IBM Cloud Pak for AIOps 4.8.1- Infrastructure Automation
PDF



© Copyright IBM Corp. 2025.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Tables of Contents

Infrastructure automation 1
- Planning 2
  - Supported platforms 2
  - Hardware requirements 4
  - Storage requirements 5
  - High availability considerations 9
  - Security considerations 9
  - Licensing 10
  - Passport Advantage part numbers 12
- Installing 12
  - Online with Cloud Pak for AIOps (console) 13
  - Online with Cloud Pak for AIOps (CLI) 18
  - Online stand-alone (console) 24
  - Online stand-alone (CLI) 31
  - Installing on cloud platforms 39
    - Installing on ROKS 39
    - Installing on ROSA 40
    - Installing on ARO 43
  - Offline 47
    - Bastion host 48
    - Portable device 54
  - Installing Infrastructure Management 61
    - Installing on AWS (appliance) 62
    - Installing on VMware vSphere (appliance) 69
    - Installing on Red Hat OpenStack Platform (appliance) 75
    - Installing on Red Hat Virtualization (appliance) 84
    - Installing on Google Compute Engine (appliance) 90
    - OIDC Configuration steps for Infrastructure Management with Cloud Pak (appliance) 96
    - Installing Infrastructure management as a containerized deployment with Cloud Pak (podified) 101
    - Reference 102
      - Infrastructure Automation CRDs 102
        - IAConfig 102
        - IAManageService 103
        - IMInstall 106
    - Troubleshooting installing and uninstalling 109
- Upgrading 111
  - Online (console) 113
  - Online (CLI) 114
  - Offline 116
  - Configuring automatic patch upgrades 118
  - Migrating to Infrastructure management appliance 120
  - Upgrading Infrastructure management appliances 122
    - Upgrade an Infrastructure management appliance from 3.6.x 123
    - Upgrade an Infrastructure management VMDB appliance 124
    - Upgrade an Infrastructure management Non-VMDB appliance 125
    - Upgrade a dedicated database appliance 126
  - Migrate from an appliance-based to a Kubernetes-based Infrastructure Management installation 127
- Administering 131
  - User interface 132
  - Roles and permissions 134
  - Backup and restore 138
    - Installing the backup and restore tools 139
    - Backing up Infrastructure Automation 144
    - Restoring Infrastructure Automation 148
    - Upgrading Infrastructure Automation backup and restore artifacts 153
    - Backup and restore Infrastructure management appliances 154
  - Migrating to IBM Cloud Pak for AIOps 155
    - Migrating IBM Cloud Private {{site.data.keyword.infra_mgmt_manageservices}} 155
    - Migrating IBM Red Hat CloudForms and {{site.data.keyword.cp4mcm_notm}} appliances 120
- Infrastructure Management 160
  - Administration 160
    - Deployment planning guide 161
    - High availability guide 183
    - Appliance hardening guide 196
    - Enable terraform-runner in an air-gapped environment 205
  - Configuring user settings 205
  - Managing infrastructure and inventory 290
  - Managing providers 371
    - Red Hat Virtualization Providers 419

OpenStack Infrastructure Providers 421
VMware vCenter Providers 422
IBM Power HMC Providers 424
OpenShift Virtualization (KubeVirt) 424
IBM Terraform Providers 425
Red Hat Satellite 6 Providers 425
Azure Providers 428
Amazon EC2 Providers 430
Google Compute Engine Providers 434
IBM CIC Providers 436
IBM Cloud VPC Providers Providers 437
IBM Power Systems Virtual Servers Providers 438
IBM PowerVC Providers 440
OpenStack Providers 442
Azure Kubernetes Providers 444
Red Hat OpenShift Providers 444
IBM Cloud Kubernetes Service Providers 447
Oracle Kubernetes Engine Providers 448
VMware Tanzu Providers 448
Amazon Elastic Block Store Providers 449
OpenStack Block Storage Providers 449
OpenStack Object Storage Providers 450
IBM Cloud Object Storage Providers 450
Provisioning virtual machines and instances 452
Creating a service in Infrastructure management for virtual machine provisioning 498
Service provisioning in Infrastructure management using an orchestration template 501
Managing controls 503
Policies and profiles guide 503
Monitoring, alerts, and reporting 536
Chargeback 571
Managing integrations 575
ServiceNow integration 575
Change risk integration 577
Infrastructure Management to {{site.data.keyword.cp4waiops_notm}} integrations 578
Assigning a custom analysis profile to a virtual machine 580
Scripting Actions in Infrastructure Management (stand-alone) 584
APIs 605
API overview 606
Prerequisites 606
REST API Authorization 606
Password Updates 607
CRUD 607
HTTP Basics 608
JSON Specification 611
Query Specification 615
Bulk Querying 617
Collection Filtering 618
Pagination 619
API reference 620
Primary Collections 621
Actions 623
Alert Management 624
Authentication Management 626
Automate Model 628
Automate Domains 631
Automation Requests 632
Automate Workspaces 634
Categories and Tags 636
Chargebacks, Rates and Assignments 640
Cloud Volumes 662
Conditions 662
Configuration Script Payloads 664
Configuration Script Source Management 668
Custom Actions 670
Custom Attributes 674
Custom Button Management 676
Event Streams 679
Flavor Management 679
Generic Object Management 680

Group Management 686
Host Management 688
Instance Management 689
Metric Rollups 690
Network Routers 691
Notifications Support 692
Orchestration Support 695
Physical Server Management 696
Picture Management 698
Policy Management 699
Policy Collections 705
Provider Support 707
Provision Requests 714
Request Support 717
Reports Support 719
Role Management 725
Security Groups 727
Service Management 728
Service Queries 737
Service Custom Actions 739
Service Dialogs 744
Service Orders 746
Setting Ownership 749
Settings Management 750
Snapshot Management 751
Tags and Tagging 753
Tasks 758
Tenant Management 758
User Management 760
Virtual Machine Management 762
API appendices 765
Quick Reference 766
Collection Metadata 767
Resource Attributes 769
Provider Types 776
Provision Request Supported Attributes 777
Advanced Provisioning Attributes 779
SOAP to REST Mapping guide 780
Deprecated Types 782
Deprecated Resource Attributes 783
API Examples 783
Foundation 783
Querying Examples 783
Paging Queries 784
Automation Requests 786
Create an Automation Request 786
Create multiple Automation Requests 787
Providers 789
Create a Provider 790
Create a Provider with multiple credentials 790
Update a Provider 791
Refresh a Provider 791
Delete a Provider 791
Provisioning Requests 793
Create a Provision Request 793
Create multiple Provision Requests 795
Service Catalogs 801
Add a Service Catalog 801
Add multiple Service Catalogs 802
Edit a Service Catalog 802
Edit Service Catalogs 803
Assign Service Templates 803
Unassign Service Templates 804
Delete Service Catalogs 804
Service Templates 805
Edit a Service Template 805
Edit Service Templates 805
Delete Service Templates 806

Services 806
Order a Service 807
Order Multiple Services 807
Edit a Service 808
Edit a Service with Patch Method 809
Edit a Service with PUT 809
Edit Services 809
Reconfigure a Service 810
Refresh Service Template Dialog Fields 811
Refresh Service Dialog Fields 812
Retire a Service 813
Retire Multiple Services 814
Trigger a Custom Action on a Service 814
Delete Services 817
VM Management 818
Start a VM 818
Stop a VM 819
Suspend a VM 819
Scan a VM 819
Set owner of a VM 820
Add Event to a VM 820
Delete a VM 820
Tag Management 821
Assign Tags 821
Assign Tags to Service Template 823
Unassign Tags 823
Unassign Tags from Service Template 824
Usage Flows 825
Compliance Policy Flow 825
Reference 833
Support matrix 834
Integration with OpenShift Container Platform 845
Methods available for automation 851
Managing authentication for Infrastructure management (stand-alone) 878
Active Directory 878
2-Factor-Authentication with IPA 880
IPA/AD trust authentication 881
LDAP authentication 883
SAML authentication 886
OpenID-Connect authentication 890
Integration with OpenShift Container Platform 845
Self service user interface guide 899
Troubleshooting 908
After an upgrade, users cannot access Infrastructure Management 908
Error using multiple tabs in the same browser 908
Non-LDAP users cannot access Infrastructure Management 909
The vmware-infra-* pods in CrashLoopBackOff error states 909
VM operator encounters an error when creating CRD 909
VM operator encounters an error when upgrading 910
Error deleting cloud volumes 910
Error logging out of and logging in to Infrastructure Management 911
New data is not shown from Infrastructure Management 911
`infrastructure-management-zookeeper-x` pod is not Ready on the Red Hat OpenShift on IBM Cloud cluster 912
Managed services 913
Uninstalling 913

Infrastructure Automation

Infrastructure Automation is a stand-alone capability/module included within IBM Cloud Pak for AIOps. When you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation.

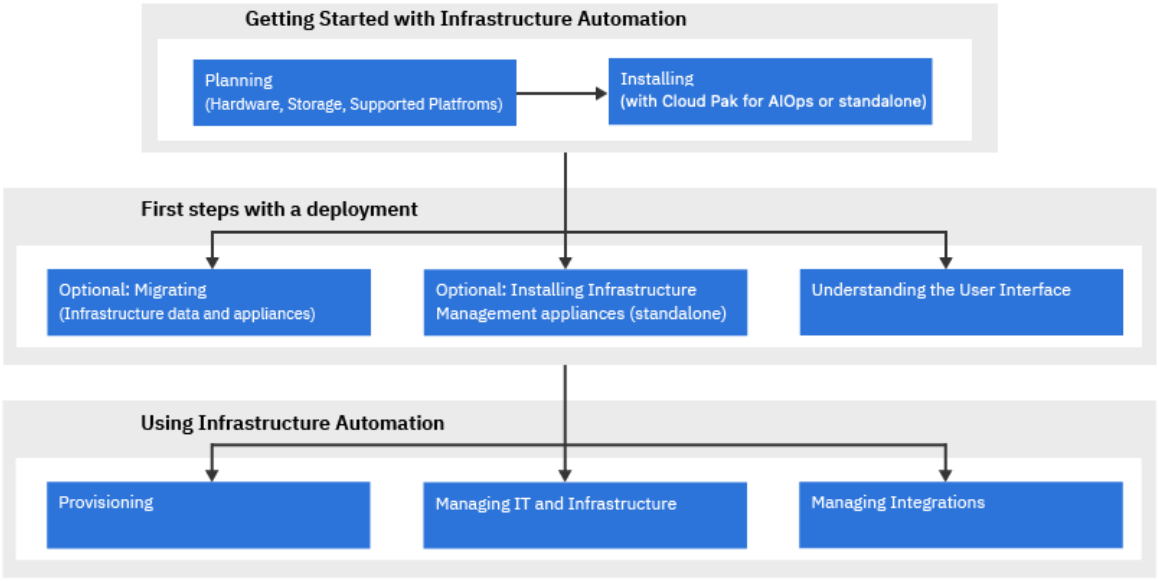
Infrastructure Automation brings together Terraform and ManageIQ for an integrated, Infrastructure Automation capability for IT Operators.

Infrastructure Automation delivers two distinct features:

- Managed services, previously called Terraform and Service Automation. This feature uses self-service capabilities to orchestrate resources.
- Infrastructure Management, previously called IBM Red Hat CloudForms. This feature provides capabilities for discovering, managing, and automating the deployment of VMs, cloud services and Kubernetes clusters.

To learn more about Infrastructure Automation, click a subject within the following image to view the related available documentation (or review the subjects that are listed following the image):

Infrastructure Automation documentation flow



Getting started

Planning

Before you install IIBM Cloud Pak for AIOps - Infrastructure Automation, plan your installation, configuration, and overall cluster requirements. For more information, see Planning.

Installing

Infrastructure Automation is included with IBM Cloud Pak for AIOps and can be installed for use with IBM Cloud Pak for AIOps or as a stand-alone component. For information about, see Installing Infrastructure automation.

First steps with a deployment

Migrating data from another deployment

You can migrate infrastructure data, such as services data and appliances, to Infrastructure Automation from other products and services, such as IBM Cloud Pak for Multicloud Management, IBM Cloud Private, and IBM Red Hat CloudForms. For more information, see Migrating to IBM Cloud Pak for AIOps.

Installing Infrastructure Management appliances

With Infrastructure Automation installed, you can install Infrastructure management appliances for your environment. These appliances can be installed on different supported platforms, such as Red Hat OpenStack Platform and Red Hat Virtualization, VMware vSphere, and Amazon Elastic Compute Cloud (Amazon EC2). For more information, see Installing and upgrading appliances.

Understanding the user interface

Learn how to use the IBM Cloud Pak for AIOps console to access and use Infrastructure Automation tools and features. For more information, see User interface.

Infrastructure Automation use cases

Once set up, you and your DevOps teams can use Infrastructure Automation to provision, integrate, and manage IT and infrastructure.

Provisioning

Infrastructure Automation provide the capability to automate provisioning of infrastructure and virtual machine applications using Terraform and Ansible. Provisioning IT brings infrastructure-as-code support and helps with sharing templates across multiple instances and with provisioning repeatable infrastructure with confidence. This infrastructure-as-code support helps to provide you with a standardized environment for your developers. For more information, see Provisioning.

Managing infrastructure and inventory

Infrastructure Automation includes supported versions of Terraform that works across multiple cloud environments like RHEV, VMWare, OpenStack and multiple cloud providers, like AWS, Microsoft Azure, Google, and IBM Cloud. For more information, see Managing IT and infrastructure.

Managing integrations

With Infrastructure Automation, you can enable default integrations with ServiceNow, Ansible Tower, Turbonomic, and Business Process Manager. Once integrations are setup, the cloud infrastructures can be provisioned through the Infrastructure Automation catalog or through APIs. Changes and operations to the provisioned infrastructures go through Infrastructure Automation, which can synchronize the changes with other systems. For more information, see Managing integrations.

Planning an installation of IBM Infrastructure Automation

Review the following information to help you plan your installation of IBM Infrastructure Automation.

Supported platforms

Learn about the Red Hat® OpenShift® Container Platform versions and public clouds that you can deploy on. Supported platforms.

Storage requirements

Review the storage requirements for your deployment, and learn about supported storage providers and storage classes. For more infomation, see Storage requirements.

Hardware requirements

Review the memory and hardware requirements for your deployment. For more information, see Hardware requirements.

High availability

Review information about how to configure high availability (HA) for your deployment. For more information, see High availability (HA) considerations.

Security considerations

Infrastructure Automation supports different mechanisms for securing your environment and your data. In addition, as you are planning or going through an installation of Infrastructure Automation, you might need to create or provide certificates. For more information, see Security considerations.

Licensing

As part of using Infrastructure Automation, tracking your license usage is required. Although Infrastructure Automation provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak foundational services License Service on all clusters where Infrastructure Automation is installed. This background service collects and stores the license usage information for tracking license consumption and for audit purposes. For more information, Licensing.

Passport Advantage part numbers

You might need to download a package from IBM® Passport Advantage® to install Infrastructure Automation. This requirement can be due to your entitlement or based on how you choose to install Infrastructure Automation. For more information, see Passport Advantage part numbers.

Supported platforms

Learn about which Red Hat OpenShift Container Platform versions and public cloud platforms are supported for Infrastructure Automation.

- Red Hat® OpenShift® Container Platform
- Cloud platforms
  - IBM Cloud®
  - AWS Marketplace

- Microsoft® Azure

Red Hat OpenShift Container Platform

Infrastructure Automation adheres to the IBM Cloud Pak and Red Hat OpenShift version compatibility objective, and support and recommend installing on a supported Red Hat OpenShift EUS minor version.

For more information about support timelines for Red Hat OpenShift, including when a specific version is going out of support, see the Red Hat OpenShift Container Platform Life-Cycle Policy in the Red Hat OpenShift documentation.

Table 1. Supported OpenShift versions

Deployment	Platform	Red Hat OpenShift Container Platform version
Infrastructure Automation (installed with IBM Cloud Pak for AIOps)	Linux x86_64	4.14 4.15 4.16 (minimum 4.16.4) 4.17
Infrastructure Automation (installed stand-alone)	Linux x86_64 Linux on Power (ppc64le) (non-FIPs) Linux® on IBM Z® and LinuxONE (s390x)	4.14 4.15 4.16 (minimum 4.16.4) 4.17

Important:

- If you are planning to upgrade Red Hat OpenShift as part of an upgrade to Infrastructure Automation 4.8.1, you must complete the Infrastructure Automation upgrade before you upgrade Red Hat OpenShift.
- The hardware architecture for installing IBM Cloud Pak for AIOps must be x86_64 (amd64). For more information about hardware requirements for IBM Cloud Pak for AIOps, see IBM Cloud Pak for AIOps: Supported hardware.
- Red Hat OpenShift v4.14 and onwards support heterogeneous architectures, but Infrastructure Automation does not support heterogeneous architectures. All nodes in the Red Hat OpenShift cluster must have the same architecture.
- If you are installing Infrastructure Automation on Linux on Power (ppc64le) with FIPS enabled, see the following known issue: Infrastructure Automation install fails on FIPS enabled Power cluster.

Red Hat OpenShift installation mode

Infrastructure Automation supports the OpenShift **`OwnNamespace`** and **`AllNamespaces`** installation modes. The **`OwnNamespace`** installation mode is the recommended installation mode because it provides better resilience and performance.

In the **`OwnNamespace`** installation mode, you must create and select a project (namespace) for the Infrastructure Automation operator to be installed into. In this mode, the operator watches and is available for use only in this single namespace. The **`All namespaces on the cluster (default)`** installation mode in the UI installation of the Infrastructure Automation operator is equivalent to the **`AllNamespaces`** mode in the CLI installation. In this mode, the operator is installed in the **`openshift-operators`** project (namespace), and watches and is available to all namespaces in the cluster. The Operator Deployment Lifecycle Manager (ODLM) installs suboperators in the same project (namespace) that you created the **`Installation`** custom resource instance in. Each suboperator watches and is available for use in its project (namespace).

For more information on Red Hat OpenShift operator installation modes, see Operator groups in the Red Hat OpenShift Container Platform documentation.

Note: If you install IBM Cloud Pak for AIOps and Infrastructure Automation, then they must use the same installation mode.

Resource allocation for nodes

Red Hat OpenShift nodes must have adequate resources reserved. For more information, see Allocating resources for nodes in an OpenShift Container Platform cluster in the Red Hat OpenShift documentation, and the Red Hat Customer Portal.

Cloud platforms

For more information about the storage providers that you can use on the following platforms, see Recommended storage providers.

- IBM Cloud®
- AWS Marketplace
- Microsoft® Azure

IBM Cloud

You can install Infrastructure Automation on IBM Cloud® with:

- Red Hat OpenShift
- Red Hat® OpenShift® on IBM Cloud® service (ROKS). For more information about ROKS, see the Red Hat OpenShift documentation Red Hat OpenShift on IBM Cloud.

For more information about installing Infrastructure Automation on ROKS, see Installing Infrastructure Automation on ROKS.

AWS Marketplace

You can install Infrastructure Automation on AWS with:

- Red Hat® OpenShift®
- Red Hat OpenShift Service on AWS (ROSA). For more information about ROSA, see the Red Hat documentation Red Hat OpenShift Service on AWS.

For more information about installing Infrastructure Automation on ROSA, see Installing Infrastructure Automation on ROSA.

Microsoft® Azure

You can install Infrastructure Automation on Azure with:

- Red Hat OpenShift
- Azure Red Hat OpenShift (ARO). For more information about ARO, see the Microsoft documentation Azure Red Hat OpenShift.

  For more information about installing Infrastructure Automation on (ARO), see Installing Infrastructure Automation on ARO.

Hardware requirements

You have the option to install IBM Infrastructure Automation (stand-alone) or install IBM Infrastructure Automation with IBM Cloud Pak for AIOps. For more information about requirements, see the following sections.

- Requirements for installing Infrastructure Automation with IBM Cloud Pak for AIOps
- Requirements for installing Infrastructure Automation (stand-alone)

Note: Your hardware must be able to support Red Hat OpenShift, Infrastructure Automation, and your chosen storage solution.

Requirements for installing Infrastructure Automation with IBM Cloud Pak for AIOps

If you are installing Infrastructure Automation with IBM Cloud Pak for AIOps, then you need to follow the hardware requirements for Infrastructure Automation and IBM Cloud Pak for AIOps.

For Cloud Pak for AIOps and Infrastructure Automation requirements, see the following sections:

- Hardware requirements - Red Hat OpenShift
- Hardware requirements Infrastructure Automation (stand-alone)

Requirements for installing Infrastructure Automation (stand-alone)

Use the following sections to review the details about the minimum resource requirements for installing Infrastructure Automation. Use this information for sizing your environment when you are deploying Infrastructure Automation. Multi-region and multi-zone clusters are not supported.

Note: Work with your IBM Sales representative (or Business Partner) to ensure that the allocated hardware resources for your environment are sufficient to best fit your business needs. For requirements for larger production deployments and to make sure that you meet the right level of resiliency, an IBM representative can work with you on your deployment plans. For Infrastructure Automation, you need to provide the following information:

- The number of cloud environments that you connect to.
- The number of OpenStack environments that you manage.
- The average number of VMs in each of the cloud environments.
- The number of on-premises virtualized environments that you manage. Identify whether it is VMware or KVM, and the number of VMs in each virtualized environment.

Resource allocation for the Red Hat OpenShift Container Platform control plane master nodes

The following table details the minimum requirements for the Red Hat OpenShift control plane master nodes that are necessary for an Red Hat OpenShift cluster to withstand the demands of an Infrastructure Automation deployment. Infrastructure Automation has many Kubernetes operators that interface with the API server and `etcd` storage, and so master nodes must be adequately sized.

Node count	vCPU (per node)	Memory (GB) (per node)	vCPU (all nodes)	Memory (GB) (all nodes)
3	8	16	24	48

Resource allocation for the Red Hat OpenShift Container Platform worker nodes

The following table provides the required Red Hat OpenShift worker resources per node for a sample Infrastructure Automation deployment.

Node count	vCPU (per node)	Memory (GB) (per node)	vCPU (all nodes)	Memory (GB) (all nodes)	Disk space (GB)	vCPU (extra capacity)	Memory (GB) (extra capacity)
3	16	32	48	96	212	17	18

The following table lists the hardware requirements for Infrastructure Automation only, and does not include the requirements for the Red Hat OpenShift control plane.

Resource	Deployment (Managed Services and Infrastructure management) No infrastructure management providers (not under load)	Deployment (Managed Services and Infrastructure management) 1 infrastructure management provider (not under load)
Worker node count	3	3
Total vCPUs	14	14
Total memory (GB)	35	38
Persistent storage (Gi)	70	70

Notes:

- These estimates are for an Infrastructure Automation installation with both Managed services and Infrastructure Management deployed.

- The requirements for Infrastructure Automation are in addition to the requirements for IBM Cloud Pak for AIOps.
- You might need extra worker nodes, resource capacity, or both, depending on your own requirements. The CPU, memory, and disk resources that are required for Infrastructure Automation depend on the number of provider instances and the total number of instances and VMs managed by the providers. Other worker node counts and sizes can be used based on your available hardware configurations, but the following conditions must be met:
  - The combined available vCPU and memory of the worker nodes must equal or exceed the total requirements.
  - Each worker node has at least the minimum vCPU and memory that is given in the preceding table.
  - **Warning**: These requirements do not include CPU and memory resources for hosting a storage provider. These storage providers can require more nodes, resources, or both, to run. The extra resources that are needed can vary based on your selected storage provider. A general recommendation is for one extra worker node in a starter deployment, and for three extra worker nodes for a production deployment where HA is required. Consult your storage provider's documentation for exact requirements.

Storage requirements

Learn about the storage requirements for Infrastructure Automation.

- Persistent storage requirements
- Recommended storage providers

Persistent storage requirements

Infrastructure Automation requires persistent storage that supports the RWO (read-write-once) access mode.

Storage class requirements

For production environments, storage classes must have `allowVolumeExpansion` enabled. This allows persistent volumes to be expanded if necessary, to avoid storage from filling up and causing unrecoverable failures. This is also highly recommended for starter deployments, since without it you are limited to the default capacity that might not be sufficient for your specific needs. To enable `allowVolumeExpansion`, edit the storage class to enable expansion. Follow the instructions in the Red Hat documentation Enabling volume expansion support.

