelet/kubectl-config -n get nodes
```

```
alias kc='kubectl -n kube-system --kubeconfig=/var/lib/kubelet/kubectl-config'
```

```
[root@ha311master1 ~]# kubectl --kubeconfig=/var/lib/kubelet/kubectl-config -n kube-system get nodes
NAME           STATUS    ROLES              AGE       VERSION
9.110.87.203   Ready     etcd,master        99d       v1.11.3+icp-ee
9.110.87.204   Ready     etcd,master        99d       v1.11.3+icp-ee
9.110.87.205   Ready     etcd,master        99d       v1.11.3+icp-ee
9.110.87.206   Ready     glusterfs,worker   99d       v1.11.3+icp-ee
9.110.87.207   Ready     glusterfs,worker   99d       v1.11.3+icp-ee
9.110.87.208   Ready     glusterfs,worker   99d       v1.11.3+icp-ee
9.110.87.209   Ready     proxy              99d       v1.11.3+icp-ee
9.110.87.211   Ready     proxy              99d       v1.11.3+icp-ee
9.110.87.215   Ready     proxy              99d       v1.11.3+icp-ee
9.110.87.216   Ready     management         99d       v1.11.3+icp-ee
9.110.87.217   Ready     va                 99d       v1.11.3+icp-ee
```

节点维护

1. 运行以下命令来将节点标记为不可调度：

```
kubectl cordon 9.111.255.122
```

注：将节点标记为不可调度会在该节点上禁用调度新 pod。

2. 运行以下命令来清空准备进行维护的节点，以移除正在该节点上运行的 pod：

```
kubectl drain 9.111.255.122 --grace-period=300 --ignore-daemonsets=true
```

有关清空节点的其他信息，请通过输入以下内容来参阅 kubectl drain 命令的帮助：kubectl help drain。由于节点已标记为不可调度，因此未将 ReplicationController、ReplicaSet、Job 和 StatefulSet 部署中的 pod 调度到此节点。 调度程序将所有工作负载移至可调度的另一个节点。

注：无需从 DaemonSet 部署中清空节点。

3. 运行以下命令来退出维护：

```
kubectl uncordon 9.111.255.122
```

添加和删除ICP集群节点

- ICP集群安装完成后,无法添加或移除主节点
- ICP集群安装完成后, 您可以添加和删除以下节点:
  - 工作程序节点
  - 代理节点
  - 管理节点
  - 漏洞顾问程序节点
  - 定制主机组节点

https://www.ibm.com/support/knowledgecenter/zh/SSBS6K_3.1.1/installing/modify_cluster.html

- 添加和删除节点时, ICP将根据命令中指定的 IP 地址自动更新 hosts 文件

移除无响应的 ICP集群节点

- 如果节点无响应, 可以通过以下命令删除该节点

```
kubectl delete node <node-IP-address>
```

- 以下是样本命令和输出:

```
kubectl delete node 172.16.151.182
node "172.16.151.182" deleted
```

- 通过该方法删除无响应节点后, 需手动在hosts 文件中移除无响应节点的 IP 地址, 否则ICP的节点信息可能出现不一致的情况

```
kubectl get nodes
```

更改 ICP集群节点 IP 地址或主机名

- 在安装 ICP集群后，不支持直接更改节点的 IP 地址或主机名
- 但是, 您可以用以下步骤, 间接更改集群中节点的 IP 地址或主机名
  1. 从集群中移除节点。
  2. 按需更改节点的 IP 地址或主机名
  3. 将节点添加回ICP集群。

https://www.ibm.com/support/knowledgecenter/zh/SSBS6K_3.1.1/installing/change_node.html

- 注意:由于**ICP**安装后将不能移除和重新添加**master**主节点, 因此,该方法不能更改已安装集群中的**master**主节点的 **IP** 地址或主机名。

替换主节点

- 硬件故障发生时，您可能会发现主节点未正常工作，完成以下步骤以替换主节点：
  1. 重新创建新的主节点。 **确保新主节点具有与旧主节点相同的主机名、IP 地址和接口名称。**
  2. 为新的主节点配置 SSH 认证。 **保持与旧主节点相同的密码或 SSH 密钥认证。**
  3. 运行命令以替换主节点(以Linux on X86为例)：