Recommended storage providers

The recommended storage providers are:

- IBM Cloud
- Red Hat OpenShift Data Foundation (ODF)
- IBM Fusion Data Foundation
- IBM Fusion Global Data Platform
- IBM Storage Scale Container Native
- Portworx
- AWS native storage

The following table shows the tested and supported storage providers for the platforms where Infrastructure Automation can be deployed.

Table 2. Supported storage providers by platform

Platform	IBM Cloud Storage (Block and File)	Red Hat® OpenShift® Data Foundation	IBM Fusion Data Foundation	IBM Fusion Global Data Platform	IBM Storage Scale Container Native	Portworx	AWS native storage
Azure Red Hat OpenShift (ARO)		Yes				Yes	
Red Hat OpenShift Container Platform		Yes	Yes	Yes	Yes	Yes	Yes
Red Hat OpenShift on IBM Cloud (ROKS)	Yes	Yes	Yes	Yes	Yes		
Red Hat OpenShift Service on AWS (ROSA)						Yes	Yes

Note:

- IBM Storage Scale Container Native and Red Hat® OpenShift® Data Foundation are part of IBM Storage Suite for IBM Cloud Paks.

The preceding storage providers are the only providers that are tested and validated for a deployment of Infrastructure Automation. You can choose to use an alternate storage provider if they meet the requirements for deploying Infrastructure Automation. Your chosen storage provider must meet the same storage and hardware requirements as the recommended storage providers. If you choose to use an alternate storage provider, your overall performance can differ from any listed sizings, throughput rates or other performance metrics that are listed in the Infrastructure Automation documentation. Work with your IBM Sales representative (or Business Partner) to ensure that your chosen storage provider is sufficient for your deployment plan.

If you are installing Infrastructure Automation with Cloud Pak for AIOps, see Storage requirements for installation on OpenShift

IBM Cloud

The following storage classes are required by Infrastructure Automation, and are created when Red Hat OpenShift on IBM Cloud (ROKS) is installed:

Storage provider	Storage class name	Large block storage class name
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold

Installing IBM Cloud storage

For more information about IBM Cloud storage, see Storing data on classic IBM Cloud File Storage and Storing data on classic IBM Cloud Block Storage in the IBM Cloud Docs.

Red Hat OpenShift Data Foundation (ODF)

The following storage classes are required by Infrastructure Automation, and are created when Red Hat OpenShift Data Foundation is installed:

Storage provider	Storage class name	Large block storage class name
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd

Installing Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation is available for purchase through the IBM Storage Suite for IBM Cloud Paks. Red Hat OpenShift Data Foundation is an implementation of the open source Ceph storage software, which is engineered to provide data and storage services on Red Hat OpenShift. Use version 4.12 or higher.

For more information about deploying Red Hat OpenShift Data Foundation, see Deploying OpenShift Data Foundation in the Red Hat documentation. Choose the appropriate deployment instructions for your deployment platform.

IBM Fusion Data Foundation

The following storage classes are required by IBM Cloud Pak for AIOps:

Storage provider	Storage class name	Large block storage class name
IBM Fusion Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd

Installing IBM Fusion Data Foundation

IBM Fusion Software version 2.9.0 and IBM Fusion HCI Systems version 2.9.x are compatible with Infrastructure Automation.

To learn more about IBM Fusion and how to install IBM Fusion Data Foundation service, see the following two methods:

IBM Fusion Software

IBM Fusion Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as part of the Red Hat OpenShift Container Platform service catalog, packaged as an operator to facilitate simple deployment and management.

The IBM Fusion Software provides the following features:

- IBM Fusion software is a software-only solution that can be deployed on a variety of hardware platforms.
- IBM Fusion software is designed to scale up to meet the needs of larger environments.
- IBM Fusion software can be integrated with a variety of third-party solutions.

Before you install IBM Fusion, ensure that you meet all of the prerequisites. For more information, see Prerequisites.

For more information about Deploying IBM Fusion, see Deploying IBM Fusion.

For more information about how to install Data Foundation service using IBM Fusion, see Data Foundation.

For more information about IBM Fusion Software storage class, see Data Foundation.

IBM Fusion HCI Systems

The IBM Fusion Data Foundation service provides a foundational data layer for applications to function and interact with data in a simplified, consistent, and scalable manner.

The IBM Fusion HCI Systems provides the following features:

- IBM Fusion HCI system is a pre-integrated, pre-tested, and pre-configured appliance that combines hardware and software.
- IBM Fusion HCI system is designed to scale out to meet growing storage needs.
- IBM Fusion HCI system is tightly integrated with IBM's other hybrid cloud solutions, such as IBM Cloud and IBM Power Systems.

For more information about Prerequisites of IBM Fusion HCI system, see Planning and prerequisites.

For more information about Deploying IBM Fusion HCI System, see Deploying IBM Fusion HCI System.

For more information about how to install Data Foundation service using IBM Fusion HCI Systems, see Data Foundation.

For more information about IBM Fusion HCI Systems storage class, see Data Foundation.

IBM Fusion Global Data Platform

The following storage classes are required by Infrastructure Automation:

Storage provider	Storage class name	Large block storage class name
IBM Fusion Global Data Platform	If you are using IBM Fusion, use ibm-spectrum-scale-sc. If you are using IBM Fusion HCI System, use ibm-storage-fusion-cp-sc	If you are using IBM Fusion, use ibm-spectrum-scale-sc. If you are using IBM Fusion HCI System, use ibm-storage-fusion-cp-sc

Installing IBM Fusion Global Data Platform

The Global Data Platform storage type provides the following features:

- File storage
- High availability via capacity-efficient erasure coding
- Metro and regional disaster recovery
- CSI snapshot support with built-in application consistency
- Encryption at rest
- Ability to mount file systems hosted by remote IBM Storage Scale clusters.

You can deploy Global Data Platform service by using the following two methods:

IBM Fusion Software

Before you install IBM Fusion, ensure that you meet all of the prerequisites. For more information, see Prerequisites.

For more information about Deploying IBM Fusion, see Deploying IBM Fusion.

For more information about how to install Global Data Platform using IBM Fusion Software, see Global Data Platform.

For more information about IBM Fusion Software storage class, see IBM Storage Scale.

IBM Fusion HCI Systems

For more information about Prerequisites of IBM Fusion HCI system, see Planning and prerequisites.

For more information about Deploying IBM Fusion HCI System, see Deploying IBM Fusion HCI System.

For more information about how to deploy Global Data Platform service using IBM Fusion HCI Systems, see Global Data Platform.

For more information about IBM Fusion HCI Systems storage class, see IBM Storage Scale.

IBM Storage Scale Container Native

The following storage class is required by Infrastructure Automation, and is created when IBM Storage Scale Container Native is installed.

Storage provider	Storage class name	Large block storage class name
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc

This class is used as both the ReadWriteMany storage class and large block storage class when you are installing Infrastructure Automation. This StorageClass includes a parameter to set permissions of data within the StorageClass to **`shared: true`**, which is required to support the Kubernetes SubPath feature. For more information about the permissions field for the class, see the IBM Spectrum Scale CSI Driver documentation.

Installing IBM Storage Scale Container Native

IBM Storage Scale Container Native is available for purchase through the IBM Storage Suite for IBM Cloud Paks. IBM Spectrum Scale is a cluster file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network attached, a mixture of SAN-attached and network attached, or in a shared nothing cluster configuration. This enables high-performance access to this common set of data to support a scale-out solution or to provide a high availability platform. IBM Storage Scale Container Native Storage Access must be at version 5.1.1.3 or higher, with IBM Spectrum Scale Container Storage Interface version 2.3.0 or higher.

For more information, see the IBM Storage Scale Container Native documentation.

To use IBM Storage Scale Container Native with Infrastructure Automation, you do not require a separate license. You can use up to 12 TB of IBM Storage Scale Container Native storage for up to 36 months, fully supported by IBM, within your production environments (Level 1 and Level 2). If you exceed these terms, a separate license is required.

To install and use IBM Storage Scale Container Native, your cluster must meet the following requirements:

- Minimum amount of storage: 1 TB or more of available space.
- Minimum amount of vCPU: 8 vCPU on each worker node. This minimum is a general requirement for the node, not a dedicated resource requirement for IBM Storage Scale Container Native. For more information, see the IBM Spectrum Scale requirements documentation about IBM Storage Scale Container Native.
- Minimum amount of memory: 16 GB of RAM on each worker node. This minimum is a general requirement for the node, not a dedicated resource requirement for IBM Storage Scale Container Native. For more information, see the IBM Spectrum Scale requirements documentation.
- In addition, your network must have sufficient performance to meet the storage I/O requirements. When I/O performance is not sufficient, services can experience poor performance or cluster instability when the services are handling a heavy load, such as functional failures with timeouts.Run a disk latency test and disk throughput test to verify whether your network meets the I/O requirements. You can use the DD command-line utility to run the following tests. These tests are based on the performance of writing data to representative storage locations by using two chosen block sizes (4 KB and 1 GB). Use the MB/s metric from the tests and ensure that your test result is comparable to, or better than, the expected targets:
    - Disk latency test

      ```
      dd if=/dev/input.file of=/directory_path/output.file bs=block-size count=number-of-blocks oflag=dsync
      ```

The value must be comparable to or better than: 2.5 MB/s.

- Disk throughput test

  ```
  dd if=/dev/input.file of=/directory_path/output.file bs=block-size count=number-of-blocks oflag=dsync
  ```

  The value must be comparable to or better than: 209 MB/s.

Where,

- `if=/dev/input.file` - Identifies the input file that you want to read using the DD command.
- `of=/directory_path/output.file` - Identifies the output file where you want to write with the DD command.
- `bs=block-size` - Indicates the block size that you want to use. For example, you can use 4096 (4 KB) for the latency test and 1G for the throughput test.
- `count=number-of-blocks` - The number of blocks to use in the test. For example, you can use 1000 for the latency test and 1 for the throughput test.
- `oflag=dsync` - Indicates the synchronization I/O for data.

To install IBM Storage Scale Container Native and the IBM Spectrum Scale Container Storage Interface, follow the IBM Storage Scale Container Native installation documentation.

If you encounter any errors or issues with installing or using IBM Storage Scale Container Native or the IBM Spectrum Scale Container Storage Interface, refer to the following documentation:

- IBM Storage Scale Container Native
- IBM Spectrum Scale Container Storage Interface

Portworx

The following storage classes are required by Infrastructure Automation, and must be created using the instructions in the following section:

Note: The `px-csi-aiops-mz` storage class is only applicable if you are installing Infrastructure Automation with Cloud Pak for AIOps. To use this storage class, you need to use multizone HADR for Cloud Pak for AIOps.

Storage provider	Storage class name	Large block storage class name
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz

Installing Portworx

Important: `sharedv4` volumes require NFS ports to be open. For more information, see the Portworx documentation Open NFS Ports.

1. Install the Portworx operator and configure a Portworx StorageCluster.

   Two editions of Portworx are available: Portworx Enterprise and Portworx Essentials. Portworx Enterprise is suitable for production deployments. Portworx Essentials is suitable only for demonstration deployments, as it can be used only on clusters of five nodes or less, and includes storage size limits. For more information, see the Portworx documentation Installing Portworx on Openshift.

   Note: You must be a cluster administrator to install Portworx on the cluster.

2. Define a custom Portworx storage class.

   The custom Portworx storage class will be used for file and block storage. The storage class is scoped to the cluster, so setting the project (namespace) is not required.

   If you are not using the Multi-zone HA technical preview, then use Option 1 to define `px-csi-aiops`. If you are using the Multi-zone HA technical preview, then use Option 2 to define `px-csi-aiops-mz`.

   Option 1: Define `px-csi-aiops`

   Log in to your OpenShift cluster's console. Click the plus icon on the upper right to open the **Import YAML** dialog box, paste in the following content, and then click **Create**.

   ```
   kind: StorageClass
   apiVersion: storage.k8s.io/v1
   metadata:
     name: px-csi-aiops
   provisioner: pxd.portworx.com
   parameters:
     fs: xfs
     io_profile: db_remote
     repl: '2'
   reclaimPolicy: Delete
   allowVolumeExpansion: true
   volumeBindingMode: Immediate
   ```

   Option 2: Define `px-csi-aiops-mz`

   Log in to your OpenShift cluster's console. Click the plus icon on the upper right to open the **Import YAML** dialog box, paste in the following content, and then click **Create**.

   ```
   kind: StorageClass
   apiVersion: storage.k8s.io/v1
   metadata:
   ```

```
  name: px-csi-aiops-mz
provisioner: pxd.portworx.com
parameters:
  fs: xfs
  io_profile: db_remote
  repl: '3'
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

AWS native storage

The following storage classes are required by Infrastructure Automation. **gp3-csi** is created for you, but you must create **efs-sc** using the instructions in the following section.

Storage provider	Storage class name	Large block storage class name
AWS native storage	efs-sc	gp3-csi

Amazon Elastic Block Store (EBS) provides block storage - the storage class is **gp3-csi**, and is created when ROSA is installed. Amazon Elastic File System (EFS) provides file storage - the storage class is **efs-sc**. You must create the **efs-sc** storage class using the instructions in the following section.

Installing AWS native storage

For more information about Amazon Web Services (AWS) native storage, see Cloud Storage on AWS in the AWS documentation.

For more information, see **Setting up the AWS EFS CSI Driver Operator** and **Creating the AWS EFS storage class** in the Red Hat OpenShift documentation. You can also review the information in the Red Hat article Enabling the AWS EFS CSI Driver Operator on ROSA.

High Availability (HA) considerations

IBM Cloud Pak for AIOps

For more information about high availability (HA) considerations for your IBM Cloud Pak® for AIOps deployment, see High Availability (HA) considerations.

Infrastructure Automation

If you are installing Infrastructure Automation for use with IBM Cloud Pak for AIOps, you can set up high availability for your IBM Cloud Pak for AIOps deployment by selecting a production (large) sizing profile.

If you are installing Infrastructure Automation as a stand-alone deployment and you are using Infrastructure Management appliances, you can configure different settings to help you set up high availability. For more information, see High Availability Guide

Security considerations

IBM Infrastructure Automation supports different mechanisms for securing your environment and your data.

View the following sections to increase the security of your Infrastructure Automation environment.

Security context constraints (SCC)

Administrators can use SCCs to control permissions for pods on their Red Hat OpenShift Container Platform cluster. These permissions include actions that a pod can perform and what resources it can access. Red Hat OpenShift Container Platform clusters contain default security context constraints (SCCs). IBM Cloud Pak for AIOps runs with the **restricted** or **restricted-v2** SCC that is included within this default set of constraints. For more information, see Red Hat® OpenShift® Container Platform SCCs and Red Hat - Managing Security Context Constraints.

Note: SecurityContextConstraints do not apply to the **default** or **kube-system** namespaces.

Permissions

For more information about the permissions that are required for Infrastructure Automation, see Permissions.

Volume level encryption

You can configure Portworx encryption or host-level encryption, depending on your requirements. Decide which level of encryption is needed for each stateful data store in your environment.

- For more information about using encrypted volumes on Red Hat OpenShift Container Storage, see Creating a storage class for persistent volume encryption.
- For more information about using the encryption feature for Portworx volumes, see Create encrypted PVCs.

Security hardening

For production environments, ensure that the Red Hat OpenShift Container Platform cluster where you are installing Infrastructure Automation is configured securely. To help harden the security for your cluster, consider completing the following tasks:

- Configure the Red Hat OpenShift Container Platform to enable etcd data encryption. For more information, see the Red Hat documentation about Encrypting etcd data.
- Implement a network egress policy that strictly controls what outbound calls can be made from applications in the cluster and from the Infrastructure Automation project (namespace). For more information, see the Red Hat documentation about Creating a network policy.
- If you use your own certificate authority (CA), replace the default certificate manager CA with your own CA.
- Ensure that your Red Hat OpenShift Container Platform and Kubernetes environment is configured according to the Center for Internet Security (CIS) Benchmark for Red Hat Enterprise Linux based OpenShift V4 and Kubernetes V1.6 security hardening. For more information, see the Center for Internet Security (CIS) Benchmarks about Securing Kubernetes.
- Configure lockout mechanisms on your authentication server to handle excessive login attempts.
- Shorten session timeouts to the shortest tolerable duration.
- Use your own certificate authority and monitor certificate expiration dates.
- Configure rate limiting connections for basic protection against distributed denial-of-service (DDoS) attacks. For more information about how to configure this limiting, see the Red Hat documentation about Route configuration.
- Limit users from viewing information about other users. By default, user IDs, user groups, roles, and permissions for other users are visible to users. To block non-administrative users from viewing user groups, roles, and permissions,

  1. Edit the config map **`product-configmap`** to add the configuration **`usermgmt_limit_user_info: "true"`**.
  2. Restart the user management pods, by running the following command:

     ```
     oc delete pod -l component=usermgmt
     ```

Protecting against HTTP Host header attacks

The mandatory request header specifies the domain that you want to access can be altered as part of an exploitative attack. If your installation does not use a load balancer or proxy that depends on http **`HOST`** headers, then use the following steps to enable HTTP Host header injection protection:

1. Ensure that the HTTP Host header matches the value of **URL_PREFIX** from the ConfigMap **product-configmap**. If necessary, specify the project (namespace) of your Infrastructure Automation deployment.
2. In configmap **`product-configmap`**, add or edit the **`HOST_INJECTION_CHECK_ENABLED`** entry and set its value to **`true`**. To set the entry by using the Red Hat OpenShift, select the **product-configmap** in your Infrastructure Automation project (namespace). From the menu, select **Edit configmap**. On the YAML page, find or add the **`HOST_INJECTION_CHECK_ENABLED`** entry, and set the value to **`true`**.
3. Restart the **`ibm-nginx-*`** pods.

   - If you are using the Red Hat OpenShift console, select the pods and use the delete pod action.
   - If you are using the CLI, use the **`oc delete <pod-name> -n <namespace>`** command.

     Where **`<namespace>`** is the project (namespace) of your Infrastructure Automation deployment.

If you need to disable this protection for any reason, remove the **`HOST_INJECTION_CHECK_ENABLED`** entry, or set its value to **`false`**, and restart the Nginx pods.

Certificates

If you want to use a custom certificate for Infrastructure Automation, then you can do this with either of the following methods:

- Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
- If you would like to use a custom certificate for IBM Cloud Pak for AIOps or Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

Licensing

Review the following to learn about the licensing and entitlements for IBM Infrastructure Automation 4.8.1.

Notice: This License Guide is intended to provide only supplementary information to assist you in deploying the Programs you have licensed from IBM within your purchased entitlement. Your license agreement (such as the IBM International Program License Agreement (IPLA) or equivalent and its transaction documents, including the License Information for Infrastructure Automation 4.8.1), is the sole and complete agreement between you and IBM regarding use of the Program.

- Viewing the license
- What's included in your purchaseof Infrastructure Automation IBM Infrastructure Automation

- Red Hat Products
- Entitlement details
- License reporting

Viewing the license

To review the license agreement for Infrastructure Automation 4.8.1 to find more information about licensing and entitlements for this offering, see L-BBAE-L53CM5.

What's included in your purchase of IBM Infrastructure Automation 4.8.1

Your purchase of IBM Infrastructure Automation includes the following entitlements:

- Bundled Programs
- Supporting Programs
- Non-Production features
- Prohibited Components

Evaluation Period

The evaluation period starts upon agreement and lasts for 60 days.

Bundled Programs

As a licensee, you are authorized to install and use the followinig Bundled Programs:

- IBM Netcool Operations Insight Operations Management
- IBM Netcool Operations Insight Connection
- IBM Operations Analytics Predictive Insights
- IBM Robotic Process Automation Platform
- IBM Process Mining
- Monitoring

Supporting Programs

As a licensee, you are authorized to install and use the following supporting programs:

- IBM Netcool Operations Insight Agile Service Manager
- IBM Cloud Pak Foundational Services
- IBM Fusion
- IBM Cognos Analytics User Cartridge for IBM Cloud Pak for Data
- IBM Cognos Analytics Administrator Cartridge for IBM Cloud Pak for Data

Non-Production

As a licensee, the following components are limited for internal development and test environments for internal non-production activities ("Non-Production"):

- Story prioritization from persistence analysis and golden signals extraction
- Deploy to single ROSA cluster configured across multiple AWS Zones
- Multi-zone high availability and disaster recovery (HADR)
- IBM Cognos Analytics Integration for AIOps Custom Dashboards and Reporting

Prohibited Components

As a licensee, you are not authorized to use any of the following components or functions of the Program:

- Application Discovery component (of IBM Netcool Operations Insight Agile Service Manager)
- Network Discovery component (of IBM Netcool Operations Insight Agile Service Manager)
- Attended Bot Agent add-on (of IBM Robotic Process Automation)
- ABBYY FineReader OCR Provider (of IBM Robotic Process Automation)
- Business Process Model and Notation Export/Import Capability (of IBM Process Mining)
- Decision Model and Notation Export/Import Capability (of IBM Process Mining)

Red Hat Products

As a licensee, your purchase of IBM Infrastructure Automation 4.8.1 includes restricted entitlements for the following Red Hat products in the quantities and/or ratios set out in your license agreement:

- Red Hat OpenShift Container Platform

Entitlement details

Various metrics and conversion ratios apply to your entitlement of Infrastructure Automation 4.8.1. These metrics and conversion ratios include, but not limited to, the following:

- Managed Client Device
- Managed Device
- Process

- Resource Unit
- Virtual Processor Core
- Virtual Server

The following table lists some of the entitlements per resource unit (perpetual) for IBM Infrastructure Automation:

Table. Entitlement per resource units

Event Processing & Analytics	Metric Processing & Analytics	Log Processing & Analytics
- 1 Managed Device = 1 resource - 20 Client Devices = 1 resource - 10,000 Monthly Events = 7 resource	- 1 Managed Device = 1 resource - 20 Client Devices = 1 resource	• 50GB Monthly logs = 7 resource

License reporting

As part of using Infrastructure Automation, tracking your license usage is required. Although Infrastructure Automation provides license options not tracked directly by the product, to track usage, you must install and deploy the IBM Cloud Pak foundational services License Service on all clusters where Infrastructure Automation is installed.

IBM Cloud Pak foundational services License Service

This background License Service collects and stores the license usage information for tracking license consumption and for audit purposes. You can retrieve this data upon request for monitoring and compliance. You can also retrieve an audit snapshot of the data that is audit evidence.

Note: For Infrastructure Automation, the license service collects usage data for only Infrastructure Automation. Although Infrastructure Automation provides license options not tracked directly by the product, to track usage, you must deploy the Infrastructure Automation License Service on all clusters where Infrastructure Automation is installed.

Refer to the IBM Cloud Pak foundational services documentation to learn about how License Service collects and measures the license usage of IBM Infrastructure Automation, and how to retrieve the license usage data.

- License Service
- License Service Reporter

Passport Advantage part numbers

Use the following information to download the packages for Infrastructure Automation.

Downloading packages

Complete the following procedure to download your packages:

1. Go to the Passport Advantage Online tab at Passport Advantage and log in with your IBM ID.
2. Find your part number.
3. Search for your files by using the part numbers that are listed in the tables.
4. Download the files to a directory on your computer.

For more information about downloading IBM Cloud Pak for AIOps and supported products, see Passport Advantage part numbers.

Infrastructure Automation packages

The following table lists the eAssembly number for Infrastructure Automation - Infrastructure Management.

Table. Software eAssembly part numbers for Infrastructure Automation

eAssembly #	Description
G0DB2EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure Automation eAssembly

The following table lists the eImage numbers for Infrastructure Automation - Infrastructure Management.

Table. Software eImage part number for Infrastructure Automation

eImage #	Description
M0PTTEN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for VMware vSphere with PVSCSI controller
M0PTVEN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Red Hat OpenStack Platform
M0PTWEN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Amazon EC2
M0PTXEN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Red Hat Virtualization
M0PTYEN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation RPM updates
M0PTZEN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Google Compute Engine

Installing Infrastructure Automation

You have the option to install stand-alone Infrastructure Automation or install Infrastructure Automation for use with Cloud Pak for AIOps. Follow these steps to install Infrastructure Automation.

If you have a license for IBM Cloud Pak® for AIOps, you are entitled to install and use Infrastructure Automation.

Choose one of the following deployment options:

- Installing for use with Cloud Pak for AIOps online (console)
- Installing for use with Cloud Pak for AIOps online (CLI)
- Installing stand-alone Infrastructure Automation online (console)
- Installing stand-alone Infrastructure Automation online (CLI)
- Installing on cloud platforms
  - Installing on ROKS
  - Installing on ROSA
  - Installing on ARO
- Offline
  - Bastion host
  - Portable device
- Installing Infrastructure Management
  - Installing on AWS (appliance)
  - Installing on VMware vSphere (appliance)
  - Installing on Red Hat OpenStack Platform (appliance)
  - Installing on Red Hat Virtualization (appliance)
  - OIDC Configuration steps for Infrastructure Management with Cloud Pak (appliance)
  - Installing as a containerized deployment with Cloud Pak (podified)

Online installation of Infrastructure Automation for use with IBM Cloud Pak for AIOps (console)

Follow these steps to use the Red Hat® OpenShift® Container Platform console to complete an online installation of Infrastructure Automation, for use with IBM Cloud Pak® for AIOps. The online installation is deployed in Red Hat® OpenShift® Container Platform.

If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation. Install IBM Cloud Pak for AIOps before you install Infrastructure Automation.

Before you begin

- Review the Planning section.
- Check that your system meets the minimum requirements for installing IBM Cloud Pak for AIOps and Infrastructure Automation. **Important:** The `storageClass` and `storageClassLargeBlock` that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the `storageClass` and `storageClassLargeBlock` that are used for creating the Infrastructure Automation custom resource.
- If you require details about the permissions that the Infrastructure Automation operators need, see Permissions (Infrastructure Automation).
- A user with `cluster-admin` privileges is needed for the following operations:
  - Install IBM Cloud Pak for AIOps
  - Create the catalog source
  - Install Cert Manager
  - Install the License Service
  - Install the Infrastructure Automation operator

Installation steps

Follow these steps to install Infrastructure Automation.

1. Install IBM Cloud Pak for AIOps
2. Create the catalog source
3. Install Cert Manager
4. Install the License Service
5. Install the Infrastructure Automation operator
6. Install Infrastructure Automation
7. Verify the deployment
8. Log in to the Infrastructure Automation console
9. Deploying Infrastructure Management
10. Assign user roles and permissions
11. Enable usage data collection (optional)

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`** and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install IBM Cloud Pak for AIOps

It is recommended that you install IBM Cloud Pak for AIOps before you install Infrastructure Automation. IBM Cloud Pak for AIOps and Infrastructure Automation can be deployed on the same Red Hat OpenShift cluster.

Important: IBM Cloud Pak for AIOps and Infrastructure Automation must be installed in the same namespace. Installing IBM Cloud Pak for AIOps and Infrastructure Automation in separate namespaces is not supported.

To install IBM Cloud Pak for AIOps, follow the instructions in Installing IBM Cloud Pak for AIOps.

2. Create the catalog source

Add the Infrastructure Automation catalog source to your Red Hat OpenShift cluster.

After installation, the **`ibm-operator-catalog`** CatalogSource object determines whether the upgrade of your Infrastructure Automation deployment is initiated automatically when a new patch becomes available. The **`ibm-operator-catalog`** CatalogSource object can be configured to automatically poll for and retrieve a newer catalog by enabling the polling attribute **`spec.updateStrategy.registryPoll`**. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your Infrastructure Automation deployment is initiated. For more information, see Upgrading Infrastructure Automation.

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

Note: **`ibm-operator-catalog`** also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

Run the steps in Create the catalog source with automatic upgrade disabled or Create the catalog source with automatic upgrade enabled.

Create the catalog source with automatic upgrade disabled

1. Create the **`ibm-operator-catalog`** CatalogSource object without polling enabled.

   Log in to your Red Hat OpenShift cluster's console. Click the plus icon in the upper right to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
   ```

2. Update the **`ibm-operator-catalog`** CatalogSource to fix it to always use the current image digest, instead of **`icr.io/cpopen/ibm-operator-catalog:latest`**. This ensures that the **`ibm-operator-catalog`** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart.

   1. Go to **Home** > **Projects**, and select **`openshift-marketplace`**.
   2. Go to **Workloads** > **Pods** (on the left menu), and then search for **`ibm-operator-catalog`**.
   3. Click the returned **`ibm-operator-catalog-<...>`** pod.
   4. Click **YAML** to switch to the YAML view.
   5. Search for **`imageID`** in the YAML, and copy down the value of **`spec.containerStatuses.imageID`**. The value is in a format similar to the following example:

      ```
      icr.io/cpopen/ibm-operator-catalog@sha256:<...>
      ```

   6. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, scroll down and click **`ibm-operator-catalog`**.
   7. Click **YAML** to switch to the YAML view.

8. Set the value of **`spec.image`** to the value of the current image digest that you found in step 2, instead of to **`icr.io/cpopen/ibm-operator-catalog:latest`**.

3. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-operator-catalog* CatalogSource object is present.

Create the catalog source with automatic upgrade enabled

1. Create the **`ibm-operator-catalog`** CatalogSource object with polling enabled.

   Log in to your Red Hat OpenShift cluster's console. Click the plus icon in the upper right corner to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

2. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-operator-catalog* CatalogSource object is present.

3. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed a certificate manager as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters, then you must ensure that you have a certificate manager on each cluster.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Log in to your Red Hat OpenShift cluster's console.
2. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.
3. In the **All Items** field, enter **`IBM Cert Manager`**. The **IBM Cert Manager** operator is displayed.
4. Click the **IBM Cert Manager** tile. The **IBM Cert Manager** window is displayed.
5. Click **Install**. You see the **Install Operator** page.
6. Set the **Update Channel** to the **`v4.2`** version. If the Channel `v4.2`` version is not available, click other **IBM Cert Manager** tile from OperatorHub to install the correct version.
7. Set **Installation Mode** to **`All namespaces on the cluster (default)`**.
8. Set **Installed Namespace** to **`ibm-cert-manager(Operator recommended)`**.
9. Set **Update approval** to **`Automatic`**.
10. Click **Install**.

4. Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not know whether the License Service is already installed, then see Verifying if License Service is already installed on the cluster in the IBM Cloud Pak foundational services documentation.

IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

Follow the instructions in Installing the License Service with OpenShift console in the IBM Cloud Pak foundational services documentation, from step 2 **`Create the ibm-licensing namespace`** onwards.

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5. Install the Infrastructure Automation operator

For more information about operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

1. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.
2. In the **All Items** field, enter **`IBM Infrastructure Automation`**. The IBM Infrastructure Automation operator is displayed.
3. Click the **IBM Infrastructure Automation** tile. The IBM Infrastructure Automation window is displayed.
4. Click **Install**. The *Install Operator* page is displayed.
5. Enter the following values:
   - Set **Update channel** to *v4.8*.
   - **Installation mode** - For more information about installation modes, see Operator installation mode.
   - **Installed Namespace** - If you are using the **`OwnNamespace`** installation mode (a specific namespace), then set this field to be the project (namespace) in which to install the operator. If you are using the **`AllNamespaces`** installation mode, then set this field to **`openshift-operators`**.
   - Set **Update approval** to *Automatic*.

   Warning: Update approval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.
6. Click **Install** and wait for the IBM Infrastructure Automation operator to install.
7. Verify that the IBM Infrastructure Automation operator is successfully installed.

   Navigate to **Operators** > **Installed Operators**, and select your project from the **Projects** dropdown. IBM Infrastructure Automation operator and its dependant operators in the project are listed with a **Status** of *Succeeded*.

6. Create Infrastructure Automation custom resource with default values

You can create the Infrastructure Automation custom resource with the default set of values, or customize these values before you create the Infrastructure Automation custom resource.

Customize the default values for the Infrastructure Automation custom resource to modify the default replica count of the pods in Managed services, or to adjust any of the following defaults for Managed services:

- Default pre-created secrets in Managed services
- DefaultManaged services proxy setting
- Default Managed services default github.com token to import templates
- Default HTTPS certificate for Managed services Management Console
- Configuring Managed services to set the SSH keys and known hosts in provider terraform pod

The following YAML file creates an instance of the Infrastructure Automation custom resource called **`IAConfig`**. The list of installation parameters with its default values are listed in Managed services installation parameters page.

1. Navigate to **Operators** > **Installed Operators**.
2. Under the *Provided APIs* section, click **IBM Infrastructure Automation**.
3. Click **Create IAConfig** tab.
4. Switch to the YAML view and paste the following file:

```
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
name: ibm-ia-installer
namespace: cp4aiops
spec:
imagePullSecret: ibm-entitlement-key
infraAutoComposableComponents:
- enabled: <set to true to install Infrastructure Management component of Infrastructure Automation, false otherwise>
   name: ibm-management-im-install
   spec: {}
- enabled: <set to true to install Managed services component of Infrastructure Automation, false otherwise>
   name: ibm-management-cam-install
   spec: {}
license:
   accept: <set true to accept the license>
storageClass: <set to the same storageClass that IBM Cloud Pak for AIOps uses>
storageClassLargeBlock: <set to the same storageClassLargeBlock that IBM Cloud Pak for AIOps uses>
```

   To customize Managed services, you must have your customization parameters under **`spec.manageservice`** section.

```
- enabled: <Set to true to install Managed services component of Infrastructure Automation, false otherwise>
  name: ibm-management-cam-install
  spec:
    manageservice:
      <Set your custom installation parameter values>
```

5. Click **Apply**.

   Note: This option does not install Infrastructure Management. If you need to install Infrastructure Management, complete the installation of Infrastructure Automation and follow the steps that are listed in Deploying Infrastructure Management..

7. Verify the deployment

After a few minutes, use the following steps to check the status of your installation.

1. Click **Operators** > **Installed Operators**.
2. From the **Project** list, select the project (namespace) where Infrastructure Automation is deployed.
3. Click **IBM Infrastructure Automation**, then click **IBM Infrastructure Automation** tab.
4. Under **IAConfigs**, look at the entry with the name that you specified in the IAConfigs custom resource (Infrastructure Automation instance), and verify that the **Status** indicates *Phase: Running*.

8. Log in to the Infrastructure Automation console

After you successfully install Infrastructure Automation, get the URL for accessing the Infrastructure Automation console.

You can use the **Launch Cloud Pak in IBM Automation** link to access the Infrastructure Automation console:

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator.
2. Click **Operators** > **Installed Operators**.
3. Click **IBM Cloud Pak for AIOps**.
4. On the *Operator Details* page, click the **IBM Cloud Pak for AIOps** tab, and then click the IBM Cloud Pak for AIOps installation name.
5. In the **Details** tab, right-click on the URL underneath **Launch Cloud Pak in IBM Automation**, and select **Open Link in New Tab**.
6. In the Cloud Pak for AIOps console login page, select one of the following login options:
   - **OpenShift authentication**: The `kubeadmin` user is automatically used to log in to the Cloud Pak for AIOps console. The `kubeadmin` user has the same privileges as the Cloud Pak for AIOps console admin user.
   - **IBM provided credentials (admin only)**: The default username to access the console is `admin`. To obtain the username and password, see Obtain IBM provided credentials (admin only).
   - **Enterprise LDAP**: LDAP users can log in to the Cloud Pak for AIOps console after IBM Cloud Pak for AIOps is configured with a single or multiple LDAP servers for the authentication and authorization. For more information, see Identity Management (IM).

Obtain IBM provided credentials (admin only)

1. To find the default username, select the project (namespace) that IBM Cloud Pak for AIOps is deployed, then navigate to **Workloads** > **Secrets**. Search the `platform-auth-idp-credentials` secret name in the search bar. Click `platform-auth-idp-credentials` to view the secret. You can see the value of username from the `admin_username` field.
2. To get the password for the *admin* username, see the `admin_password` field in `platform-auth-idp-credentials` secret.

   The following is a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be `EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA`.

   Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

9. Deploying Infrastructure Management

For more information about how to deploy Infrastructure Management, see the following topics:

- Installing Infrastructure management as a containerized deployment (podified)
- Installing Infrastructure management as a virtual machine appliance

10. Assign user roles and permissions

When you install Infrastructure Automation and deploy Infrastructure Management, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can access and use Infrastructure Automation tools, such as Managed services or the Service catalog. For instance, users that do not have an Administrator role are not able to use the Infrastructure Management Managed services and Service Catalog or create user groups. For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

11. Disable usage data collection (optional)

To help the development of Infrastructure Automation - Infrastructure Management, aggregated usage data is collected to analyze how Infrastructure Management is used. The collection of usage data is enabled by default, but can be disabled.

For the data collection, Infrastructure Management uses the existing daily job that is used for audit logging of managed resources and for licensing tracking. This job is extended to collect and send the usage data metrics to IBM. The sent data is then stored in IBM controlled GDPR-compliant systems. The usage data that is collected does not include personal information, passwords, or specific details. Only the following data is collected:

- The number of virtual machines (VMs), hosts, providers (and provider types), services, and service catalog items that are being used in the Infrastructure Management inventory.
- Whether Infrastructure Management is deployed as a containerized deployment (podified) or virtual machine appliance.
- (Stand-alone deployments) The architecture where Infrastructure Management is deployed (Linux x86_64, Linux on Power (ppc64le), Linux® on IBM Z® and LinuxONE (s390x)).

Configuring the collection of usage data

To configure the collection of usage data, a secret is used, which includes your opt-in details, or your opt-out (disabling) of the data collection. Infrastructure Management uses the same **`aiops-metrics-processor`** secret as IBM Cloud Pak for AIOps to configure the opt-in or opt-out details for usage data collection. If you deployed Infrastructure Automation - Infrastructure Management and IBM Cloud Pak for AIOps in the same namespace, you can share this secret for configuring the data collection.

Follow the steps to create the **`aiops-metrics-processor`** secret to configure your opt-in details:

1. From OpenShift Container Platform console, click **Workloads** > **Secrets**.
2. From the Project menu, select the project that you created earlier in *Create a custom project (namespace)* step.
3. Click **Create** > **Key/value** secret from the menu. The create key/value secret form is displayed.
4. Enter the following Key/Value pairs:
    - *Secret name*: name of the secret, for example **`aiops-metrics-processor`**.
    - *customerName*: your company name.
    - *customerICN*: your IBM Customer Number (ICN).
    - *environment*: you can choose **`trial`** for testing, **`poc`** for proof of concept, or **`production`** for production environments.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must disable the collection of usage data.

Disabling the collection of usage data

To disable the collection of usage data, add the key/value pair **`enableCollection=false`** in the created secret.

1. From OpenShift Container Platform console, click **Workloads** > **Secrets**.
2. Click **secret name** that you created for collection of usage data, for example **`aiops-metrics-processor`**.
3. Add **`enableCollection`** as key and **`false`** as its value.
4. Click **Save**.

Note: You can update your usage data collection preferences after installation.

Online installation of Infrastructure Automation for use with IBM Cloud Pak for AIOps (CLI)

Follow these steps to complete an online installation of Infrastructure Automation, for use with IBM Cloud Pak® for AIOps.

If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation. Install IBM Cloud Pak for AIOps before you install Infrastructure Automation.

Before you begin

- Review the Planning section.
- Check that your system meets the minimum requirements for installing IBM Cloud Pak for AIOps and Infrastructure Automation. **Important:** The **`storageClass`** and **`storageClassLargeBlock`** that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the **`storageClass`** and **`storageClassLargeBlock`** that are used for creating the Infrastructure Automation custom resource.
- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with **`oc login`** for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the Infrastructure Automation operators need, see Permissions (Infrastructure Automation).
- A user with **`cluster-admin`** privileges is needed for the following operations:
    - Install IBM Cloud Pak for AIOps
    - Create the catalog source
    - Install Cert Manager
    - Install the License Service
    - Create an OperatorGroup

Installation steps

Follow these steps to install Infrastructure Automation.

1. Install IBM Cloud Pak for AIOps
2. Create the catalog source
3. Install Cert Manager
4. Install the License Service

5. Install the Infrastructure Automation operator
6. Create the custom resource for Infrastructure Automation
7. Verify the deployment
8. Log in to the IBM Cloud Pak for AIOps console
9. Deploying Infrastructure Management
10. Assign user roles and permissions
11. Enable usage data collection (optional)

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`** and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install IBM Cloud Pak for AIOps

It is recommended that you install IBM Cloud Pak for AIOps before you install Infrastructure Automation. IBM Cloud Pak for AIOps and Infrastructure Automation can be deployed on the same Red Hat OpenShift cluster.

Important: IBM Cloud Pak for AIOps and Infrastructure Automation must be installed in the same namespace. Installing IBM Cloud Pak for AIOps and Infrastructure Automation in separate namespaces is not supported.

To install IBM Cloud Pak for AIOps, follow the instructions in Installing IBM Cloud Pak for AIOps.

2. Create the catalog source

Add the Infrastructure Automation catalog source to your Red Hat OpenShift cluster.

After installation, the **`ibm-operator-catalog`** CatalogSource object determines whether the upgrade of your Infrastructure Automation deployment is initiated automatically when a new patch becomes available. The **`ibm-operator-catalog`** CatalogSource object can be configured to automatically poll for and retrieve a newer catalog by enabling the polling attribute **`spec.updateStrategy.registryPoll`**. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your Infrastructure Automation deployment is initiated. For more information, see Upgrading Infrastructure Automation.

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

Note: **`ibm-operator-catalog`** also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

Run the steps in Create the catalog source with automatic upgrade disabled or Create the catalog source with automatic upgrade enabled.

Create the catalog source with automatic upgrade disabled

1. Run the following command to create the **`ibm-operator-catalog`** CatalogSource object without polling enabled.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
EOF
```

2. Update the **`ibm-operator-catalog`** CatalogSource to fix it to always use the current image digest, instead of **`icr.io/cpopen/ibm-operator-catalog:latest`**. This ensures that the **`ibm-operator-catalog`** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart. Run the following commands:

```
IMGDIGEST=`oc get pods -n openshift-marketplace -l=olm.catalogSource=ibm-operator-catalog --no-headers -o=jsonpath="
{.items[0].status.containerStatuses[0].imageID}" -n openshift-marketplace`
CATALOGIMG=`oc get catalogsource ibm-operator-catalog -n openshift-marketplace -o jsonpath='{.spec.image}'`
if [ "${CATALOGIMG}" = "icr.io/cpopen/ibm-operator-catalog:latest" ]; then
  oc patch catalogsource ibm-operator-catalog -n openshift-marketplace --type=json -p "[{ "op": "replace", "path":
"/spec/image", "value": "\"$IMGDIGEST\"" }]"
fi
```

3. Verify that the *ibm-operator-catalog* CatalogSource object is present, and is returned by the following command.

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

Example output:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
NAME                    DISPLAY               TYPE   PUBLISHER   AGE
ibm-operator-catalog    IBM Operator Catalog    grpc   IBM         4h13m
```

Create the catalog source with automatic upgrade enabled

1. Run the following command to create the **ibm-operator-catalog** CatalogSource object with polling enabled.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

2. Verify that the *ibm-operator-catalog* CatalogSource object is present, and is returned by the following command.

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

Example output:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
NAME                    DISPLAY               TYPE   PUBLISHER   AGE
ibm-operator-catalog    IBM Operator Catalog    grpc   IBM         4h13m
```

3. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed a certificate manager as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters, then you must ensure that you have a certificate manager on each cluster.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
```

```
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-cert-manager get pods
   ```

   Example output for a successful IBM Cloud Pak foundational services Cert Manager installation:

   ```
   NAME                                       READY   STATUS    RESTARTS   AGE
   cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
   cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
   cert-manager-webhook-8598787c8-s4lkt       1/1     Running   0          8d
   ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
   ```

4. Install the License Service

Skip this step if the IBM Cloud Pak foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed IBM Cloud Pak foundational services License Service as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters then you must ensure that IBM Cloud Pak foundational services License Service is installed on each cluster.

Infrastructure Automation requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

1. Run the following command to create the resource definitions that you need:

   ```
   cat << EOF | oc apply -f -
   apiVersion: v1
   kind: Namespace
   metadata:
     name: ibm-licensing
   ---
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: ibm-licensing-operator-group
     namespace: ibm-licensing
   spec:
     targetNamespaces:
     - ibm-licensing
   ---
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-licensing-operator-app
     namespace: ibm-licensing
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-licensing-operator-app
     source: ibm-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

2. Run the following command to ensure that the IBM Cloud Pak foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-licensing get pods
   ```

   Example output for a successful IBM Cloud Pak foundational services License Service installation:

   ```
   NAME                                              READY   STATUS    RESTARTS   AGE
   ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
   ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
   ```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5. Install the Infrastructure Automation operator

For more information about operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the `All Namespaces` installation mode, and proceed to step 2, Install the Infrastructure Automation operator.

If you are installing using the **OwnNamespace** installation mode, then you must create an operator group in your custom project (namespace), or the Infrastructure Automation operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one. For more information, see Operator installation modes.

Create the operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cp4aiops-operator-group
  namespace: cp4aiops
spec:
  targetNamespaces:
    - cp4aiops
EOF
```

2. Set an environment variable for the Infrastructure Automation catalog.

```
export CATALOG_SRC_IA=<ia_catalog>
```

Where **<ia_catalog>** is **ibm-operator-catalog** for an online deployment, or **ibm-infrastructure-automation-operator-catalog** for an offline deployment.

3. Install the Infrastructure Automation operator.

Run the following command.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-infrastructure-automation-operator
  namespace: cp4aiops
spec:
  channel: v4.8
  installPlanApproval: Automatic
  name: ibm-infrastructure-automation-operator
  source: ${CATALOG_SRC_IA}
  sourceNamespace: openshift-marketplace
EOF
```

Warning: **installPlanApproval** must not be changed to **Manual**. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

After a few minutes, verify that the Infrastructure Automation operator is installed in the **cp4aiops** project (namespace) with the following command:

```
oc get pods -n cp4aiops | grep ibm-infrastructure-automation-operator-controller-manager
```

6. Create Infrastructure Automation custom resource with default values

You can create the Infrastructure Automation custom resource with the default set of values, or customize these values before you create the Infrastructure Automation custom resource.

Customize the default values for the Infrastructure Automation custom resource to modify the default replica count of the pods in Managed services, or to adjust any of the following defaults for Managed services:

- Default pre-created secrets in Managed services
- DefaultManaged services proxy setting
- Default Managed services default github.com token to import templates
- Default HTTPS certificate for Managed services Management Console
- Configuring Managed services to set the SSH keys and known hosts in provider terraform pod

Run the following command to create an instance of the Infrastructure Automation custom resource called **IAConfig** with its default values. With the default values, you can install Managed services.

The list of installation parameters with its default values are listed in Managed services installation parameters page.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
  - enabled: <set to true to install Infrastructure Management component of Infrastructure Automation, false otherwise>
    name: ibm-management-im-install
    spec: {}
  - enabled: <set to true to install Managed services component of Infrastructure Automation, false otherwise>
    name: ibm-management-cam-install
    spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <set to the same storageClass that IBM Cloud Pak for AIOps uses>
```

```
  storageClassLargeBlock: <set to the same storageClassLargeBlock that IBM Cloud Pak for AIOps uses>
EOF
```

To customize Managed services, you must have your customization parameters under **spec.manageservice** section.

```
- enabled: <Set to true to install Managed services component of Infrastructure Automation, false otherwise>
  name: ibm-management-cam-install
  spec:
    manageservice:
      <Set your custom installation parameter values>
```

The list of install parameters that can be customized along with its default values are listed in Managed services installation parameters page.

Note: This option does not install Infrastructure Management. If you need to install Infrastructure Management, complete the installation of Infrastructure Automation and follow the steps that are listed in Deploying Infrastructure Management.

7. Verify the deployment

After a few minutes, run the following command to verify that your deployment is successful.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running       4.8.1     rook-cephfs    rook-ceph-block          8m44s
```

8. Log in to the IBM Cloud Pak for AIOps console

After you successfully install Infrastructure Automation, get the URL for accessing the Cloud Pak for AIOps console.