```
docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp-inception-amd64:3.1.1-ee install -l master-node-ip
```

  - 如果 etcd pod 在新的主节点上运行，那么它可能会启动失败，您还需要更新etcd成员，请参考以下文档：

```
https://www.ibm.com/support/knowledgecenter/zh/SSBS6K_3.1.1/troubleshoot/replace_master_node.html
```

查看节点工作负载

- 可以使用节点描述命令来查看节点的负载信息

```
kubectl describe node 9.110.87.208
```

```
Capacity:
 cpu:                4
 ephemeral-storage:  99561988Ki
 hugepages-2Mi:      0
 memory:             8174116Ki
 pods:               40
Allocatable:
 cpu:                4
 ephemeral-storage:  91756327989
 hugepages-2Mi:      0
 memory:             8071716Ki
 pods:               40
```

```
Allocated resources:
  (Total limits may be over 100 percent,
  Resource   Requests      Limits
  --------   --------      ------
  cpu        1800m (45%)   2 (50%)
  memory     1814Mi (23%)  2Gi (25%)
```

增加system和 kubelet 预留资源

- 编辑每个节点上的/etc/cfc/kubelet/kubelet-service-config文件并保存

```
systemReserved:
  cpu: "200m"
  memory: "512Mi"
  ephemeral-storage: "1Gi"
kubeReserved:
  cpu: "200m"
  memory: "512Mi"
  ephemeral-storage: "1Gi"
```

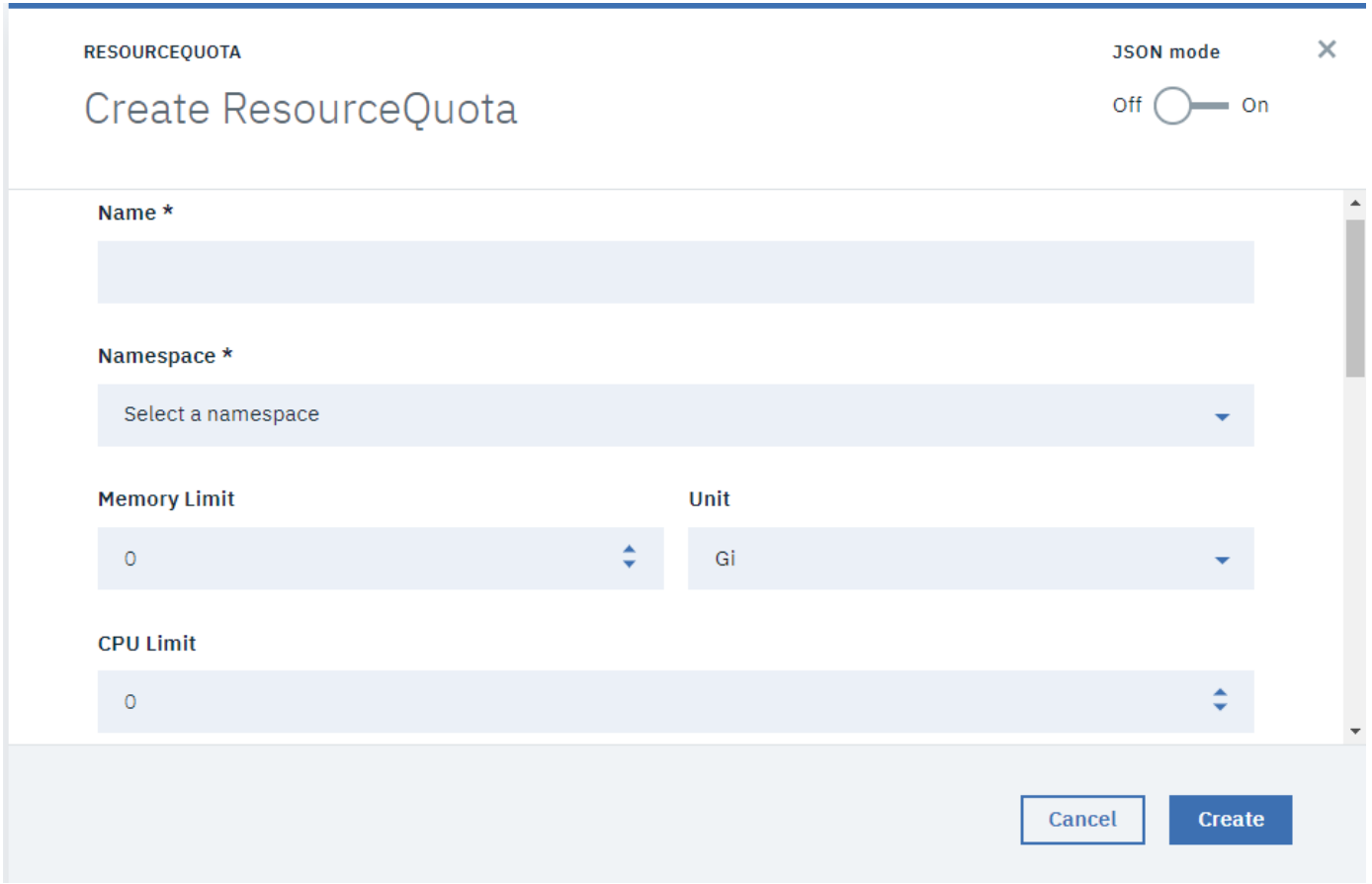
- 重启kubelet服务