1. Use the following command to get the URL to access the Cloud Pak for AIOps console:

   ```
   oc get route -n cp4aiops cpd -o jsonpath={.spec.host}
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Cloud Pak for AIOps console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Cloud Pak for AIOps console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

   ```
   oc -n cp4aiops get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d && echo
   ```

2. Get the password for the **admin** username.

   ```
   oc -n cp4aiops get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following extract shows a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

9. Deploying Infrastructure Management

For more information about how to deploy Infrastructure Management, see the following topics:

- Installing Infrastructure management as a containerized deployment (podified)
- Installing Infrastructure management as a virtual machine appliance

10. Assign user roles and permissions

When you install Infrastructure Automation and deploy Infrastructure Management, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can access and use Infrastructure Automation tools, such as Managed services or the Service catalog. For instance, users that do not have an Administrator role are not able to use the Infrastructure Management Managed services and Service Catalog or create user groups. For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

11. Disable usage data collection (optional)

To help the development of Infrastructure Automation - Infrastructure Management, aggregated usage data is collected to analyze how Infrastructure Management is used. The collection of usage data is enabled by default, but can be disabled.

For the data collection, Infrastructure Management uses the existing daily job that is used for audit logging of managed resources and for licensing tracking. This job is extended to collect and send the usage data metrics to IBM. The sent data is then stored in IBM controlled GDPR-compliant systems. The usage data that is collected does not include personal information, passwords, or specific details. Only the following data is collected:

- The number of virtual machines (VMs), hosts, providers (and provider types), services, and service catalog items that are being used in the Infrastructure Management inventory.
- Whether Infrastructure Management is deployed as a containerized deployment (podified) or virtual machine appliance.
- (Stand-alone deployments) The architecture where Infrastructure Management is deployed (Linux x86_64, Linux on Power (ppc64le), Linux® on IBM Z® and LinuxONE (s390x)).

If you want to disable the collection of usage data, run the following command to set **`enableCollection=false`** in the secret:

```
oc create secret generic aiops-metrics-processor -n <PROJECT_NAMESPACE> --from-literal=customerName=<CUSTOMER_NAME> --from-literal=customerICN=<CUSTOMER_ICN --from-literal=environment=<CUSTOMER_ENVIRONMENT> --from-literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation.

Online installation of stand-alone Infrastructure Automation (console)

Follow these steps to complete an online installation of stand-alone Infrastructure Automation.

If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation.

Before you begin

- Review the Planning section.
- Some steps must still be run with the command line. Ensure that you are logged in to your Red Hat OpenShift cluster with **`oc login`** for any steps that use the Red Hat OpenShift command-line interface (CLI).
- Check that your system meets the minimum requirements for installing Infrastructure Automation.
- If you require details about the permissions that the Infrastructure Automation operators need, see Permissions (Infrastructure Automation).
- A user with **`cluster-admin`** privileges is needed for the following operations:
  - Create a custom project
  - Install Cert Manager
  - Install the License Service
  - Create the catalog source

Procedure

Follow these steps to install Infrastructure Automation.

1. Install and configure Red Hat OpenShift
2. Configure storage
3. Create a custom project (namespace)
4. Create the entitlement key secret
5. Create the catalog source
6. Install Cert Manager
7. Install the License Service
8. Create an EgressFirewall
9. Install the Infrastructure Automation operator
10. Install Infrastructure Automation
11. Verify the deployment
12. Check the deployed pods in the deployed namespace
13. Log in to the Infrastructure Automation console
14. Assign user roles and permissions

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`** and IBM Cloud Pak® foundational services catalog source.

Site	Description

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install and configure Red Hat OpenShift Container Platform

For more information about the supported Red Hat OpenShift versions, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.
2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

You must configure your own storage for use with Infrastructure Automation. For more information, see Storage considerations.

3. Create a custom project (namespace)

Create a project (namespace) called `cp4aiops` for your Infrastructure Automation deployment:

1. From Red Hat OpenShift Container Platform console, navigate to **Home** > **Projects**.
2. Click **Create Project**.
3. Enter a *project (namespace)* name. For example, `cp4aiops`.
4. Add a *display name* and *description* as needed.
5. Click **Create**.

Note: Infrastructure Automation does not support deploying into different namespaces within a cluster.

4. Create the entitlement key pull secret

Complete the following steps to create a registry secret to enable your deployment to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Active entitlement keys* section, select **Copy** to copy the entitlement key to the clipboard.
3. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.
4. From the **Project** menu, select the project that you created earlier in Create a custom project (namespace).
5. Click the **Create** button, and select **Image pull secret** from the menu. The **Create image pull secret** form is displayed. Enter the following values and then click **Create**.
   - **Secret name**: `ibm-entitlement-key`
   - **Authentication type**: `Image registry credentials`
   - **Registry server address**: `cp.icr.io`
   - **Username**: `cp`
   - **Password**: use the entitlement key that you copied in step 2.

5. Create the catalog source

Add the Infrastructure Automation catalog source to your Red Hat OpenShift cluster.

After installation, the `ibm-operator-catalog` CatalogSource object determines whether the upgrade of your Infrastructure Automation deployment is initiated automatically when a new patch becomes available. The `ibm-operator-catalog` CatalogSource object can be configured to automatically poll for and retrieve a

newer catalog by enabling the polling attribute **`spec.updateStrategy.registryPoll`**. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your Infrastructure Automation deployment is initiated. For more information, see Upgrading Infrastructure Automation.

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

Note: **`ibm-operator-catalog`** also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

Run the steps in Create the catalog source with automatic upgrade disabled or Create the catalog source with automatic upgrade enabled.

Create the catalog source with automatic upgrade disabled

1. Create the **`ibm-operator-catalog`** CatalogSource object without polling enabled.

   Log in to your Red Hat OpenShift cluster's console. Click the plus icon in the upper right to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
   ```

2. Update the **`ibm-operator-catalog`** CatalogSource to fix it to always use the current image digest, instead of **`icr.io/cpopen/ibm-operator-catalog:latest`**. This ensures that the **`ibm-operator-catalog`** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart.

   1. Go to **Home** > **Projects**, and select **`openshift-marketplace`**.
   2. Go to **Workloads** > **Pods** (on the left menu), and then search for **`ibm-operator-catalog`**.
   3. Click the returned **`ibm-operator-catalog-<...>`** pod.
   4. Click **YAML** to switch to the YAML view.
   5. Search for **`imageID`** in the YAML, and copy down the value of **`spec.containerStatuses.imageID`**. The value is in a format similar to the following example:

      ```
      icr.io/cpopen/ibm-operator-catalog@sha256:<...>
      ```

   6. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, scroll down and click **`ibm-operator-catalog`**.
   7. Click **YAML** to switch to the YAML view.
   8. Set the value of **`spec.image`** to the value of the current image digest that you found in step 2, instead of to **`icr.io/cpopen/ibm-operator-catalog:latest`**.

3. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-operator-catalog* CatalogSource object is present.

Create the catalog source with automatic upgrade enabled

1. Create the **`ibm-operator-catalog`** CatalogSource object with polling enabled.

   Log in to your Red Hat OpenShift cluster's console. Click the plus icon in the upper right corner to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
     updateStrategy:
       registryPoll:
         interval: 45m
   ```

2. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-operator-catalog* CatalogSource object is present.

6. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Log in to your Red Hat OpenShift cluster's console.
2. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.
3. In the **All Items** field, enter **`IBM Cert Manager`**. The **IBM Cert Manager** operator is displayed.
4. Click the **IBM Cert Manager** tile. The **IBM Cert Manager** window is displayed.
5. Click **Install**. You see the **Install Operator** page.
6. Set the **Update Channel** to the **`v4.2`** version. If the Channel `v4.2`` version is not available, click other **IBM Cert Manager** tile from OperatorHub to install the correct version.
7. Set **Installation Mode** to **`All namespaces on the cluster (default)`**.
8. Set **Installed Namespace** to **`ibm-cert-manager(Operator recommended)`**.
9. Set **Update approval** to **`Automatic`**.
10. Click **Install**.

7. Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

Infrastructure Automation requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

Follow the instructions in Installing the License Service with OpenShift console in the IBM Cloud Pak foundational services documentation, from step 2 **`Create the ibm-licensing namespace`** onwards.

8. Create an EgressFirewall

There is no egress firewall policy defined when you install Infrastructure Automation, so outgoing traffic from workload pods to the internal and external network is unrestricted.

If you require a more secure environment, then use the following steps.

1. Create an **`EgressFirewall`** on your Red Hat OpenShift cluster to limit egress from the Infrastructure Automation project (namespace).

   For information on creating an **`EgressFirewall`**, see Configuring an egress firewall for a project.

   Note: There must be only one **`EgressFirewall`** per project (namespace).

2. Configure exceptions to the **`EgressFirewall`**.

   You must edit your **`EgressFirewall`** to allow traffic for external services, outbound cloud connections, and Infrastructure Management providers that you manage that have egress dependencies, otherwise these components fail when attempting egress.

   For more information about Infrastructure Automation connections and providers, see Managing connections and Managing Providers.

   Edit your **`EgressFirewall`** to allow or deny egress, as in the following example:

   ```
   kind: EgressFirewall
   metadata:
     name: default
   spec:
     egress:
     - type: Allow
       to:
         cidrSelector: <1.2.3.0/24>
     - type: Allow
       to:
         dnsName: <www.example.com>
     - type: Allow
       to:
         dnsName: <www.developer.kubernetes.com>
     - type: Deny
       to:
         cidrSelector: <0.0.0.0/0>
   ```

   Where the values you enter for **`dnsName`** and **`cidrSelector`** are the DNS names and addresses of sources that you require access for.

9. Install the Infrastructure Automation operator

For more information about operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

1. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.

2. In the **All Items** field, enter **`IBM Infrastructure Automation`**. The IBM Infrastructure Automation operator is displayed.
3. Click the **IBM Infrastructure Automation** tile. The IBM Infrastructure Automation window is displayed.
4. Click **Install**. The *Install Operator* page is displayed.
5. Enter the following values:
   - Set **Update channel** to *v4.8*.
   - **Installation mode** - For more information about installation modes, see Operator installation mode.
   - **Installed Namespace** - If you are using the **`OwnNamespace`** installation mode (a specific namespace), then set this field to be the project (namespace) in which to install the operator. If you are using the **`AllNamespaces`** installation mode, then set this field to **`openshift-operators`**.
   - Set **Update approval** to *Automatic*.

   Warning: Update approval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.
6. Click **Install** and wait for the IBM Infrastructure Automation operator to install.

   Warning: installPlanApproval must not be changed to Manual. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.
7. Verify that the IBM Infrastructure Automation operator is successfully installed.
8. Navigate to **Operators** > **Installed Operators**, and select your project from the **Projects** dropdown. IBM Infrastructure Automation operator and its dependant operators in the project are listed with a **Status** of *Succeeded*.

10. Install Infrastructure Automation

You can create the Infrastructure Automation custom resource with the default set of values, or customize these values before you create the Infrastructure Automation custom resource.

Customize the default values for the Infrastructure Automation custom resource to modify the default replica count of the pods in Managed services, or to adjust any of the following defaults for Managed services:

- Default pre-created secrets in Managed services
- DefaultManaged services proxy setting
- Default Managed services default github.com token to import templates
- Default HTTPS certificate for Managed services Management Console
- Configuring Managed services to set the SSH keys and known hosts in provider terraform pod

Create Infrastructure Automation custom resource with default values

The following YAML file creates an instance of the Infrastructure Automation custom resource called **`IAConfig`**. The list of installation parameters with its default values are listed in Managed services installation parameters page.

1. Navigate to **Operators** > **Installed Operators**.
2. Select the project (namespace) that you created earlier for your Infrastructure Automation deployment.
3. Under the *Provided APIs* section, click **IBM Infrastructure Automation**.
4. Click **Create IAConfig** tab.
5. Switch to the YAML view and paste the following file:

```
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
    - enabled: <set to true to install Infrastructure Management component of Infrastructure Automation, false
otherwise>
      name: ibm-management-im-install
      spec: {}
    - enabled: <set to true to install Managed services component of Infrastructure Automation, false otherwise>
      name: ibm-management-cam-install
      spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWO>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
```

   To customize Managed services, you must have your customization parameters under **`spec.manageservice`** section.

```
- enabled: <Set to true to install Managed services component of Infrastructure Automation, false otherwise>
  name: ibm-management-cam-install
  spec:
    manageservice:
      <Set your custom installation parameter values>
```

6. Click **Apply**.

Note: This option does not install Infrastructure Management. If you need to install Infrastructure Management, complete the installation of Infrastructure Automation and follow the steps that are listed in Create Infrastructure Automation custom resource and deploying Infrastructure Management operand.

Create Infrastructure Automation custom resource and deploying Infrastructure Management operand.

The following YAML file creates an instance of the Infrastructure Automation custom resource called IAConfig. The list of install parameters will deploy Infrastructure Management operand.

1. Navigate to **Operators** > **Installed Operators**.
2. Select the project (namespace) that you created earlier for your Infrastructure Automation deployment.
3. Under the *Provided APIs* section, click **IBM Infrastructure Automation**.
4. Click **Create IAConfig** tab.
5. Switch to the YAML view and paste the following file:

```
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
    - enabled: <set to true to install Infrastructure Management component of Infrastructure Automation, false
otherwise>
      name: ibm-management-im-install
      spec:
        iminstall:
          applicationDomain: <YOUR_IM_HTTPD_ROUTE>
          imagePullPolicy: Always
          imagePullSecret: ibm-entitlement-key
          initialAdminGroupName: <YOUR_LDAP_USER_GROUP>
          license:
            accept: true
    - enabled: <set to true to install Managed services component of Infrastructure Automation, false otherwise>
      name: ibm-management-cam-install
      spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWO>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
```

Where:

- You must specify an LDAP user-group and it must contain at least one user that is able to login to the Infrastructure Automation console. For example, you have an existing LDAP group that is named group100 and a user with the username user100 is a member of the group. You enter group100 for the value of <YOUR_LDAP_USER_GROUP>.
- YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your Infrastructure Automation console route. Obtain the console route and modify it to derive your Infrastructure Automation route.
    1. Click **Networking** > **Routes**.
    2. Click the route that is named as **`cp-console`**.
    3. Click **YAML**.
    4. Find the value for **`spec.host`**.

       Example output:

       `cp-console.apps.mycluster.myibm.com`

       Replace the first section cp-console with inframgmtinstall to create YOUR_IM_HTTPD_ROUTE

       For example: **`inframgmtinstall.apps.mycluster.myibm.com`**
- **`YOUR_LDAP_USER_GROUP`** is an existing user-group that is defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure Automation to match your existing LDAP group by name, and assigned an account role that facilitates SSO login.

Note: This option installs Infrastructure Management.

Important: After deploying the Infrastructure Management operand, make sure that the LDAP group is added to the Platform UI by navigating to **Administration > Access control**, and then create a new LDAP group from existing LDAP group. Add the LDAP group to the Platform UI. For more information, see Configuring LDAP connections.

11. Verify the deployment

After a few minutes, use the following steps to check the status of your Infrastructure Automation installation.

1. Click **Operators** > **Installed Operators**.
2. From the **Project** list, select the project (namespace) where Infrastructure Automation is deployed.

3. Click **IBM Infrastructure Automation**, then click **IBM Infrastructure Automation** tab.
4. Under **IAConfigs**, look at the entry with the name that you specified in the IAConfigs custom resource (Infrastructure Automation instance), and verify that the **Status** indicates *Phase: Running*.

Note: The STORAGECLASSLARGEBLOCK is blank if the environment is upgraded from a previous release.

There is no egress firewall policy defined when you install Infrastructure Automation, so outgoing traffic from workload pods to the internal and external network is unrestricted. There is a need to create an Egress network policy to make your environment more secure. Follow the steps here to create an **`EgressFirewall`**.

12. Check the deployed pods in the deployed namespace

From the OpenShift Container Platform console, navigate to **Workloads** > **Pods**, and verify that the following pods are running:

Name	Status	Status
1-automation	Running	1/1
1-ems-metrics-processor	Running	2/2
1-event-handler	Running	1/1
1-generic	Running	2/2
1-priority	Running	2/2
1-remote-console	Running	1/1
1-reporting	Running	2/2
1-schedule	Running	1/1
1-ui	Running	2/2
1-web-service	Running	2/2
cam-iaas	Running	1/1
cam-install-operator-controller-manager	Running	1/1
cam-mongo	Running	1/1
cam-orchestration	Running	1/1
cam-portal-ui	Running	1/1
cam-provider-ansible	Running	0/0
cam-provider-bpm	Running	0/0
cam-provider-terraform-api	Running	1/1
cam-provider-terraform-runtime	Running	0/0
cam-proxy	Running	1/1
cam-service-composer-api	Running	1/1
cam-service-composer-ui	Running	1/1
cam-service-library-ui	Running	1/1
cam-service-library-ui-api	Running	1/1
cam-tenant-api	Running	1/1
cam-ui-basic	Running	1/1
ibm-infra-management-application	Running	1/1
ibm-infra-management-install-operator	Running	1/1
ibm-infrastructure-automation-operator-controller-manager	Running	1/1

13. Access the Infrastructure Automation console

After you successfully install Infrastructure Automation, get the URL for accessing the Infrastructure Automation console.

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator.
2. Click **Networking** > **Routes**.
3. Search for **cpd** route and click on this route.
4. Click the URL that is displayed under **Location**. For example, sample URL is **`https://cpd-cp4aiops.apps.mycluster.mydomain`**.
5. In the Cloud Pak for AIOps console login page, select one of the following login options:
   - **OpenShift authentication**: The **`kubeadmin`** user is automatically used to log in to the Cloud Pak for AIOps console. The **`kubeadmin`** user has the same privileges as the Cloud Pak for AIOps console admin user.
   - **IBM provided credentials (admin only)**: The default username to access the console is **`cpadmin`**. To obtain the username and password, see Obtain IBM provided credentials (admin only).
   - **Enterprise LDAP**: LDAP users can log in to the Cloud Pak for AIOps console after IBM Cloud Pak for AIOps is configured with a single or multiple LDAP servers for the authentication and authorization. For more information, see Identity Management (IM).
6. Once you are logged in, go to the navigation bar, and expand **Automate Infrastructure**.
7. To access Infrastructure Management, click **Infrastructure management**. Login using the three mentioned methods.

Find the Infrastructure Automation console username and password

The default username to access the Infrastructure Automation console is **`admin`**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. To find the default username, select the project (namespace) that IBM Cloud Pak for AIOps is deployed, then navigate to **Workloads** > **Secrets**. Search the **`platform-auth-idp-credentials`** secret name in the search bar. Click **`platform-auth-idp-credentials`** to view the secret. You can see the value of username from the **`admin_username`** field.
2. To get the password, you can get the value from the **`admin_password`** field of the secret.

The following extract shows a sample output:

```
EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
```

Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

14. Assign user roles and permissions

When you install Infrastructure Automation and deploy Infrastructure Management, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can access and use Infrastructure Automation tools, such as Managed services or the Service catalog. For instance, users that do not have an Administrator role are not able to use the Infrastructure Management Managed services and Service Catalog or create user groups. For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

15. Configure usage data collection (optional)

To help the development of Infrastructure Automation - Infrastructure Management, aggregated usage data is collected to analyze how Infrastructure Management is used. The collection of usage data is enabled by default, but can be disabled.

For the data collection, Infrastructure Management uses the existing daily job that is used for audit logging of managed resources and for licensing tracking. This job is extended to collect and send the usage data metrics to IBM. The sent data is then stored in IBM controlled GDPR-compliant systems. The usage data that is collected does not include personal information, passwords, or specific details. Only the following data is collected:

- The number of virtual machines (VMs), hosts, providers (and provider types), services, and service catalog items that are being used in the Infrastructure Management inventory.
- Whether Infrastructure Management is deployed as a containerized deployment (podified) or virtual machine appliance.
- (Stand-alone deployments) The architecture where Infrastructure Management is deployed (Linux x86_64, Linux on Power (ppc64le), Linux® on IBM Z® and LinuxONE (s390x)).

Enabling the collection of usage data

To configure the collection of usage data, a secret is used, which includes your opt-in details, or your opt-out (disabling) of the data collection. Infrastructure Management uses the same **aiops-metrics-processor** secret as IBM Cloud Pak for AIOps to configure the opt-in or opt-out details for usage data collection. If you deployed Infrastructure Automation - Infrastructure Management and IBM Cloud Pak for AIOps in the same namespace, you can share this secret for configuring the data collection.

Follow the steps to create the **aiops-metrics-processor** secret to configure your opt-in details:

1. From OpenShift Container Platform console, click **Workloads** > **Secrets**.
2. From the Project menu, select the project that you created earlier in *Create a custom project (namespace)* step.
3. Click **Create** > **Key/value** secret from the menu. The create key/value secret form is displayed.
4. Enter the following Key/Value pairs:
    - *Secret name*: name of the secret, for example **aiops-metrics-processor**.
    - *customerName*: your company name.
    - *customerICN*: your IBM Customer Number (ICN).
    - *environment*: you can choose **trial** for testing, **poc** for proof of concept, or **production** for production environments.
5. If you have a firewall enabled, ensure that outbound traffic to **https://api.segment.io** is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must disable the collection of usage data. For more information about Disabling the collection of usage data, see Disabling the collection of usage data.

Disabling the collection of usage data

To disable the collection of usage data, add the key/value pair **enableCollection=false** in the created secret.

1. From OpenShift Container Platform console, click **Workloads** > **Secrets**.
2. Click **secret name** that you created for collection of usage data, for example **aiops-metrics-processor**.
3. Add **enableCollection** as key and **false** as its value.
4. Click **Save**.

Note: You can update your usage data collection preferences after installation.

Online installation of Infrastructure Automation (CLI)

Follow these steps to complete an online installation of Infrastructure Automation.

If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation.

Before you begin

- Review the Planning section.
- Check that your system meets the minimum requirements for installing Infrastructure Automation.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the Infrastructure Automation operators need, see Permissions (Infrastructure Automation).
- A user with `cluster-admin` privileges is needed for the following operations:
  - Create a custom project
  - Install Cert Manager
  - Install the License Service
  - Create the catalog source
  - Create an OperatorGroup

Procedure

Follow these steps to install Infrastructure Automation.

1. Install and configure Red Hat OpenShift
2. Configure storage
3. Create a custom project (namespace)
4. Create the entitlement key secret
5. Create the catalog source
6. Install Cert Manager
7. Install the License Service
8. Create an EgressFirewall
9. Install the Infrastructure Automation operator
10. Install Infrastructure Automation
11. Verify the deployment
12. Check the deployed pods in the deployed namespace
13. Log in to the Infrastructure Automation console
14. Assign user roles and permissions

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry` and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install and configure Red Hat OpenShift Container Platform

For more information about the supported Red Hat OpenShift versions, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.
2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

You must configure your own storage for use with Infrastructure Automation. For more information, see Storage considerations.

3. Create a custom project (namespace)

Create a project (namespace) called **cp4aiops** for your Infrastructure Automation deployment, by running the following command:

```
oc create namespace cp4aiops
```

Note: Infrastructure Automation does not support deploying into different namespaces within a cluster.

4. Create the entitlement key pull secret

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement keys* section, select **Copy key** to copy your entitlement key to the clipboard.
3. From the Red Hat OpenShift CLI, run the following command:

```
oc create secret docker-registry ibm-entitlement-key \
    --docker-username=cp\
    --docker-password=<entitlement-key> \
    --docker-server=cp.icr.io \
    --namespace=cp4aiops
```

Where **<entitlement-key>** is the entitlement key that you copied in the previous step.

5. Create the catalog source

Add the Infrastructure Automation catalog source to your Red Hat OpenShift cluster.

After installation, the **ibm-operator-catalog** CatalogSource object determines whether the upgrade of your Infrastructure Automation deployment is initiated automatically when a new patch becomes available. The **ibm-operator-catalog** CatalogSource object can be configured to automatically poll for and retrieve a newer catalog by enabling the polling attribute **spec.updateStrategy.registryPoll**. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your Infrastructure Automation deployment is initiated. For more information, see Upgrading Infrastructure Automation.

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

Run the steps in Create the catalog source with automatic upgrade disabled or Create the catalog source with automatic upgrade enabled.

Create the catalog source with automatic upgrade disabled

1. Run the following command to create the **ibm-operator-catalog** CatalogSource object without polling enabled.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
EOF
```

2. Update the **ibm-operator-catalog** CatalogSource to fix it to always use the current image digest, instead of **icr.io/cpopen/ibm-operator-catalog:latest**. This ensures that the **ibm-operator-catalog** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart. Run the following commands:

```
IMGDIGEST=`oc get pods -n openshift-marketplace -l=olm.catalogSource=ibm-operator-catalog --no-headers -o=jsonpath="
{.items[0].status.containerStatuses[0].imageID}" -n openshift-marketplace`
CATALOGIMG=`oc get catalogsource ibm-operator-catalog -n openshift-marketplace -o jsonpath='{.spec.image}'`
if [ "${CATALOGIMG}" = "icr.io/cpopen/ibm-operator-catalog:latest" ]; then
  oc patch catalogsource ibm-operator-catalog -n openshift-marketplace --type=json -p "[{ "op": "replace", "path":
"/spec/image", "value": "\"$IMGDIGEST\"" }]"
fi
```

3. Verify that the *ibm-operator-catalog* CatalogSource object is present, and is returned by the following command.

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

Example output:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
NAME                    DISPLAY                 TYPE   PUBLISHER   AGE
ibm-operator-catalog    IBM Operator Catalog    grpc   IBM         4h13m
```

Create the catalog source with automatic upgrade enabled

1. Run the following command to create the **ibm-operator-catalog** CatalogSource object with polling enabled.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

2. Verify that the *ibm-operator-catalog* CatalogSource object is present, and is returned by the following command.

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

Example output:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
NAME                   DISPLAY                TYPE   PUBLISHER   AGE
ibm-operator-catalog   IBM Operator Catalog   grpc   IBM         4h13m
```

6. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-cert-manager get pods
```

Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

```
NAME                                        READY   STATUS    RESTARTS   AGE
cert-manager-cainjector-674854c49d-vstq4    1/1     Running   0          8d
cert-manager-controller-646d4bd6fd-zwmqm    1/1     Running   0          8d
cert-manager-webhook-8598787c8-s4lkt        1/1     Running   0          8d
ibm-cert-manager-operator-c96957695-dkxnm   1/1     Running   0          8d
```

7. Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

Infrastructure Automation requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-licensing
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-licensing-operator-group
  namespace: ibm-licensing
spec:
  targetNamespaces:
  - ibm-licensing
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-licensing-operator-app
  namespace: ibm-licensing
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-licensing-operator-app
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-licensing get pods
```

Example output for a successful IBM Cloud Pak® foundational services License Service installation:

```
NAME                                                 READY   STATUS    RESTARTS   AGE
ibm-licensing-operator-db4cd746c-xzmlf               1/1     Running   0          8d
ibm-licensing-service-instance-596b99588f-76cc5      1/1     Running   0          8d
```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

8. Create an EgressFirewall

There is no egress firewall policy defined when you install Infrastructure Automation, so outgoing traffic from workload pods to the internal and external network is unrestricted.

If you require a more secure environment, then use the following steps.

1. Create an **EgressFirewall** on your Red Hat OpenShift cluster to limit egress from the Infrastructure Automation project (namespace).

   For information on creating an **EgressFirewall**, see Configuring an egress firewall for a project.

   Note: There must be only one **EgressFirewall** per project (namespace).

2. Configure exceptions to the **EgressFirewall**.

   You must edit your **EgressFirewall** to allow traffic for external services, outbound cloud connections, and Infrastructure Management providers that you manage that have egress dependencies, otherwise these components fail when attempting egress.

   For more information about Infrastructure Automation connections and providers, see Managing connections and Managing Providers.

   Edit your **EgressFirewall** to allow or deny egress, as in the following example:

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.example.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

   Where the values you enter for **dnsName** and **cidrSelector** are the DNS names and addresses of sources that you require access for.

9. Install the Infrastructure Automation operator

For more information about operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the **`All Namespaces`** installation mode, and proceed to step 2, *Set an environment variable for the Infrastructure Automation catalog*.

   If you are installing using the **`OwnNamespace`** installation mode, then you must create an operator group in your custom project (namespace), or the Infrastructure Automation operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one. For more information, see Operator installation modes.

   Create the operator group by running the following command:

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: cp4aiops-operator-group
     namespace: cp4aiops
   spec:
     targetNamespaces:
       - cp4aiops
   EOF
   ```

2. Set an environment variable for the Infrastructure Automation catalog.

   ```
   export CATALOG_SRC_IA=<ia_catalog>
   ```

   Where **`<ia_catalog>`** is *ibm-operator-catalog* for an online deployment, or *ibm-infrastructure-automation-operator-catalog* for an offline deployment.

3. Install the Infrastructure Automation operator.

   Run the following command.

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-infrastructure-automation-operator
     namespace: cp4aiops
   spec:
     channel: v4.8
     installPlanApproval: Automatic
     name: ibm-infrastructure-automation-operator
     source:  ${CATALOG_SRC_IA}
     sourceNamespace: openshift-marketplace
   EOF
   ```

Warning: installPlanApproval must not be changed to Manual. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

After a few minutes, verify that the Infrastructure Automation operator is installed in the **`cp4aiops`** project (namespace) with the following command:

```
oc get pods -n cp4aiops | grep ibm-infrastructure-automation-operator-controller-manager
```

10. Install Infrastructure Automation

You can create the Infrastructure Automation custom resource with the default set of values, or customize these values before you create the Infrastructure Automation custom resource.

Customize the default values for the Infrastructure Automation custom resource to modify the default replica count of the pods in Managed services, or to adjust any of the following defaults for Managed services:

- Default pre-created secrets in Managed services
- DefaultManaged services proxy setting
- Default Managed services default github.com token to import templates
- Default HTTPS certificate for Managed services Management Console
- Configuring Managed services to set the SSH keys and known hosts in provider terraform pod

Create Infrastructure Automation custom resource with default values

With the default values, you can install Managed services.

Run the following command to create an instance of the Infrastructure Automation custom resource called **`IAConfig`**. The list of installation parameters with its default values are listed in Managed services installation parameters page.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
```

```
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
    - enabled: <set to true to install Infrastructure Management component of Infrastructure Automation, false otherwise>
      name: ibm-management-im-install
      spec: {}
    - enabled: <set to true to install Managed services component of Infrastructure Automation, false otherwise>
      name: ibm-management-cam-install
      spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWO>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

To customize Managed services, you must have your customization parameters under **spec.manageservice** section.

```
name: ibm-management-cam-install
spec:
  manageservice:
    <Set your custom installation parameter values>
```

The list of install parameters that can be customized are listed in Managed services installation parameters page.

Note: This option does not install Infrastructure Management. If you need to install Infrastructure Management, complete the installation of Infrastructure Automation and follow the steps that are listed in Create Infrastructure Management custom resource.

An example of this is shown here

```
name: ibm-management-cam-install
spec:
  manageservice:
    global:
      offline: true
    license:
    accept: true
```

where you use customization when installing Managed services in offline mode.

Create Infrastructure Management custom resource

Important: If you previously created **IAConfig** with **spec.manageservice** section (customized parameters for Managed services), then you can edit the existing IAConfig file that you created, copy the content of **spec.iminstall** section as listed in the following example YAML file, and paste this content in the existing IAConfig file. In this case, you do not need to create the following YAML file.

Run the following command to create an instance of the Infrastructure Automation custom resource called IAConfig. The list of install parameters will deploy Infrastructure Management operand.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
    - enabled: <set to true to install Infrastructure Management component of Infrastructure Automation, false otherwise>
      name: ibm-management-im-install
      spec:
        iminstall:
          applicationDomain: <YOUR_IM_HTTPD_ROUTE>
          imagePullPolicy: Always
          imagePullSecret: ibm-entitlement-key
          initialAdminGroupName: <YOUR_LDAP_USER_GROUP>
          license:
            accept: true
    - enabled: <set to true to install Managed services component of Infrastructure Automation, false otherwise>
      name: ibm-management-cam-install
      spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWO>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

Where:

- YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your Infrastructure Automation console route. Use the following oc command to obtain the console route and modify it to derive your Infrastructure Automation route.

  ```
  oc get ingress.config.openshift.io -o=jsonpath='{.items[0].spec.domain}'
  ```

  Example output:

  ```
  apps.mycluster.myibm.com
  ```

- Add inframgmtinstall to the output to create **YOUR_IM_HTTPD_ROUTE**

  For example: **inframgmtinstall.apps.mycluster.myibm.com**

- YOUR_LDAP_USER_GROUP is an existing user-group that is defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure Automation to match your existing LDAP group by name, and assigned an account role that facilitates SSO login.

Important:

- You must specify an LDAP user-group and it must contain at least one user that is able to login to the Infrastructure Automation console. For example, you have an existing LDAP group that is named group100 and a user with the username user100 is a member of the group. You enter group100 for the value of <YOUR_LDAP_USER_GROUP>.
- YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your Infrastructure Automation console route. Use the following oc command to obtain the console route and modify it to derive your Infrastructure Automation route.

  ```
  oc -n cp4aiops get route cp-console --template `{{.spec.host}}`
  ```

  Example output:

  ```
  cp-console.apps.mycluster.myibm.com
  ```

  Replace the first section cp-console with inframgmtinstall to create YOUR_IM_HTTPD_ROUTE

  For example: **`inframgmtinstall.apps.mycluster.myibm.com`**

- **`YOUR_LDAP_USER_GROUP`** is an existing user-group that is defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure Automation to match your existing LDAP group by name, and assigned an account role that facilitates SSO login.

Note: This option installs Infrastructure Management.

Important: After deploying the Infrastructure Management operand, make sure that the LDAP group is added to the Platform UI by navigating to **Administration > Access control**, and then create a new LDAP group from existing LDAP group. Add the LDAP group to the Platform UI. For more information, see Configuring LDAP connections.

11. Verify the deployment

After a few minutes, run the following command to verify that your deployment is successful.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS    STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Installing    4.8.1     rook-cephfs     rook-ceph-block          8m44s
```

Note: The STORAGECLASSLARGEBLOCK is blank if the environment is upgraded from a previous release.

There is no egress firewall policy defined when you install Infrastructure Automation, so outgoing traffic from workload pods to the internal and external network is unrestricted. There is a need to create an Egress network policy to make your environment more secure. Follow the steps here to create an **`EgressFirewall`**.

12. Check the deployed pods in the deployed namespace

```
oc -n cp4aiops get deploy | grep -E "cam|ibm-infra|1-"
```

Example output:

```
1-automation                                        1/1     1            1           15m
1-ems-metrics-processor                             2/2     2            2           15m
1-event-handler                                     1/1     1            1           15m
1-generic                                           2/2     2            2           15m
1-priority                                          2/2     2            2           15m
1-remote-console                                    1/1     1            1           15m
1-reporting                                         2/2     2            2           15m
1-schedule                                          1/1     1            1           15m
1-ui                                                2/2     2            2           15m
1-web-service                                       2/2     2            2           15m
cam-iaas                                            1/1     1            1           17m
cam-install-operator-controller-manager             1/1     1            1           20m
cam-mongo                                           1/1     1            1           17m
cam-orchestration                                   1/1     1            1           17m
cam-portal-ui                                       1/1     1            1           17m
cam-provider-ansible                                0/0     0            0           17m
cam-provider-bpm                                    0/0     0            0           17m
cam-provider-terraform-api                          1/1     1            1           17m
cam-provider-terraform-runtime                      0/0     0            0           17m
cam-proxy                                           1/1     1            1           17m
cam-service-composer-api                            1/1     1            1           17m
cam-service-composer-ui                             1/1     1            1           17m
cam-service-library-ui                              1/1     1            1           17m
cam-service-library-ui-api                          1/1     1            1           17m
cam-tenant-api                                      1/1     1            1           17m
cam-ui-basic                                        1/1     1            1           17m
ibm-infra-management-application                    1/1     1            1           20m
ibm-infra-management-install-operator               1/1     1            1           20m
ibm-infrastructure-automation-operator-controller-manager   1/1     1            1           41m
```

13. Access the Infrastructure Automation console

After you successfully install Infrastructure Automation, get the URL for accessing the Infrastructure Automation console.

1. Use the following command to get the URL to access the Infrastructure Automation console:

   ```
   oc get route -n cp4aiops cpd -o jsonpath={.spec.host}
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **`https://cpd-cp4aiops.apps.mycluster.mydomain`**

2. Enter the URL in your browser to open the Infrastructure Automation console. Log in with your username and password.

Find the Infrastructure Automation console username and password

The default username to access the Infrastructure Automation console is **`admin`**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

   ```
   oc -n cp4aiops get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d && echo
   ```

2. Get the password for the **`admin`** username.

   ```
   oc -n cp4aiops get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following extract shows a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be **`EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA`**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

14. Assign user roles and permissions

When you install Infrastructure Automation and deploy Infrastructure Management, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can access and use Infrastructure Automation tools, such as Managed services or the Service catalog. For instance, users that do not have an Administrator role are not able to use the Infrastructure Management Managed services and Service Catalog or create user groups. For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

Installing on cloud platforms

When you are installing IBM Infrastructure Automation on specific cloud platforms, there are some suggested and best practices that you can follow to help you complete the installation process.

While the following suggested and required configurations are only for deployment on Red Hat® OpenShift® on IBM Cloud® service (ROKS), Red Hat OpenShift Service on AWS (ROSA), and Azure Red Hat OpenShift (ARO), IBM Infrastructure Automation can be installed on any platform that has a supported version of Red Hat® OpenShift® Container Platform installed that meets all other prerequisities.

View the following topics for guidance on installing IBM Cloud Pak for AIOps on cloud platforms:

- Installing IBM Infrastructure Automation on ROKS
- Installing IBM Infrastructure Automation on ROSA
- Installing IBM Infrastructure Automation on ARO

View the following topics for details about installing IBM Infrastructure Automation on Red Hat OpenShift.

- Installing for use with Cloud Pak for AIOps online (console)
- Installing for use with Cloud Pak for AIOps online (CLI)
- Installing stand-alone Infrastructure Automation online (console)
- Installing stand-alone Infrastructure Automation online (CLI)

Installing IBM Infrastructure Automation on Red Hat OpenShift on IBM Cloud (ROKS)

If you are installing on Red Hat OpenShift on IBM Cloud (ROKS), you can follow any of the main online installation options for installing IBM Infrastructure Automation. For installing on Red Hat OpenShift on IBM Cloud, you need to replace any storage class configuration with your configured storage class for Red Hat

OpenShift on IBM Cloud.

- Installing for use with Cloud Pak for AIOps online (console)
- Installing for use with Cloud Pak for AIOps online (CLI)
- Installing stand-alone Infrastructure Automation online (console)
- Installing stand-alone Infrastructure Automation online (CLI)

Storage

Installing on a classic infrastructure:

When you are configuring storage as part of the installation, consider using the default storage that is available with Red Hat OpenShift on IBM Cloud (ROKS). This storage has been tested with a Infrastructure Automation deployment on Red Hat OpenShift on IBM Cloud. You can choose to deploy and use your own storage if needed or preferred.

The following storage classes are the default storage classes for Red Hat OpenShift on IBM Cloud:

- `storageClass: ibmc-file-gold-gid`
- `storageClassLargeBlock: ibmc-block-gold`

Note: `ibmc-file-gold-gid` does not have volume expansion enabled by default, which is required by Infrastructure Automation. For more information, see Storage class requirements.

Installing on VPC infrastructure (using Red Hat OpenShift Data Foundation storage):

Consider using Red Hat OpenShift Data Foundation storage. This storage has been tested with a Infrastructure Automation deployment on Red Hat OpenShift on IBM Cloud. To use this storage provider, you need to first deploy Red Hat OpenShift Data Foundation on your CPV cluster. For more information, see Deploying OpenShift Data Foundation on VPC clusters.

When you do use this storage, the following storage classes are the default storage classes to use for Red Hat OpenShift on IBM Cloud:

- `storageClass: ocs-storagecluster-cephfs`
- `storageClassLargeBlock: ocs-storagecluster-ceph-rbd`

For more information about these storage classes, see the IBM Cloud Docs Storage class reference.

Installing IBM Infrastructure Automation on Red Hat OpenShift Service on AWS (ROSA)

If you are installing IBM Infrastructure Automation on Red Hat OpenShift Service on AWS (ROSA), then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Infrastructure Automation on Red Hat® OpenShift® Container Platform.

Before you begin

Ensure that you meet the following prerequisites:

- You have an AWS account.
- You have a Red Hat subscription.
- You have a Portworx account and license. The following steps are tested with, and require, Portworx storage. If needed, register for a Portworx account.
- Ensure that your environment meets the prerequisites for Red Hat OpenShift Service on AWS.

Installation procedure

Follow these steps to install Infrastructure Automation on Red Hat OpenShift Service on AWS

1. Configure AWS and Red Hat access
2. Create a Red Hat OpenShift Service on AWS Red Hat OpenShift Container Platform cluster
3. Configure storage
4. Install Infrastructure Automation

1. Configure AWS and Red Hat access

1. Log in to the AWS console to verify that you have access to the console. If needed, the *Login* page provides a link to register for an account.
2. From the AWS console, deploy Red Hat OpenShift Container Platform into a VPC within your AWS environment if it is not already deployed. For more information see Red Hat OpenShift Service on AWS quickstart guide.
3. Log in to the Red Hat console to verify that you have access to the console. If needed, the *Login* page provides a link to register for an account.
4. Retrieve your Red Hat Red Hat OpenShift Cluster Manager API Token by entering the following URL:
   - https://console.redhat.com/openshift/token/rosa/show
5. Configure the ROSA CLI to work with your AWS account. For more information, see Install and configure the latest ROSA CLI.

6. (Optional) Create a test cluster to validate your environment configuration

    1. Log in to Red Hat OpenShift Service on AWS.

       ```
       rosa login --token="<token>"
       ```

    2. Configure your AWS account to allow a IAM (non-STS) ROSA cluster.

       ```
       rosa init
       ```

    3. Create the test cluster.

       ```
       rosa create cluster --cluster-name=mytest
       ```

       During the cluster creation, you can review the installation logs to watch the progress.

       ```
       rosa logs install -c mytest --watch
       ```

    4. After the installation completes, list the cluster and note the cluster ID.

       ```
       rosa list clusters
       ```

    5. Describe your cluster using the cluster ID.

       ```
       rosa describe cluster -c 1ab23de4fghijk5lmno6p78q9r1stu2v
       ```

    6. Add an identity provider.

       ```
       rosa create idp --cluster 1ab23de4fghijk5lmno6p78q9r1stu2v --interactive
       ```

       You need to use the noninteractive mode and get some constant values to use

    7. Create your initial admin account.

       ```
       rosa create admin --cluster=1ab23de4fghijk5lmno6p78q9r1stu2v
       ```

    8. Log in to your test cluster by using the admin and password.

       ```
       oc login https://api.jgtest01.dj5a.p1.openshiftapps.com:6443 --username cluster-admin --password XXXXXX
       ```

       If your login is successful, your test cluster is working and your environment is configured.

    9. Clean up and delete the test cluster to proceed with creating your main cluster.

       ```
       rosa delete cluster -c 1ab23de4fghijk5lmno6p78q9r1stu2v
       ```

2. Create a Red Hat OpenShift Service on AWS Red Hat OpenShift Container Platform cluster

1. Log in to the Red Hat OpenShift Service on AWS CLI.

   ```
   rosa login
   ```

   Log in to the site https://cloud.redhat.com/openshift/token/rosa to retrieve your token. Then, copy and paste the token into the CLI prompt.

2. Create your cluster.

   ```
   rosa create cluster --cluster-name=<myclustername> --compute-machine-type=m5.8xlarge  --compute-nodes=<Number of Compute Nodes>  --version <Red Hat OpenShift_version>
   ```

   Example:

   ```
   rosa create cluster --cluster-name=cluster-test1  --compute-machine-type=m5.8xlarge  --compute-nodes=6 --version 4.16
   ```

3. After your cluster is ready, create your cluster administrator account.

    1. Run the following command:

       ```
       rosa create admin -c <myclustername>
       ```

       Important: Record the admin username (cluster-admin) and password for future use.

    2. Run the `oc login` command with the cluster administrator credentials.

4. Verify that all nodes are in `Ready` state before proceeding. Your cluster may take 40+ minutes to create and for you to be able to successfully log in with the 'oc login' command.

    1. Run the following command:

       ```
       oc get nodes
       ```

    2. Run the following command:

       ```
       rosa describe cluster -c <myclustername>
       ```

       Record the console URL for the Red Hat OpenShift Console and the Details Page for viewing the cluster details.

3. Verify that you can access the Red Hat OpenShift console by logging in to the provided Red Hat OpenShift console URL using the **`cluster-admin`** role and credentials.

3. Configure storage

AWS native storage and Portworx are tested and supported storage options for installing Infrastructure Automation on ROSA. Use the following instructions to configure AWS native storage or Portworx as your storage solution.

AWS native storage

The AWS native storage classes that are required by Infrastructure Automation are **`efs-sc`** and **`gp3-csi`**.

Amazon Elastic Block Store (EBS) provides block storage. The storage class is **`gp3-csi`**, and is created when ROSA is installed.

Amazon Elastic File System (EFS) provides file storage. The storage class is **`efs-sc`**, and you must create it. For more information, see **`Setting up the AWS EFS CSI Driver Operator`** and **`Creating the AWS EFS storage class`** in the Red Hat OpenShift documentation. You can also review the information in the Red Hat article Enabling the AWS EFS CSI Driver Operator on ROSA.

Portworx

3.1 Configure the Portworx services spec

1. Log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Next**.
2. On the *Spec Generator - Enterprise* page, enter or select the following settings for your storage:
    1. Select the checkbox for *Use the Portworx Operator*.
    2. Select **`2.10`** or higher for the Portworx Version.
    3. Select the *Built-in* radio button for ETCD. Click **Next**.
    4. Select the *Cloud* radio button. Then, select *AWS* for Cloud Platform.
    5. Select the type of disk: *Create Using a Spec*
    6. Select the EBS volume type: **`GP2`**, Size (GB): **`2000`**. Click **Next**.
    7. Click **Next** to skip the Network configuration page.
    8. On the *Customize* page, select *OpenShift 4+*. Click **Finish**.
    9. Click *Agree* to accept the license agreement.
    10. Choose your own values to enter under the *Spec Name* and *Spec Tags* fields. Then, click **Save Spec**.
3. From *Spec List* page, find your Spec name and expand the Actions column menu and select **Copy to Clipboard**. Save the kubectl command. You use this command later.

3.2 Configure the AWS infrastructure for Portworx storage

1. Edit the Inbound Rules for both your master and worker nodes to allow for Network File System (NFS).
    1. Log in to the AWS EC2 Console.
    2. Under *EC2*, select *Instances*. Click an Instance ID for one of your worker nodes.
    3. Click the **Security** tab, and click the Security Group Name for the node.
    4. Click **Edit Inbound Rules**. Scroll to the bottom, and click **Add Rule**.
    5. Update the following settings for the rule:
        - Expand the first drop-down menu, and change the setting from *Custom TCP* to *All TCP*.
        - In the *CIDR blocks Source* field, enter **`10.0.0.0/8`**.
        - For the ports, open the following ports: 111, 2049, 20048, 17001-17020, 27017.
        - If you encounter issues during the installation, considering opening all inbound ports from **`10.0.0.0/8`**.
    6. Click **Save rules**.
    7. Repeat the above steps for your other nodes.
2. Create your Portworx policy in the AWS Identity and Access Management (IAM) tool.
    1. Go to **IAM** > **Access management** > **Policies**.
    2. Choose **Create policy**.
    3. Choose the **JSON** tab.
    4. Replace all the text with the following content:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PortworxPolicy",
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:ModifyVolume",
        "ec2:DetachVolume",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:DeleteTags",
        "ec2:DeleteVolume",
        "ec2:DescribeTags",
        "ec2:DescribeVolumeAttribute",
        "ec2:DescribeVolumesModifications",
```

```
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstances",
        "autoscaling:DescribeAutoScalingGroups"
      ],
    "Resource": [
      "*"
    ]
  }
]
}
```

5. Click **Next: Tags**. Then, click **Next: Review**.
6. Enter the name for your new policy. Then, click **Create policy**.