```
systemctl restart kubelet
```

设置资源配额

- 配置资源限制以通过命名空间限制应用程序能够请求的计算和存储资源量

1. 从导航菜单中，单击**管理>资源安全->配额。**
2. 单击**创建 ResourceQuota**
3. 输入资源配额详细信息。
4. 单击**创建。**



迁移etcd数据到独立高速磁盘

- mount /var/lib/etcd 到独立的高速磁盘, 可以遵循以下步骤
  1. 停止etcd: `mv /etc/cfc/pods/etcd.json /etc/cfc/etcd.json`
  2. 把/var/lib/etcd 和 /var/lib/etcd-wal 的数据移动一个临时目录
  3. mount /var/lib/etcd 和 /var/lib/etcd-wal 到独立的高速磁盘
  4. 把/var/lib/etcd 和 /var/lib/etcd-wal 数据再挪回来
  5. 启动etcd: `mv /etc/cfc/etcd.json /etc/cfc/pods/etcd.json`

- 重要说明: 请勿在/etc/cfc/pods下创建任何备份文件，例如不要运行以下命令:

```
cp /etc/cfc/pods/etcd.json /etc/cfc/pods/etcd.json.bak
```

etcd metrics收集

```
curl --cacert /etc/cfc/conf/etcd/ca.pem --cert /etc/cfc/conf/etcd/client.pem --key
/etc/cfc/conf/etcd/client-key.pem https://9.110.87.203:4001/metrics > etcd-metrics.txt
```

```
[root@ha311master1 ~]# curl --cacert /etc/cfc/conf/etcd/ca.pem --cert /etc/cfc/conf/etcd/client.p
em --key /etc/cfc/conf/etcd/client-key.pem https://9.110.87.203:4001/metrics > etcd-metrics.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 37777  100 37777    0     0   685k      0 --:--:-- --:--:-- --:--:--  696k
[root@ha311master1 ~]#
```

etcd-metrics.txt

etcd空间配额设置

空间配额

使用标志 --quota-backend-bytes 命令来设置空间配额。 空间配额的缺省值为 2 GB，它是适合于大多数应用程序的保留空间配额。 最大值为 8 GB。

您可以在安装前后更改空间配额值。

- 要在安装 IBM Cloud Private 之前配置 --quota-backend-bytes，请按如下编辑 cluster/config.yaml；使用 1G 作为示例：

```
etcd_extra_args: ["--quota-backend-bytes=1073741824"]
```

- 要在安装 IBM Cloud Private 之后配置 --quota-backend-bytes，请在 IBM Cloud Private 主节点上编辑 /etc/cfc/pods/etcd.json 文件，然后向 etcd 命令添加以下行：