3. Attach the policy.
    1. Go to **IAM** > **Roles**.
    2. Click the worker-role name for your cluster.
    3. Click **Attach policy**. Search for, and select, the policy that you created. Then, select to attach the policy.

3.3 Install the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your Red Hat OpenShift Service on AWS cluster.
2. Go to **Operators** > **OperatorHub**.
3. Search for Portworx Enterprise. Then, click **Install** > **Install**.
4. Portworx Enterprise is now be listed under *Installed Operators*.

3.4 Configure the Portworx operator

1. Import the Spec from Portworx.
    1. Run `oc login` to log in to your Red Hat OpenShift Service on AWS cluster.
    2. Run `kubectl apply -f <command>` where `<command>` is the command that you saved earlier from the Portworx console.
2. Verify that Portworx Enterprise shows as one of the Installed Operators in `Project: kube-system`. Click Portworx Enterprise tile.

   In the **Storage Cluster** tab, you can see the storage cluster being initialized. Wait until Status is `Phase: Online`.

3. Create the Portworx storage classes. Follow the `Define a custom Portworx storage class` instructions in Installing recommended storage providers and configuring storage classes: Portworx.

4. Install Infrastructure Automation

You are now ready to install Infrastructure Automation.

Follow the procedure for installing stand-alone Infrastructure Automation from step 3 in one of the following topics, according to your requirements:

- Online production installation of stand-alone Infrastructure Automation (console method)
- Online production installation of stand-alone Infrastructure Automation (CLI method)

Follow the entire procedure for installing {site.data.keyword.infra_automation} with Cloud Pak in one of the following topics, according to your requirements:

- Online production installation of Infrastructure Automation with Cloud Pak (console method)
- Online production installation of Infrastructure Automation with Cloud Pak (CLI method)

Deleting the Red Hat OpenShift Service on AWS cluster

If you no longer require your Red Hat OpenShift Service on AWS cluster, you can remove it with the following steps.

1. Run `oc login` to log in to your Red Hat OpenShift Service on AWS cluster.
2. Run the following command to delete the cluster:

   ```
   rosa delete cluster -c <clustername> --watch
   ```

3. Run the following command to verify that the cluster is deleted:

   ```
   rosa list clusters
   ```

   Ensure that your cluster no longer is listed.

4. Delete any Identity and Access Management (IAM) policies for the cluster.

Installing IBM Infrastructure Automation on Azure Red Hat OpenShift (ARO)

If you are installing IBM Infrastructure Automation on Azure Red Hat OpenShift (ARO), then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Infrastructure Automation on Red Hat® OpenShift® Container Platform.

Before you begin

Ensure that you meet the following prerequisites:

- Access to the Azure portal (requires email invitation).
- Red Hat subscription, with username and password.
- Azure account permissions to deploy an ARO cluster. You must have the following permissions directly on the virtual network, on the resource group, or on the subscription containing it:
  - Contributor and User Access Administrator permissions, or
  - Owner permissions

  For example, an Azure user with assigned roles of **`Application administrator`** and **`Conditional Access administrator`** has adequate permission.

Installation procedure

Follow these steps to install Infrastructure Automation on ARO.

1. Configure Azure and Red Hat access
2. Create an ARO cluster
3. Configure storage
4. Install Infrastructure Automation

1. Configure Azure and Red Hat access

1. Install the Azure command-line interface (CLI), **`az`**.

   Follow the instructions Install the Azure CLI on Linux in the Azure documentation.

2. Get an OpenShift pull secret to enable access to the OpenShift container registries.

   1. Log in to the OpenShift cluster manager portal.
   2. Click **Download pull secret** to download the pull secret, and save it to a file called **`pull-secrets.json`**. This file is used in step 2.2

3. Log in to the Azure CLI.

   Run the following command to log in to the Azure CLI, and to find your subscription-id.

   ```
   az login
   az account show | jq .id
   ```

4. Register providers to your Azure account.

   ```
   az account set --subscription <subscription-id>
   az provider register -n Microsoft.RedHatOpenShift --wait
   az provider register -n Microsoft.Compute --wait
   az provider register -n Microsoft.Storage --wait
   az provider register -n Microsoft.Authorization --wait
   az feature register --namespace Microsoft.RedHatOpenShift --name preview
   ```

   Where **`<subscription-id>`** is the subscription-id that you found in the previous step.

2. Create an ARO cluster

1. Create dependant assets.

   Run the following commands to create a virtual network that contains two empty subnets.

   ```
   # Set environment variables for use by the `az` commands
   export SUFFIX=waiops4test
   export LOCATION=eastus
   export RESOURCEGROUP=aro-$SUFFIX
   export CLUSTER=cluster-$SUFFIX

   # Create a resource group
   az group create -g $RESOURCEGROUP -l $LOCATION

   # Create a new virtual network in the resource group that you created above:
   az network vnet create -g $RESOURCEGROUP \
     -n aro-vnet-$SUFFIX \
     --address-prefixes 10.0.0.0/22

   # Add an empty subnet for the master nodes:
   az network vnet subnet create -g $RESOURCEGROUP \
     -n master-subnet-$SUFFIX \
     --vnet-name aro-vnet-$SUFFIX \
     --address-prefixes 10.0.0.0/23 \
     --service-endpoints Microsoft.ContainerRegistry \
     --disable-private-link-service-network-policies true

   # Add an empty subnet for the worker nodes:
   ```

```
az network vnet subnet create -g $RESOURCEGROUP \
  -n worker-subnet-$SUFFIX \
  --vnet-name aro-vnet-$SUFFIX \
  --address-prefixes 10.0.2.0/23 \
  --service-endpoints Microsoft.ContainerRegistry
```

2. Run the following command to create an ARO cluster.

   Note: Use worker-vm-size **Standard_D16s_v3** for a small, starter deployment of Infrastructure Automation and a worker-vm-size of **Standard_D48s_v3** for a large, production deployment of Infrastructure Automation.

```
az aro create -g $RESOURCEGROUP \
  -n $CLUSTER \
  --vnet aro-vnet-$SUFFIX \
  --master-subnet master-subnet-$SUFFIX \
  --worker-subnet worker-subnet-$SUFFIX \
  --worker-count 6 \
  --worker-vm-size Standard_D16s_v3 \
  --master-vm-size Standard_D8s_v3 \
  --pull-secret @pull-secrets.json
```

   Note: **pull-secrets.json** must be the file that the Red Hat OpenShift secret was stored in, in step 1.2.

   It takes approximately 35 minutes for the ARO cluster to be created. Progress can be viewed from the Azure console or CLI.

3. When the ARO cluster is created, run the following command to log in to it.

```
oc login \
 -u kubeadmin -p $(az aro list-credentials  -n $CLUSTER -g $RESOURCEGROUP | jq -r .kubeadminPassword) \
 -s $(az aro show  -n $CLUSTER -g $RESOURCEGROUP | jq -r .apiserverProfile.url)
```

3. Configure storage

Azure's native storage is not compatible with some Infrastructure Automation components. Configure one of the following supported RWX storage options for Infrastructure Automation:

- Red Hat® OpenShift® Data Foundation (ODF)
- Portworx Enterprise

3.1 Configuring Red Hat OpenShift Data Foundation

To configure Red Hat OpenShift Data Foundation storage for your Infrastructure Automation on ARO deployment, use the instructions in Deploying OpenShift Data Foundation on Azure Red Hat OpenShift in the Red Hat documentation.

3.2 Configuring Portworx Enterprise

To configure Portworx storage for your Infrastructure Automation on ARO deployment, use the following instructions.

3.2.1 Create secret for Portworx

1. Use a browser to access the Azure portal, **portal.azure.com**, click **Azure Active Directory** and find the value of **Tenant ID** in the returned page.
2. From the command line, run the following command to set an environment that contains the tenant ID.

```
export AZURE_TENANT_ID=<value of the tenant ID>
```

3. Use a browser to access the Azure portal, **portal.azure.com**, and then click **Virtual machines**.
4. In the search bar on the **Virtual machines** page, enter the value of your $SUFFIX environment variable as a filter and click the **Name** of any of the returned VMs.
5. Click **Access control (IAM)** on the menu.
6. On the **cluster name | Access control (IAM)** page, click **View** in the **View deny assignments** section.
7. On the **cluster name | Access control (IAM)** page's **Deny assignments** tab, click the link beneath **Name**.
8. On the **Deny assignment | Users** page, click the link in the **Name** column in the **Deny assignment excludes** section.
9. In the **enterprise application | Overview** page, locate the **Application ID** field and copy it.
10. From the command line, run the following command to set an environment variable that contains the application ID.

```
export AZURE_CLIENT_ID=<value of the application ID>
```

11. Use a browser to navigate to the Azure portal, **portal.azure.com**, and then click **Azure Active Directory**.
12. On the **Azure Active Directory | Overview** page, click **App registrations** on the menu.
13. On the **Azure Active Directory | App registrations** page, enter the cluster's client-id (that you just exported) in the filter field, and then click the link in the **Display name** column.
14. On the returned **Application registration details** page, click **Certificates and secrets** on the menu.
15. In the **Certficates and secrets** page, select **+ New client secret** to create a new client secret.

16. In the **Add a client secret** page, enter a **Description** and an **Expiry**, and select **Add**.
17. On the **Certificates and secrets** page, locate the secret that you just added in the **Client secrets** section. Then select the copy icon next to the **Value** column to get the secret.
18. From the command line, run the following command to set an environment variable that contains the client secret.

    ```
    export AZURE_CLIENT_SECRET=<value of the client secret>
    ```

19. Create a secret

    Run the following command to create the secret

    ```
    oc delete secret px-azure -n kube-system
    oc create secret generic px-azure \
    -n kube-system \
    --from-literal=AZURE_TENANT_ID=${AZURE_TENANT_ID} \
    --from-literal=AZURE_CLIENT_ID=${AZURE_CLIENT_ID} \
    --from-literal=AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
    ```

3.2.2 Create a Portworx spec

A Portworx Enterprise spec is required to set up a Portworx storage cluster.

1. From a browser, go to https://portworx.com, select **LOGIN**, and log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Continue**, and **Continue** again on the Portworx product comparison page.
2. On the **Spec Generator - Enterprise** page, enter or select the following settings and then click **Next**.
    1. Select the checkbox for **Use the Portworx Operator**.
    2. Select *2.10* for the **Portworx Version**.
    3. Select the *Built-in* radio button for **ETCD**.
3. On the next page, set the following values and then click **Next**.
    1. For **Select your environment**, select the *Cloud* radio button.
    2. Under **Select Cloud Platform**, select *AZURE*.
    3. Under **Configure storage devices**, select *Create Using a Spec*
    4. Under **Size(GB)**, select *2000 GB*
4. Click **Next** to skip the Network configuration page.
5. On the **Customize** page, select **Azure Kubernetes Service (AKS)**, then click **Finish**, and click **Agree** to accept the license agreement.
6. In the **spec** page, highlight and select the live download URL.

    For example, `https://install.portworx.com/2.10?operator=true&mc=false&kbver=&b=true&kd=type%3DPremium_LRS%2Csize%3D150&s=%22type%3DPremium_LRS%2Csize%3D2000%22&c=px-cluster-e0fb19ba-4e45-4175-b92e-18003948b965&aks=true&stork=true&csi=true&mon=true&tel=false&st=k8s&promop=true`

3.2.3 Installing the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your Red Hat OpenShift Service on AWS cluster.

    If you need to find your ARO cluster credentials and console URL, you can use the following commands.

    ```
    az aro list-credentials  -n $CLUSTER -g $RESOURCEGROUP
    az aro show  -n $CLUSTER -g $RESOURCEGROUP | jq -r .consoleProfile.url
    ```

2. Go to **Operators** > **OperatorHub**.
3. Search for *Portworx Enterprise*. Then, click **Install** > **Install**.

    Warning: Do not click *Create StorageCluster*. You will create the storage cluster in the next step, with the spec that you saved.

4. *Portworx Enterprise* is now listed under **Operators** > **Installed Operators**.

3.2.4 Create the Portworx storage cluster

1. From the command line, run the following command to load the Portworx spec that you created into the ARO cluster.

    ```
    export PORTWORX_SPEC_URL=<spec_URL>
    curl -ks "${PORTWORX_SPEC_URL}&osft=true" | oc apply -f -
    ```

    Where `<spec_URL>` is the spec URL that you copied earlier.

2. From the console, go to **Operators** > **Installed Operators** > **Portworx Enterprise** and select the **StorageClusters** tab. The storage cluster is ready when the **Status** changes from *Phase: Initializing* to *Phase: Online*.
3. Create the Portworx storage classes. Follow the `Define a custom Portworx storage class` instructions in Installing recommended storage providers and configuring storage classes: Portworx.

4. Install Infrastructure Automation

You are now ready to install Infrastructure Automation.

Follow the procedure for installing stand-alone Infrastructure Automation from step 3 in one of the following topics, according to your requirements:

- Online production installation of stand-alone Infrastructure Automation (console method)
- Online production installation of stand-alone Infrastructure Automation (CLI method)

Follow the entire procedure for installing {site.data.keyword.infra_automation} with Cloud Pak in one of the following topics, according to your requirements:

- Online production installation of Infrastructure Automation with Cloud Pak (console method)
- Online production installation of Infrastructure Automation with Cloud Pak (CLI method)

Deleting the ARO cluster

If you no longer require your ARO cluster, you can remove it with the following command.

```
az aro delete -y -g $RESOURCEGROUP -n $CLUSTER
```

Installing Infrastructure Automation for use with IBM Cloud Pak for AIOps in an air-gapped environment (offline)

If your cluster is not connected to the internet, you can complete a production install of IBM Cloud Pak® for AIOps Infrastructure Automation in your cluster by using either a bastion host, portable compute device, or a portable storage device.

The IBM Catalog Management Plug-in for IBM Cloud Pak® is now used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering IBM product images, uses standard tools for registry and cluster access, and extends the Red Hat OpenShift CLI (`oc`) capability to streamline the air-gapped deployment process.

Overview

It is common in production to have a cluster that does not have internet access. In these cases, you can still install Infrastructure Automation and Red Hat® OpenShift® Container Platform in an air-gapped (otherwise known as offline or disconnected) environment. Unlike online installations, air-gapped installations require you to enable the IBM Operator Catalog to mimic a typical online installation by using images in your own registry.

The following diagram provides an overview of air-gapped installation scenarios for portable compute, portable storage, and bastion hosts:

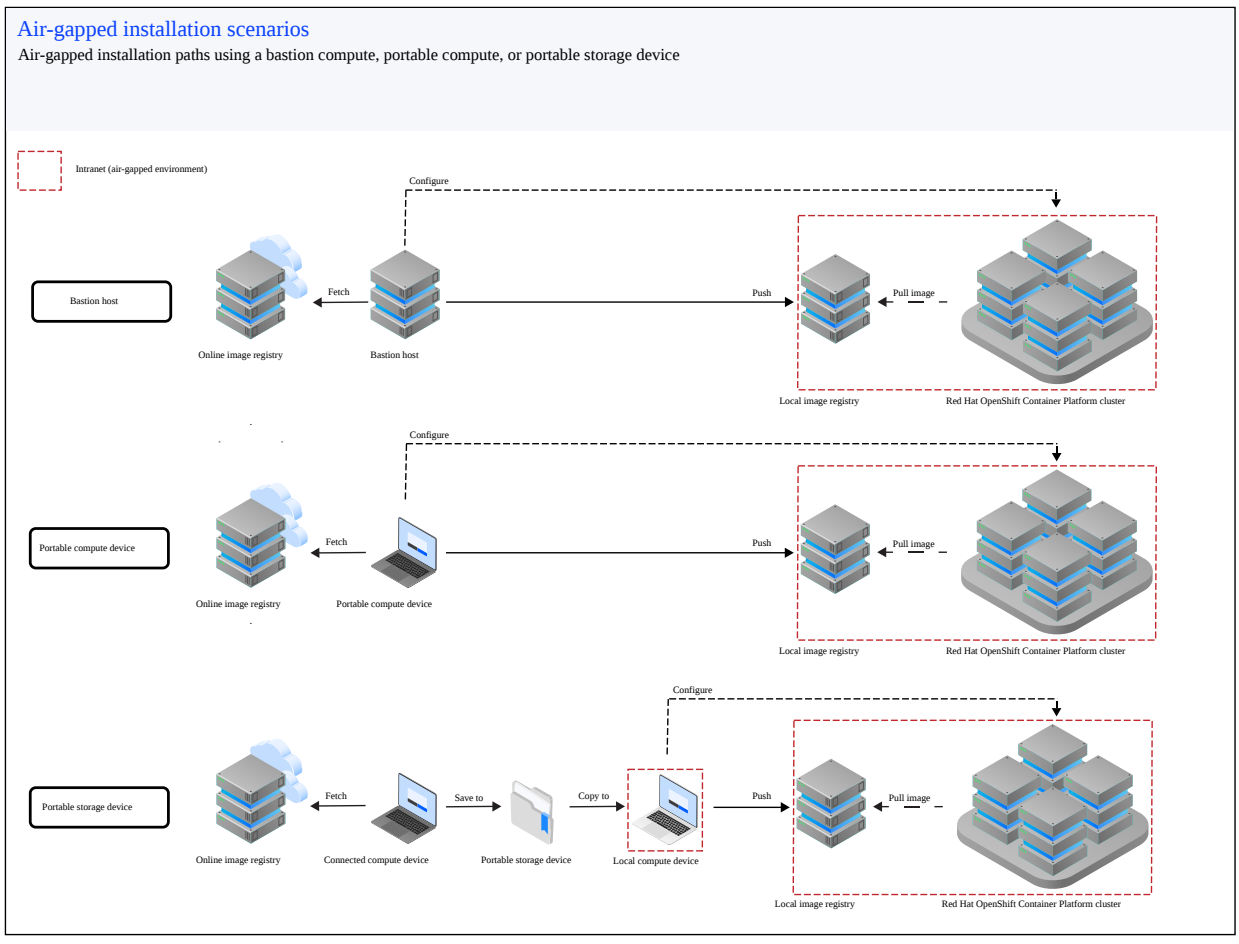


Figure. Air-gap scenario

All of these scenarios use Container Application Software for Enterprises (CASE) files to mirror content from a source to a target. CASE is a specification that defines metadata and structure for packaging, managing, and unpacking containerized applications.

You can store the product code and images to a portable compute device (like a laptop), portable storage device (like an external hard disk drive), or a bastion host and then transfer them to a local air-gapped network. If you are using either a portable compute device or a portable storage device, you don't need a bastion host to install in your air-gapped environment.

The images that are stored are large, and you must ensure that each of your airgap devices have enough storage, as follows.

- Bastion host method - the local registry and the bastion host must each have at least 62 GB of storage space.
- Portable compute device method - the local registry and the portable compute device must each have at least 62 GB of storage space.
- Portable storage device - the local registry, connected compute device, portable storage device, and the local compute device must each have at least 62 GB of storage space.

Table 1. Air-gapped installation scenarios

Air-gapped installation approach	Example	Description
Bastion host	Bastion host	A bastion server is a device that has access to both the public internet and the local intranet where a local registry and Red Hat OpenShift Container Platform clusters are located. Using the bastion server, you can replicate your images through the bastion server directly to the local, intranet registry behind the firewall.
Portable compute device	Laptop	A portable compute device, such as a laptop, can be used to download images from the entitled registry to a portable container registry running locally on the device. You can then bring the device behind your firewall and copy the images from your portable registry on the device to the local, intranet registry behind the firewall.
Portable storage device	Portable hard disk drive	Images are mirrored to a container registry on a connected compute device, and then copied to a portable storage device. The portable storage device is plugged into a local compute device inside your firewall, and the images are copied to the local compute device. The images from the local compute device are then mirrored into the local, intranet registry behind the firewall.

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your image registry access and mirroring environment (One-time action)
2. Set environment variables and download CASE files
3. Mirror images depending on installation scenario
4. Configure storage
5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

Installation methods

The following sections illustrate the required steps for each approach to an air-gapped installation. Pick the appropriate methodology and follow the steps within that section. Step ordering might vary depending on your methodology:

- Bastion host
- Portable device

Installing Infrastructure Automation in an air-gapped environment (offline) using a bastion host

If your cluster is not connected to the internet, you can deploy a production install of IBM Cloud Pak® for AIOps Infrastructure Automation on your cluster by using a bastion host.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat® OpenShift® Container Platform cluster on which Infrastructure Automation is to be installed. The bastion host has access to the internet and to the offline environment. Infrastructure Automation images are mirrored from the internet through the bastion server to the target registry in the offline environment. Infrastructure Automation can then be installed in the offline environment by using the target registry.

Before you begin

- Review the Planning section. Your environment must meet the system requirements.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the Infrastructure Automation operators need, see Permissions (Infrastructure Automation).
- The `storageClass` and `storageClassLargeBlock` that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the `storageClass` and `storageClassLargeBlock` that are used for creating the Infrastructure Automation custom resource.
- A user with `cluster-admin` privileges is needed for the following operations:
  - Create a custom project
  - Configure the cluster
  - Create the catalog source
  - Install Cert Manager
  - Install the License Service
  - Install Infrastructure Automation

Important: The following procedure is based on an Red Hat OpenShift 4.17 environment and includes links for that version. If your environment uses a different supported version of Red Hat OpenShift, ensure that you follow the Red Hat OpenShift documentation for that version.

Procedure

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your mirroring environment
2. Set environment variables and download CASE files
3. Mirror images to your final location
4. Configure storage
5. Install Infrastructure Automation

1. Set up your mirroring environment

Before you install any IBM Cloud Pak® on an air-gapped environment, you must set up a host that can be connected to the internet to complete configuring your mirroring environment. To set up your mirroring environment, complete the following steps:

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`**, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

1.1 Install and configure Red Hat OpenShift

Infrastructure Automation requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.
2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

1.2 Set up a target registry

You must have a local Docker type production-grade registry available to store the Infrastructure Automation images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.3 Prepare a host

You must prepare a bastion host that can connect to the internet and to the air-gapped network with access to the Red Hat® OpenShift® Container Platform cluster and the local, intranet Docker type registry. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak® CLI and the

Red Hat® OpenShift® Container Platform CLI support. If you are on a Windows platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux (WSL) terminal.

Your host must have 62 GB storage to hold all of the software that is to be transferred to the local, intranet Docker type registry.

The following table explains the software requirements for installing Infrastructure Automation in an air-gapped environment:

Table 2. Software requirements and purpose

Software	Purpose
OpenSSL	Validating certificates when you run the air-gapped install scripts
Podman	Container management
Apache httpd-tools	Creating an account when you run the air-gapped install scripts
Red Hat® OpenShift® Container Platform CLI (oc)	Red Hat OpenShift Container Platform administration
IBM Catalog Management Plug-in for IBM Cloud Pak	Running CASE commands

Complete the following steps on your host:

1. Install OpenSSL version 1.11.1 or higher.
2. Install Podman. For more information, see Podman Installation Instructions.

   Note: Docker is not shipped or supported by Red Hat for Red Hat Enterprise Linux (RHEL) 8 and (RHEL) 9 systems. The podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 and 9 systems. For more information, see the Red Hat documentation Running containers without Docker.
3. Install **`httpd-tools`**.

   ```
   yum install httpd-tools
   ```
4. Install the **`oc`** Red Hat® OpenShift® Container Platform CLI tool.

   `oc` is required for Red Hat OpenShift management. For more information, see Getting started with the OpenShift CLI in the Red Hat OpenShift documentation.

1.4 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**`ibm-pak-plugin`**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (**`oc`**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **`ibm-pak-plugin`** for your host operating system from github.com/IBM.
2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```
3. Run the following command to move the file to the **`/usr/local/bin`** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.
4. Confirm that the **`ibm-pak-plugin`** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The **`ibm-pak-plugin`** usage is displayed.