```
"--quota-backend-bytes=1073741824",
```

日志和监控服务优化

监控服务仪表盘无数据

- 查看监控服务相关pod

```
monitoring-prometheus-55d4949dfb-89l6q 3/4 Running 1086 6d 10.1.207.247 icp311-management-2 <none>
```

```
 prometheus:
 Container ID:  docker://12a8630ac2f554cbc25ec089efb602bfae3cc29a2cf8a8d1999f29629f09a197
 Image:   mycluster.icp:8500/ibmcom/prometheus:v2.3.1
 Image ID:  docker-pullable://mycluster.icp:8500/ibmcom/prometheus@sha256:91cba3cbf7a61499969a8a5a27347830ed1740b2285de9ac31e231e7ce44deb7
 Port:   9090/TCP
 Host Port:  0/TCP
 Args:
  --config.file=/etc/config/prometheus.yml
  --web.enable-lifecycle
  --web.enable-admin-api
  --storage.tsdb.path=/var/lib/prometheus/data
  --storage.tsdb.retention=24h
  --web.external-url=https://127.0.0.1:8443/prometheus
 State:   Running
  Started:  Thu, 20 Jun 2019 07:09:17 +0000
 Last State:  Terminated
  Reason:  OOMKilled
  Exit Code: 137
  Started:  Thu, 20 Jun 2019 07:00:19 +0000
  Finished:  Thu, 20 Jun 2019 07:08:27 +0000
 Ready:   False
 Restart Count: 1086

```

增大内存和CPU资源请求和限制

- 如何增加Prometheus 容器的内存和CPU资源
  1. 在控制台找到monitoring-prometheus 部署
  2. 将monitoring-prometheus 部署的实例减少为0个
  3. 编辑monitoring-prometheus 部署, 找到

```
202 "resources": {
203   "limits": {
204     "cpu": "1",
205     "memory": "4Gi"
206   },
207   "requests": {
208     "cpu": "500m",
209     "memory": "2Gi"
210   }
```

  4. 保存退出
  5. 将monitoring-prometheus部署的实例恢复为1个
  6. monitoring-prometheus部署将会自动生成新的pod, 并按照修改后的分配cpu和memory资源

日志和监控服务优化

monitoring

Name	Namespace	Desired	Current	Ready	Available
monitoring-grafana	kube-system	1	1	1	1
monitoring-prometheus	kube-system	1	1	1	1

logging

Create Deployment

Name	Namespace	Desired	Current	Ready	Available	Created	
logging-elk-client	kube-system	1	1	-	-	6 days ago	Launch
logging-elk-kibana	kube-system	1	1	1	1	6 days ago	Launch
logging-elk-logstash	kube-system	1	1	1	1	6 days ago	Launch
logging-elk-master	kube-system	1	1	1	1	6 days ago	Launch

StatefulSets

Search

Name	Namespace	Desired	Current	Created
logging-elk-data	kube-system	1	1	6 days ago

监控服务历史数据保存时长调整

- ICP监控服务的历史数据默认保存时间是24小时 -即使启用了persistent volume
- 启用persistent volume只是决定了监控数据的存储位置
- 设置方法同增加内存和CPU资源限制, 设置参数为 --storage.tsdb.retention

```
309 monitoring:
310   prometheus:
311     scrapeInterval: 1m
312     evaluationInterval: 1m
313     retention: 24h
314     persistentVolume:
315       enabled: true
316       storageClass: "glusterfs"
```

```
186         "name": "prometheus",
187         "image": "mycluster.icp:8500/ibmcom/prometheus:v2.3.1",
188         "args": [
189           "--config.file=/etc/config/prometheus.yml",
190           "--web.enable-lifecycle",
191           "--web.enable-admin-api",
192           "--storage.tsdb.path=/var/lib/prometheus/data",
193           "--storage.tsdb.retention=240h",
194           "--web.external-url=https://9.110.87.50:8443/prometheus"
```

THANK YOU