1.5 Download documentation and scripts for offline access

Download the following documents and scripts that you will need to be able to access offline during your Infrastructure Automation installation or uninstallation, and copy them to your air-gapped environment.

Cloud Pak for AIOps 4.8.1 documentation

Download the Cloud Pak for AIOps 4.8.1 documentation (this documentation) so that you can access it offline. You can download a PDF of the Cloud Pak for AIOps Infrastructure Automation documentation from PDF documentation.

Red Hat OpenShift Container Platform documentation

The Red Hat OpenShift Container Platform documentation can be downloaded for offline access from Red Hat.

Download the following PDFs:

- Red Hat OpenShift Container Platform 4.16 Operators PDF.
- Red Hat Operators Container Platform 4.16 Installing PDF.

IBM Cloud Paks® documentation

Download the IBM Cloud Paks® documentation so that you can access it offline. You can download a PDF of the documentation from PDF documentation.

2. Set environment variables and download CASE files

Before mirroring your images, set environment variables on your mirroring device, and connect to the internet so that you can download the corresponding CASE files. To finish preparing your host, complete the following steps.

Note: Save a copy of your environment variable values to a text editor. You can use that file as a reference to cut and paste from when completing your air-gapped installation tasks.

1. Create the following environment variables with the installer image name and the image inventory on your host.

   ```
   export IA_CASE_NAME=ibm-ia-installer
   export IA_CASE_VERSION=1.12.1
   export IA_CASE_ARCHIVE=$IA_CASE_NAME-$IA_CASE_VERSION.tgz
   export IA_CASE_INVENTORY_SETUP=ibmInfrastructureAutomationOperatorSetup
   export IA_CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_local_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_local_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   ```

   The target registry is the registry where the Infrastructure Automation images will be mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.1 Set up a target registry.

   If your bastion host must connect to the internet through a proxy, then also set the following environment variables.

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Connect your host to the internet and disconnect it from the local air-gapped network.

3. Download the Infrastructure Automation installer and image inventory to your host.

   ```
   oc ibm-pak get $IA_CASE_NAME --version $IA_CASE_VERSION
   ```

   The CASE is downloaded to **~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION**. The logs files are available at **~/.ibm-pak/logs/oc-ibm_pak.log**.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by **ibm-pak-plugin** is **~/.ibm-pak**. If required, the root directory can be configured by setting the **IBMPAK_HOME** environment variable.

4. Verify that the images are downloaded by running **ls** against your offline directory.

   ```
   ls ~/.ibm-pak/data/cases/${IA_CASE_NAME}/${IA_CASE_VERSION}
   ```

Your host is now configured and you are ready to mirror your images.

3. Mirror images to your final location

The process of mirroring images takes the image from the internet to your host, then effectively copies that image on to your air-gapped environment. After you mirror your images, you can configure your cluster and complete an air-gapped installation.

Notes:

- If you want to install subsequent updates to your air-gapped environment, you must do a CASE save to get the image list when performing those updates.
- For details on ensuring you do not introduce changes to multiple environments, see Setting up a repeatable air-gap process.
- For your mirror registry certs, include a SAN entry. In addition, configure your cluster with the certs or mark the registry as insecure. For more information about how to complete the configuration, see the Red Hat OpenShift documentation Setting up additional trusted certificate authorities for builds.

Complete the following steps to mirror the Infrastructure Automation, IBM Cloud Pak foundational services Cert Manager, and IBM Cloud Pak foundational services License Service images from the internet to the target registry in the air-gapped environment.

- 3.1 Generate mirror manifests
- 3.2 Authenticate with the registries
- 3.3 Mirror images to the host
- 3.4 Connect your host to your air-gapped environment and set up your container
- 3.5 Mirror images to final location and configure the cluster

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests $IA_CASE_NAME $TARGET_REGISTRY --version $IA_CASE_VERSION
```

A new directory named **~/.ibm-pak/mirror** is created when you issue the **oc ibm-pak generate mirror-manifests** command. The files **images-mapping.txt** and **image-content-source-policy.yaml** are generated at **~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION**.

3.2. Authenticate with the registries

Log in to the registries to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the **oc image mirror** command to pull the images from the IBM Entitled Registry, and push them to the target registry.

1. Get the authentication credentials for the IBM Entitled Registry.

1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.

2. Run the following command to create an environment variable containing your entitlement key.

   ```
   export ENTITLED_REGISTRY_PASSWORD=<key>
   ```

   Where **<key>** is the entitlement key that you copied in the previous step.

3. Store the authentication credentials for the IBM Entitled Registry and the target registry.

   Run the following commands:

   ```
   podman login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
   podman login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
   export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
   unset ENTITLED_REGISTRY_PASSWORD
   ```

   Note: The authentication file is usually at **`${XDG_RUNTIME_DIR}/containers/auth.json`**. For more information, see the Options section in the Podman documentation.

3.3. Mirror the images

Complete these steps on your host that is connected to both the local registry and the Red Hat OpenShift cluster.

Mirror images to the target registry.

```
nohup oc image mirror \
 -f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/images-mapping.txt \
 -a $REGISTRY_AUTH_FILE \
 --filter-by-os '.*' \
 --insecure \
 --skip-multiple-scopes \
 --max-per-registry=1 \
 --continue-on-error=true > my-mirror-progress.txt 2>&1 &
```

The UNIX command **`nohup`** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibilty.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress.txt
```

Note: If an error occurs during mirroring, the mirror command can be re-run.

3.4 Configure the cluster

1. Log in to your OpenShift cluster.

   You can identify your specific **`oc login`** command by clicking the user menu in the upper left corner of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **`image-content-source-policy.yaml`**, which you will apply to your cluster in the next step.

3. Create the **`ImageContentSourcePolicy`**.

   Run the following command to create the ImageContentSourcePolicy.

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/image-content-source-policy.yaml
   ```

4. Verify that the **`ImageContentSourcePolicy`** resource is created.

   ```
   oc get imageContentSourcePolicy
   ```

5. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **`ImageContentsourcePolicy`** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all of the **`MachineConfigPools`** are updated before you proceed to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **`insecureRegistries`** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing Infrastructure Automation. For more information, see Storage.

5. Install Infrastructure Automation

Now that your images are mirrored to your air-gapped environment, you can deploy Infrastructure Automation to that environment. When you mirrored your environment, you created a parallel offline version of everything that you needed to install an operator into Red Hat® OpenShift® Container Platform. To install Infrastructure Automation, complete the following steps.

- 5.1 Create a custom project
- 5.2 Create the catalog source
- 5.3 Install Cert Manager
- 5.4 Install the License Service
- 5.5 Install Infrastructure Automation

5.1 Create a custom project (namespace)

Create a project (namespace) called **cp4aiops** for your Infrastructure Automation deployment, by running the following command:

```
oc create namespace cp4aiops
```

5.2 Create the catalog source

1. Run the following command:

   ```
   oc apply -f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/catalog-sources.yaml
   ```

2. Run the following command to verify that you have all the required catalog sources created.

   ```
   oc get pods -n openshift-marketplace
   oc get catalogsource -n openshift-marketplace
   ```

   The output must include:

   ```
   cam-install-operator-controller-manager-catalog
   cloud-native-postgresql-catalog
   ibm-cert-manager-catalog
   ibm-infra-management-install-operator-catalog
   ibm-infrastructure-automation-operator-catalog
   ibm-licensing-catalog
   opencloud-operators
   ```

5.3 Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed the IBM Cloud Pak foundational services Cert Manager as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters, then you must ensure that IBM Cloud Pak foundational services Cert Manager is installed on each cluster.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

   ```
   cat << EOF | oc apply -f -
   apiVersion: v1
   kind: Namespace
   metadata:
     name: ibm-cert-manager
   ---
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: ibm-cert-manager-operator-group
     namespace: ibm-cert-manager
   ---
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-cert-manager-operator
     namespace: ibm-cert-manager
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-cert-manager-operator
     source: ibm-cert-manager-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

2. Run the following command to ensure that the IBM Cloud Pak foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-cert-manager get pods
```

Example output for a successful IBM Cloud Pak foundational services Cert Manager installation:

```
NAME                                        READY   STATUS    RESTARTS   AGE
cert-manager-cainjector-674854c49d-vstq4    1/1     Running   0          8d
cert-manager-controller-646d4bd6fd-zwmqm    1/1     Running   0          8d
cert-manager-webhook-8598787c8-s4lkt        1/1     Running   0          8d
ibm-cert-manager-operator-c96957695-dkxnm   1/1     Running   0          8d
```

5.4 Install the License Service

Skip this step if the IBM Cloud Pak foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed IBM Cloud Pak foundational services License Service as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters, then you must ensure that IBM Cloud Pak foundational services License Service is installed on each cluster.

Infrastructure Automation requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-licensing
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-licensing-operator-group
  namespace: ibm-licensing
spec:
  targetNamespaces:
  - ibm-licensing
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-licensing-operator-app
  namespace: ibm-licensing
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-licensing-operator-app
  source: ibm-licensing-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-licensing get pods
```

Example output for a successful IBM Cloud Pak foundational services License Service installation:

```
NAME                                              READY   STATUS    RESTARTS   AGE
ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5.5 Install Infrastructure Automation

If you want to install Infrastructure Automation with Cloud Pak for AIOps, follow step 5 **`Install theInfrastructure Automation operator`** and onwards in Online installation of Infrastructure Automation for use with IBM Cloud Pak for AIOps (CLI).

If you want to install stand-alone Infrastructure Automation, follow step 8 and onwards Online installation of Infrastructure Automation (CLI).

Installing Infrastructure Automation in an air-gapped environment (offline) using a portable device

If your cluster is not connected to the internet, you can deploy an installation of Infrastructure Automation on your Red Hat® OpenShift® Container Platform cluster by using a portable compute device such as a laptop, or a portable storage device such as a USB device.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat OpenShift cluster on which Infrastructure Automation is to be installed. Infrastructure Automation images are mirrored from the internet to a file system on a portable compute device or a portable storage device. The portable device is then disconnected from the internet and connected in the offline environment, where the images are loaded to the target registry. Infrastructure Automation can then be installed in the offline environment by using the target registry.

Before you begin

- Review the Planning section. Your environment must meet the system requirements.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the Infrastructure Automation operators need, see Permissions (Infrastructure Automation).
- The `storageClass` and `storageClassLargeBlock` that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the `storageClass` and `storageClassLargeBlock` that are used for creating the Infrastructure Automation custom resource.
- A user with `cluster-admin` privileges is needed for the following operations:
  - Create a custom project
  - Configure the cluster
  - Create the catalog source
  - Install Cert Manager
  - Install the License Service
  - Install Infrastructure Automation

Important: The following procedure is based on an Red Hat OpenShift 4.17 environment and includes links for that version. If your environment uses a different supported version of Red Hat OpenShift, ensure that you follow the Red Hat OpenShift documentation for that version.

Installation procedure

1. Set up the mirroring environment
2. Download the CASE
3. Mirror images
4. Configure storage
5. Install Infrastructure Automation

1. Set up the mirroring environment

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry`, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

1.1 Download documentation for offline access

Download the following documents that you might need to access during your Infrastructure Automation installation, and copy them to your air-gapped environment.

1. IBM Cloud Pak for AIOps 4.8.1 documentation

   Download the IBM Cloud Pak for AIOps 4.8.1 PDF (this documentation) so that you can access it offline.

2. Red Hat OpenShift documentation

   The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The `Security and compliance`, `Installing`, `CLI Tools`, `Images`, and `Operators` sections are referenced by this documentation.

1.2 Install and configure Red Hat OpenShift

Infrastructure Automation requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.
2. Install the Red Hat OpenShift command line interface (**oc**) on your cluster's boot node and run **oc login**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. Infrastructure Automation runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause Infrastructure Automation to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **MachineConfig** custom resource to configure chrony to connect to your NTP servers.
4. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

1.3 Set up a target registry

You must have a local Docker type production-grade registry available in the air-gapped environment to store the Infrastructure Automation images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 62 GB of storage to hold all of the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.4 Prepare a host

Prepare a portable compute device, or if you are using a portable storage device then a connected compute device, as follows.

You must be able to connect your host to the internet. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak CLI and the Red Hat OpenShift CLI support. If you are on a Windows® platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux® (WSL) terminal.

Your portable device and any intermediary devices must have 62 GB of storage to hold all the software that is to be transferred to the local target registry.

Complete the following steps on your host.

1. Install Podman, see the Podman Installation Instructions.

   Note: Docker is not shipped or supported for Red Hat Enterprise Linux (RHEL) 8 and (RHEL) 9. The Podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 and 9 systems. For more information, see Running containers without Docker in the Red Hat documentation.

2. Install the Red Hat OpenShift CLI tool, **oc**.

   oc is required for Red Hat OpenShift management. For more information, see Getting started with the Red Hat OpenShift CLI in the Red Hat OpenShift documentation.

1.5 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**ibm-pak-plugin**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (**oc**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **ibm-pak-plugin** for your host operating system from github.com/IBM.
2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the **/usr/local/bin** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that **ibm-pak-plugin** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

Expected result: The `ibm-pak-plugin` usage is displayed.

2. Download the CASE

Set environment variables on the portable device, and connect it to the internet so that you can download the Infrastructure Automation CASE files.

Note: Save a copy of your environment variable values to a file that you can use as a reference when you are completing your air-gapped installation tasks.

1. Create the following environment variables.

```
export IA_CASE_NAME=ibm-ia-installer
export IA_CASE_VERSION=1.12.1
export IA_CASE_INVENTORY_SETUP=ibmInfrastructureAutomationOperatorSetup
export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
```

   The target registry is the registry where the Infrastructure Automation images are mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.3 Set up a target registry.

   If your portable device must connect to the internet through a proxy, then also set the following environment variables.

```
export https_proxy=http://proxy-server-hostname:port
export http_proxy=http://proxy-server-hostname:port
```

2. Connect your portable device to the internet and disconnect it from the air-gapped environment.

3. Download the Infrastructure Automation installer and image inventory to your portable device.

```
oc ibm-pak get ${IA_CASE_NAME} --version ${IA_CASE_VERSION}
```

   The CASE is downloaded to `~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION`. The log files are available at `~/.ibm-pak/logs/oc-ibm_pak.log`.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by the `ibm-pak-plugin` is `~/.ibm-pak`. If required, the root directory can be configured by setting the `IBMPAK_HOME` environment variable.

3. Mirror images

Complete the following steps to mirror the Infrastructure Automation, IBM Cloud Pak foundational services Cert Manager, and IBM Cloud Pak foundational services License Service images from the internet to the file system on your portable device, and then from the file system to the target registry in the air-gapped environment.

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests ${IA_CASE_NAME} file://cpfs --version ${IA_CASE_VERSION} --final-registry ${TARGET_REGISTRY}/cpfs
```

Table 2. Parameter description

Argument	Description
`file://cpfs`	Specifies the path extension where images are mirrored to. Images are mirrored to `$IMAGE_PATH/cpfs` when the `oc image mirror` command is run with `images-mapping-to-filesystem.txt`. For more information, see Mirror the images to the file system.
`$TARGET_REGISTRY/cpfs`	This argument generates an image-mapping file that is used by the `oc image mirror` commands to mirror images to the `TARGET_REGISTRY` at namespace `cpfs`. For example, if the CASE you are installing needs the image, `quay.io/opencloudio/ibm-events-kafka-2.6.0@sha256:10d422dddd29ff19c87066fc6510eee05f5fa4ff608b87a06e898b3b6a3a13c7`, then its final URL in your target registry will be `$TARGET_REGISTRY/cpfs/opencloudio/ibm-events-kafka-2.6.0`. Note the new namespace of `cpfs` in the URL. The namespace path can be multi level if your target registry supports it.

The command generates the following files at `~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION`:

- images-mapping-to-filesystem.txt
- images-mapping-from-filesystem.txt
- image-content-source-policy.yaml

3.2. Authenticate with the IBM Entitled Registry

Log in to the IBM Entitled Registry to generate an authentication file containing the IBM Entitled Registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the `oc image mirror` command to pull the images from the IBM Entitled Registry.

1. Get the authentication credentials for the IBM Entitled Registry.

   1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

   2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.

2. Run the following command to create an environment variable that contains your entitlement key.

```
export ENTITLED_REGISTRY_PASSWORD=<key>
```

Where **<key>** is the entitlement key that you copied in the previous step.

3. Store the authentication credentials for the IBM Entitled Registry.

   Run the following command:

   ```
   podman login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
   export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
   unset ENTITLED_REGISTRY_PASSWORD
   ```

   Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

3.3. Mirror the images to the file system

Complete these steps to mirror the images from the internet to a file system on your portable device.

1. Create an environment variable to store the location of the file system where the images are to be stored.

   ```
   export IMAGE_PATH=<image-path>
   ```

   Where **<image-path>** is the directory where you want the images to be stored.

2. Run the following command to mirror the images from the IBM Entitled Registry to the file system.

   ```
   nohup oc image mirror \
   -f ~/.ibm-pak/data/mirror/${IA_CASE_NAME}/${IA_CASE_VERSION}/images-mapping-to-filesystem.txt \
   -a ${REGISTRY_AUTH_FILE} \
   --filter-by-os '.*' \
   --insecure \
   --skip-multiple-scopes \
   --dir "${IMAGE_PATH}" \
   --max-per-registry=1 > my-mirror-progress.txt 2>&1 &
   ```

   The UNIX® command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

   Run the following command if you want to see the progress of the mirroring:

   ```
   tail -f my-mirror-progress.txt
   ```

   Note: If an error occurs during mirroring, the mirror command can be rerun.

3.4 Setup the file system in the air-gapped environment

1. Copy files to the air-gapped environment (portable storage device only)

   If you are using a portable storage device, you must copy the files from the portable storage device to a local compute device in the air-gapped environment that has access to the target registry. If you are using a portable compute device, then these items are already present, and you can proceed to the next step.

   Copy the following items to your local compute device:

   - the file system located at $IMAGE_PATH, which you specified earlier
   - **~/.ibm-pak** directory
2. Disconnect the device that has your file system, (the portable compute device or the local compute device) from the internet and connect it to the air-gapped environment.

3. Ensure that environment variables are set on the device in the air-gapped environment that has access to the target registry.

   If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

   If you are using a portable compute device that you have restarted since mirroring the images, then your environment variables will have been lost and you will need to set the following environment variables on your portable compute device again.

   ```
   export IA_CASE_NAME=ibm-ia-installer
   export IA_CASE_VERSION=1.12.1
   export IA_CASE_INVENTORY_SETUP=ibmInfrastructureAutomationOperatorSetup
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=${TARGET_REGISTRY_HOST}:${TARGET_REGISTRY_PORT}
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export IMAGE_PATH=<image_path>
   ```

3.5 Authenticate with the target registry

Authenticate with the target registry in the air-gapped environment that you are mirroring the images into.

Run the following command:

```
podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

3.6 Mirror the images to the target registry from the file system

Complete the steps in this section on the device that has your file system (the portable compute device or the local compute device) to copy the images from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat OpenShift cluster.

Run the following command to copy the images referenced in the **images-mapping-from-filesystem.txt** from the $IMAGE_PATH file system to the final target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/${IA_CASE_NAME}/${IA_CASE_VERSION}/images-mapping-from-filesystem.txt \
--from-dir "${IMAGE_PATH}" \
-a ${REGISTRY_AUTH_FILE} \
--filter-by-os '.*' \
--insecure  \
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress2.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress2.txt
```

Note: If an error occurs during mirroring, the mirror command can be rerun.

3.7 Configure the cluster

1. Log in to your Red Hat OpenShift cluster.

   You can identify your specific **oc login** command by clicking the user menu in the upper left corner of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **image-content-source-policy.yaml**, which you will apply to your cluster in the next step.

3. Create the **ImageContentSourcePolicy**.

   Run the following command to create an ImageContentSourcePolicy (ICSP).

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/${IA_CASE_NAME}/${IA_CASE_VERSION}/image-content-source-policy.yaml
   ```

4. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **ImageContentsourcePolicy** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all of the **MachineConfigPools** are updated before proceeding to the next step.

5. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **insecureRegistries** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing Infrastructure Automation. For more information, see Storage.

5. Install Infrastructure Automation

Now that the images are mirrored to your air-gapped environment, you can deploy Infrastructure Automation to that environment. To install Infrastructure Automation, complete the following steps.

- 5.1 Create the catalog source
- 5.2 Create a custom project
- 5.3 Install Cert Manager
- 5.4 Install the License Service
- 5.5 Install Infrastructure Automation

5.1 Create a custom project (namespace)

Run the following command to create a project (namespace) called **cp4aiops** to deploy Infrastructure Automation into.

```
oc create namespace cp4aiops
```

5.2 Create the catalog source

1. Run the following command:

```
oc apply -f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/catalog-sources.yaml
```

2. Run the following command to verify that you have all the required catalog sources created.

```
oc get pods -n openshift-marketplace
oc get catalogsource -n openshift-marketplace
```

The output must include:

```
cam-install-operator-controller-manager-catalog
cloud-native-postgresql-catalog
ibm-cert-manager-catalog
ibm-infra-management-install-operator-catalog
ibm-infrastructure-automation-operator-catalog
ibm-licensing-catalog
opencloud-operators
```

5.3 Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed a certificate manager as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters, then you must ensure that a certificate manager is installed on each cluster.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
  source: ibm-cert-manager-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-cert-manager get pods
```

Example output for a successful IBM Cloud Pak foundational services Cert Manager installation:

```
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
cert-manager-webhook-8598787c8-s4lkt       1/1     Running   0          8d
ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
```

5.4 Install the License Service

Skip this step if the IBM Cloud Pak foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

Note: If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on the same Red Hat OpenShift cluster, then you already installed the IBM Cloud Pak foundational services License Service as part of the IBM Cloud Pak for AIOps installation process. If you are installing Infrastructure Automation and IBM Cloud Pak for AIOps on different clusters, then you must ensure that IBM Cloud Pak foundational services License Service is installed on each cluster.

Infrastructure Automation requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing Infrastructure Automation on.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-licensing
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-licensing-operator-group
  namespace: ibm-licensing
spec:
  targetNamespaces:
  - ibm-licensing
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-licensing-operator-app
  namespace: ibm-licensing
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-licensing-operator-app
  source: ibm-licensing-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-licensing get pods
```

Example output for a successful IBM Cloud Pak foundational services License Service installation:

```
NAME                                                  READY   STATUS    RESTARTS   AGE
ibm-licensing-operator-db4cd746c-xzmlf                1/1     Running   0          8d
ibm-licensing-service-instance-596b99588f-76cc5       1/1     Running   0          8d
```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5.5 Install Infrastructure Automation

If you want to install Infrastructure Automation with Cloud Pak for AIOps, follow step 5 **`Install theInfrastructure Automation operator`** and onwards in Online installation of Infrastructure Automation for use with IBM Cloud Pak for AIOps (CLI).

If you want to install stand-alone Infrastructure Automation, follow step 8 and onwards Online installation of Infrastructure Automation (CLI).

Installing and upgrading Infrastructure Management components

Use the following instructions to install and upgrade Infrastructure Automation - Infrastructure Management. Install Infrastructure Management for use with Cloud Pak for AIOps or just install Infrastructure Management as a stand-alone component.

Install the Infrastructure Management appliance

You can use the following steps to install Infrastructure Management as an appliance:

1. If you are installing Infrastructure Management for use with Cloud Pak for AIOps then use the listed prerequisites steps. For more information, see Prerequisites for use with Cloud Pak for AIOps.

   Skip this step if you only want to install Infrastructure Management and do not want to use Infrastructure Management with Cloud Pak for AIOps.

2. Download the Infrastructure Management appliance package for your environment. For more information, see Download the Infrastructure Management appliance.

3. Set up a virtual machine based on the appliance and configure the Infrastructure Management appliance. Follow the steps in the installation topic for your platform:

   - Installing Infrastructure Management on Amazon Elastic Compute Cloud (Amazon EC2)
   - Installing Infrastructure Management on VMware vSphere
   - Installing Infrastructure Management on Red Hat OpenStack Platform
   - Installing Infrastructure Management on Red Hat Virtualization
   - Installing Infrastructure Management on Google Compute Engine

4. Configure OIDC integration between IBM Cloud Pak® and the Infrastructure Management appliance. For more information, see Configure OIDC integration between IBM Cloud Pak® and the Infrastructure Management appliance.

   Skip this step if you only want to install Infrastructure Management and do not want to use Infrastructure Management with Cloud Pak for AIOps.

Note: Only OIDC-based authentication (OpenID Connect) is supported. The configuration of OpenID Connect (OIDC) is required for integration with Infrastructure Management and IBM Cloud Pak®.

5. For more information about usage data collection and the recommended configuration files, see Enable usage data collection (optional).

Prerequisites for use with Cloud Pak for AIOps

Note: Skip the *Prerequisites for use with Cloud Pak for AIOps* section if you only want to install Infrastructure Management and do not want to use Infrastructure Management with Cloud Pak for AIOps.

- Ensure you enable the operators for **`Infrastructure management`** by opening the installation YAML file. Locate the **`pakModules`** section, and change **`enabled: false`** to **`enabled: true`**. Enable these Infrastructure Management-related operators:
  - **`ibm-management-im-install`** for Infrastructure Management has been installed. For more information, see Infrastructure management.
- You must configure and connect an LDAP directory with IBM Cloud Pak®. You must have an LDAP group in your configuration for IBM Cloud Pak with users defined who will access Infrastructure Management.

Downloading Infrastructure Management packages

Complete the following procedure to download your packages:

1. Go to the IBM® Passport Advantage® Online tab at Passport Advantage and log in with your IBM ID.
2. Find your part number.
3. Search for the files by using the part numbers that are listed in the table.
4. Download the files to a directory on your system.

For more information about the available packages for Infrastructure Automation, including for Infrastructure Management, see Passport Advantage part numbers.

Enable usage data collection (optional)

To help the development of Infrastructure Automation - Infrastructure Management, aggregated usage data is collected to analyze how Infrastructure Management is used. The collection of usage data is enabled by default, but can be disabled.

For the data collection, Infrastructure Management uses the existing daily job that is used for audit logging of managed resources and for licensing tracking. This job is extended to collect and send the usage data metrics to IBM. The sent data is then stored in IBM controlled GDPR-compliant systems. The usage data that is collected does not include personal information, passwords, or specific details. Only the following data is collected:

- The number of virtual machines (VMs), hosts, providers (and provider types), services, and service catalog items that are being used in the Infrastructure Management inventory.
- Whether Infrastructure Management is deployed as a containerized deployment (podified) or virtual machine appliance.
- (Stand-alone deployments) The architecture where Infrastructure Management is deployed (Linux x86_64, Linux on Power (ppc64le), Linux® on IBM Z® and LinuxONE (s390x)).

Disabling the collection of usage data

If you want to disable the collection of usage data, create the following configuration files:

1. Create the **`/run/secrets/segment`** directory.

   ```
   mkdir -p /run/secrets/segment
   ```

2. Create the file with your company name:

   ```
   echo "<COMPANY_NAME>" > /run/secrets/segment/CUSTOMER_NAME
   ```

   Where **`<COMPANY_NAME>`** is the name of your company.

3. Create the file with your customer number:

   ```
   echo "<CUSTOMER_NUMBER>" > /run/secrets/segment/CUSTOMER_ICN
   ```

   Where **`<CUSTOMER_NUMBER>`** is your IBM customer number.

4. Create the file with your environment name:

   ```
   echo "<ENV>" > /run/secrets/segment/ENVIRONMENT
   ```

   Where **`<ENV>`** can be **`trial`**, **`poc`** or **`production`**.

To opt out of usage data collection, create the following configuration file:

```
echo "false" > /run/secrets/segment/ENABLE_COLLECTION
```


- Installing on Amazon Elastic Compute Cloud (Amazon EC2)
  - Installing Infrastructure Automation
    - Obtaining the appliance
    - Requirements
    - Infrastructure Automation Requirements
      - Amazon EC2 Requirements
    - Uploading the Appliance to an Amazon S3 Bucket

        - Configuring Amazon EC2 to Import the Appliance
        - Importing the Appliance to Amazon Elastic Compute Cloud (Amazon EC2)
    - Configuring Infrastructure Automation
        - Accessing the Appliance Console
        - Configuring a Database
            - Configuring an Internal Database
            - Configuring an External Database
        - Configure Messaging
        - Configuring a Worker Appliance
- Appendix
- Appliance Console Command-Line Interface (CLI)
    - Database Configuration Options
    - Messaging Configuration Options
    - v2_key Options
    - IPA Server Options
    - Certificate Options
    - Other Options

Installing on Amazon Elastic Compute Cloud (Amazon EC2)

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine snapshot template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

Obtaining the appliance

Requirements

Below are the two sets of requirements for installing Infrastructure Automation on Amazon EC2.

Infrastructure Automation Requirements

- 44 GB of space on the chosen datastore.
- 12 GB RAM.
- 4 vCPUs.
- Check that there are inbound rules to access ports 443, 22, and 80.

Amazon EC2 Requirements

- An Amazon S3 bucket to store the disk image that will be imported to AWS as a snapshot.
- A VM import service role (IAM role) named `vmimport`.

For information on creating an Amazon S3 bucket and a VM Import Service Role, see the Amazon EC2 documentation.

Uploading the Appliance to an Amazon S3 Bucket

From your local file system, you can now upload the Infrastructure Automation appliance VHD image obtained in Obtaining the appliance to the Amazon S3 bucket, using your choice of tool, such as AWS client or directly upload the Appliance image from your S3 bucket on AWS Management Console.

Configuring Amazon EC2 to Import the Appliance

Important:

These are the procedural steps as of the time of writing. For the latest information on importing a virtual machine as an image, see the Amazon EC2 documentation.

1. Install the AWS client on the computer you want to interact with the AWS API from.

   ```
   $ pip install awscli
   ```

2. Configure and download your AWS secret/access key by following the steps in the Managing Access Keys for Your AWS Account documentation.
3. Configure the AWS client with your access/secret key. For example:

```
$ aws configure
AWS Access Key ID [******]: ACCESS_KEY
AWS Secret Access Key [******]: SECRET_KEY
Default region name [None]:
Default output format [None]:
```

4. Create the **trust-policy.json** file for the **vmimport** role. For example:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": { "Service": "vmie.amazonaws.com" },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals":{
               "sts:Externalid": "vmimport"
            }
         }
      }
   ]
}
```

5. Create the **vmimport** role using the **trust-policy.json** file that you just created.

```
$ aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
```

Note:

This user must have permissions to create and modify IAM roles.

6. Create the **role-policy.json** file. Be sure to use the exact S3 bucket name. For example:

```
{
  "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "s3:ListAllMyBuckets"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "s3:CreateBucket",
       "s3:DeleteBucket",
       "s3:DeleteObject",
       "s3:GetBucketLocation",
       "s3:GetObject",
       "s3:ListBucket",
       "s3:PutObject",
       "s3:GetBucketAcl"
     ],
     "Resource": ["arn:aws:s3:::BUCKET_TO_UPLOAD_IMAGE","arn:aws:s3:::BUCKET_TO_UPLOAD_IMAGE/*"]
   },
   {
     "Effect": "Allow",
     "Action": [
       "iam:CreateRole",
       "iam:PutRolePolicy"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "ec2:ModifySnapshotAttribute",
       "ec2:CopySnapshot",
       "ec2:RegisterImage",
       "ec2:Describe*"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "ec2:CancelConversionTask",
       "ec2:CancelExportTask",
       "ec2:CreateImage",
       "ec2:CreateInstanceExportTask",
       "ec2:CreateTags",
       "ec2:DeleteTags",
       "ec2:DescribeConversionTasks",
       "ec2:DescribeExportTasks",
       "ec2:DescribeExportImageTasks",
       "ec2:DescribeInstanceAttribute",
       "ec2:DescribeInstanceStatus",
       "ec2:DescribeInstances",
```

```
        "ec2:DescribeTags",
        "ec2:ExportImage",
        "ec2:ImportInstance",
        "ec2:ImportVolume",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances",
        "ec2:ImportImage",
        "ec2:ImportSnapshot",
        "ec2:DescribeImportImageTasks",
        "ec2:DescribeImportSnapshotTasks",
        "ec2:CancelImportTask",
        "ec2:DescribeImages",
        "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
    }
 ]
}
```

7. Apply the **vmimport** role to the Infrastructure Automation appliance image you uploaded to the S3 bucket.

```
$ aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://role-policy.json
```

Importing the Appliance to Amazon Elastic Compute Cloud (Amazon EC2)

To import the appliance:

1. Create a containers.json file:

```
{
    "Description": "NAME OF IMPORTED SNAPSHOT IN AWS",
    "Format": "vhd",
    "UserBucket": {
        "S3Bucket": "BUCKET WITH UPLOADED .VHD IMAGE",
        "S3Key": "PATH OF .VHD IMAGE"
    }
}
```

   See the AWS documentation on VM import and export requirements, such as image formats, instances, volume and file system types, and using regions.

2. Use the AWS-CLI tools to import a disk as a snapshot. See the AWS documentation on using VM Import/Export to import a disk as a snapshot.

   Note:

   You can either specify a region, or if not, ensure that the S3 bucket is in the same region where you want to import the snapshot.

```
$ aws ec2 import-snapshot --disk-container file://containers.json
```

3. Check the progress of your snapshot import by running the following command:

```
$ aws ec2 describe-import-snapshot-tasks --import-task-ids SNAPSHOT_ID_GOT_FROM_RESPONSE
```

4. Create an AMI from the snapshot. See the AWS documentation on using options with the following command to create and register a Linux AMI from a snapshot.

```
$ aws ec2 register-image
```

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Press the number for the item that you want to change, and press **Enter**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **Enter** to accept a setting where applicable.

Note: The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
    - If this is the first Infrastructure Automation appliance, choose **Create key**.
    - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

      Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

    - Enter **1** to choose **`/dev/vdb`** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **`/var/lib/pgsql`**, which can be found in the environment variable **`$APPLIANCE_PG_MOUNT_POINT`**.
    - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).
8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
    - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
    - Select **N** to configure the appliance with the full administrative user interface.
9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you need to configure the appliance to use an external PostgreSQL database. For example, you can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **`postgresql.conf`** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **`postgresql.conf`** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **`postgresql.conf`** to suit your system. For example, customize the **`shared_buffers`** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances that are expected to connect to the PostgreSQL instance, it might be necessary to alter the **`max_connections`** setting.

Note:

`- Infrastructure Automation requires PostgreSQL version 13.`

`- `postgresql.conf` controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.`

Use the following steps to configure an external PostgreSQL database:

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The appliance console summary screen is displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**`vmdb_production`**).
8. Enter the database username or leave blank for the default (**`root`**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation configures an external database.

Configure Messaging

Configuring messaging is required for the appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one Kafka broker per region.

1. You can either configure the current appliance as a Kafka broker, or point the appliance to an existing external Kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external Kafka broker. Fill in the required **Message Client Parameters**, such as IP address, username and password.
2. Select **Proceed**. **`appliance_console`** applies the configuration that you requested.
3. Restart **`evmserverd`** to pick up the changes.

Note: Use your Fully Qualified Domain Name (FQDN) as the messaging hostname rather than **`localhost`**, ensure that a resolvable and reachable non-localhost name entry is present in **`/etc/hosts`**.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**`5432`**).
9. Enter the database name or leave blank for the default (**`vmdb_production`**).
10. Enter the database username or leave blank for the default (**`root`**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region
14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Note: When configuring messaging, please refer to Amazon EC2 instance hostname types on hostname guidelines. For example, inclusion of **`.ec2.internal`** or **`.compute.internal`** in the hostname, which can be determined using **`hostname -f`**.

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**
2. Enter the default credentials as listed for the initial login.
   - **Username**: **admin**
   - **Password**: **smartvm**
3. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **`appliance_console_cli`** feature is a subset of the full functionality of the **`appliance_console`** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **`root`** and the default password of **`smartvm`**. This displays the Bash prompt for the root user.
2. Enter the **`appliance_console_cli`** or **`appliance_console_cli --help`** command to see a list of options available with the command, or simply enter **`appliance_console_cli --option <argument>`** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **`5432`**)
–username (-U)	database username (defaults to **`root`**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **`vmdb_production`**)

Messaging Configuration Options

| | | | | ———————— | ———————————— | | Option | Description | | –message-server-config | configure appliance as a messaging server (boolean) | | –message-server-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-client-config | configure appliance as a messaging server (boolean) | | –message-client-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-keystore-username | messaging keystore username (defaults to admin) | | –message-keystore-password | messaging keystore password | | –message-server-username | messaging server username (defaults to admin) | | –message-server-password | messaging server password | | –message-server-port | messaging server port (defaults to 9093) | | –message-server-use-ipaddr | use ipaddress not hostname for messaging server | | –message-server-host | messaging server hostname or ipaddress | | –message-truststore-path-src | path for messaging server truststore | | –message-ca-cert-path-src | path to a certificate authority for messaging SSL | | –message-persistent-disk | configure a new volume for storing messaging data |

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **`root`**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **`admin`**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

Set up an all-in-one appliance as a local database and messaging appliance and start the MIQ Server:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm --message-server-config --message-keystore-password smartvm --server=start
```

Join a worker appliance to an existing region:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm --server start
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword
```

To configure an appliance as a messaging server:

```
# appliance_console_cli --message-server-config --message-keystore-password="smartvm"
```

To configure an appliance as a messaging client:

```
# appliance_console_cli --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on VMware vSphere
  - Installing Infrastructure Automation
    - Obtaining the appliance
    - Uploading the Appliance on VMware vSphere
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database

        - Configuring an Internal Database
        - Configuring an External Database
      - Configure Messaging
      - Configuring a Worker Appliance
    - Additional Configuration for Appliances on VMware vSphere
      - Tuning Appliance Performance
    - Logging In After Installing Infrastructure Automation
      - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
    - Database Configuration Options
    - Messaging Configuration Options
    - v2_key Options
    - IPA Server Options
    - Certificate Options
    - Other Options

Installing on VMware vSphere

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

Obtaining the appliance

Uploading the Appliance on VMware vSphere

Uploading the Infrastructure Automation appliance file onto VMware vSphere systems has the following requirements:

- 44 GB of space on the chosen vSphere datastore.
- 12 GB RAM.
- 4 VCPUs.
- Administrator access to the vSphere Client.
- Depending on your infrastructure, allow time for the upload.

Note:

These are the procedural steps as of the time of writing. For more information, consult the “Deploying an OVF or OVA Template” topic in the VMware documentation.

Use the following procedure to upload the Infrastructure Automation appliance OVF template from your local file system using the vSphere Client.

1. In the vSphere Client, select **File > Deploy OVF Template**. The Deploy OVF Template wizard appears.
2. Specify the source location and click **Next**.
    - Select **Deploy from File** to browse your file system for the OVF template, for example `manageiq-vsphere-ivanchuk-4.ova`.
    - Select **Deploy from URL** to specify a URL to an OVF template located on the internet.
3. View the **OVF Template Details** page and click **Next**.
4. Select the deployment configuration from the drop-down menu and click **Next**. The option selected typically controls the memory settings, number of CPUs and reservations, and application-level configuration parameters.
5. Select the host or cluster on which you want to deploy the OVF template and click **Next**.
6. Select the host on which you want to run the appliance, and click **Next**.
7. Navigate to, and select the resource pool where you want to run the Infrastructure Automation appliance and click **Next**.
8. Select a datastore to store the deployed Infrastructure Automation appliance, and click Next. Ensure to select a datastore large enough to accommodate the virtual machine and all of its virtual disk files.
9. Select the disk format to store the virtual machine virtual disks, and click **Next**.

- Select **Thin Provisioned** if the storage is allocated on demand as data is written to the virtual disks.
- Select **Thick Provisioned** if all storage is immediately allocated.

10. For each network specified in the OVF template, select a network by right-clicking the **Destination Network** column in your infrastructure to set up the network mapping and click **Next**.
11. The **IP Allocation** page does not require any configuration changes. Leave the default settings in the **IP Allocation** page and click **Next**.
12. Set the user-configurable properties and click **Next**. The properties to enter depend on the selected IP allocation scheme. For example, you are prompted for IP related information for the deployed virtual machines only in the case of a fixed IP allocation scheme.
13. Review your settings and click **Finish**.

The progress of the import task appears in the vSphere Client Status panel.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the `appliance_console` command. The Infrastructure Automation appliance summary screen displays.
3. Press `Enter` to manually configure settings.
4. Press the number for the item that you want to change, and press `Enter`. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press `Enter` to accept a setting where applicable.

Note: The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the `appliance_console` command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
   - If this is the first Infrastructure Automation appliance, choose **Create key**.
   - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

     Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see `Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.` message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

```
1) /dev/vdb: 20480
2) Don't partition the disk
```

- Enter **1** to choose **/dev/vdb** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **/var/lib/pgsql**, which can be found in the environment variable **$APPLIANCE_PG_MOUNT_POINT**.
- Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).

8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
   - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
   - Select **N** to configure the appliance with the full administrative user interface.
9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you need to configure the appliance to use an external PostgreSQL database. For example, you can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **postgresql.conf** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **postgresql.conf** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **postgresql.conf** to suit your system. For example, customize the **shared_buffers** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances that are expected to connect to the PostgreSQL instance, it might be necessary to alter the **max_connections** setting.

Note:

```
- Infrastructure Automation requires PostgreSQL version 13.
```

```
- `postgresql.conf` controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.
```

Use the following steps to configure an external PostgreSQL database:

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The appliance console summary screen is displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**vmdb_production**).
8. Enter the database username or leave blank for the default (**root**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation configures an external database.

Configure Messaging

Configuring messaging is required for the appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one Kafka broker per region.

1. You can either configure the current appliance as a Kafka broker, or point the appliance to an existing external Kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external Kafka broker. Fill in the required **Message Client Parameters**, such as IP address, username and password.
2. Select **Proceed**. **appliance_console** applies the configuration that you requested.
3. Restart **evmserverd** to pick up the changes.

Note: Use your Fully Qualified Domain Name (FQDN) as the messaging hostname rather than **`localhost`**, ensure that a resolvable and reachable non-localhost name entry is present in **`/etc/hosts`**.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**`5432`**).
9. Enter the database name or leave blank for the default (**`vmdb_production`**).
10. Enter the database username or leave blank for the default (**`root`**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region
14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Additional Configuration for Appliances on VMware vSphere

Tuning Appliance Performance

By default, the Infrastructure Automation appliance uses the **`tuned`** service and its **`virtual-guest`** profile to optimize performance. In most cases, this profile provides the best performance for the appliance.

However on some VMware setups (for example, with a large vCenter database), the following additional tuning may further improve appliance performance:

- When using the **`virtual-guest`** profile in **`tuned`**, edit the **`vm.swappiness`** setting to **`1`** in the **`tuned.conf`** file from the default of **`vm.swappiness = 30`**.
- Use the **`noop`** scheduler instead. See the VMware documentation for more details on the best scheduler for your environment. See Setting the Default I/O Scheduler in the Red Hat Enterprise Linux *Performance Tuning Guide* for instructions on changing the default I/O scheduler.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**
2. Enter the default credentials as listed for the initial login.
   - **Username**: **admin**
   - **Password**: **smartvm**
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**.
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.

6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **appliance_console_cli** feature is a subset of the full functionality of the **appliance_console** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **root** and the default password of **smartvm**. This displays the Bash prompt for the root user.
2. Enter the **appliance_console_cli** or **appliance_console_cli --help** command to see a list of options available with the command, or simply enter **appliance_console_cli --option <argument>** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **5432**)
–username (-U)	database username (defaults to **root**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **vmdb_production**)

Messaging Configuration Options

| | | | ———————— | ———————————————— | | Option | Description | | –message-server-config | configure appliance as a messaging server (boolean) | | –message-server-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-client-config | configure appliance as a messaging server (boolean) | | –message-client-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-keystore-username | messaging keystore username (defaults to admin) | | –message-keystore-password | messaging keystore password | | –message-server-username | messaging server username (defaults to admin) | | –message-server-password | messaging server password | | –message-server-port | messaging server port (defaults to 9093) | | –message-server-use-ipaddr | use ipaddress not hostname for messaging server | | –message-server-host | messaging server hostname or ipaddress | | –message-truststore-path-src | path for messaging server truststore | | –message-ca-cert-path-src | path to a certificate authority for messaging SSL | | –message-persistent-disk | configure a new volume for storing messaging data |

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **root**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

Set up an all-in-one appliance as a local database and messaging appliance and start the MIQ Server:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm --message-server-config --message-keystore-password smartvm --server=start
```

Join a worker appliance to an existing region:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm --server start
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword
```

To configure an appliance as a messaging server:

```
# appliance_console_cli --message-server-config --message-keystore-password="smartvm"
```

To configure an appliance as a messaging client:

```
# appliance_console_cli --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on Red Hat OpenStack Platform
  - Installing Infrastructure Automation
    - Obtaining the appliance
    - Uploading the Appliance on OpenStack
    - Adding a Rule to a Security Group
    - Creating a Custom Flavor
    - Launching the Instance
    - Adding a Floating IP Address
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database
      - Configuring an Internal Database
      - Configuring an External Database

        - Configure Messaging
        - Configuring a Worker Appliance
    - Logging In After Installing Infrastructure Automation
        - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
    - Database Configuration Options
    - Messaging Configuration Options
    - v2_key Options
    - IPA Server Options
    - Certificate Options
    - Other Options

Installing on Red Hat OpenStack Platform

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

Obtaining the appliance

Uploading the Appliance on OpenStack

Log in to your OpenStack dashboard to upload your Infrastructure Automation appliance.

1. Log in to the OpenStack dashboard.
2. In the **Project** tab, browse to menu: **Compute > Images**.
3. Click **Create Image**.
4. In **Name**, enter a name for the image.
5. From **Image Source** list, select **Image Location**. Note that currently only images available via an HTTP URL are supported.
6. In **Image Location**, add an external (HTTP) URL to load the image from. For example, **http://example.com/image.iso**.
7. From the **Format** list, select the image format. For example, `ISO
    - Optical Disk Image`.
8. Specify the **Architecture**. For example, **`i386`** for a 32-bit architecture or **`x86-64`** for a 64-bit architecture.
9. Leave the **Minimum Disk (GB)** and **Minimum RAM (MB)** fields empty.
10. Check the **Public** box to make the appliance available to all users.
11. Check the **Protected** box to protect the image from being accidentally deleted.
12. Click **Create Image**.

You have successfully uploaded the Infrastructure Automation appliance.

The appliance image is placed in a queue to be uploaded. It may take some time before the Status of the image changes from Queued to Active.

Adding a Rule to a Security Group

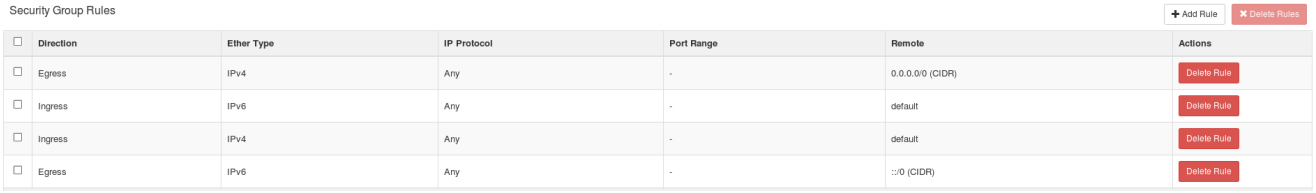
Security groups specify what IP traffic is allowed to reach an instance on its public IP address. Security group rules are processed before network traffic reaches firewall rules defined within the guest itself.

Note:

In the default configuration, the default security group accepts all connections from the default source; all instances within the default group can talk to each other on any port.

1. From the OpenStack dashboard, browse to menu: **Project > Compute > Access & Security**.

2. Browse to menu: **Security Groups > Manage Rules on the row for the default security group.
   Manage Security Group Rules: default



3. Click **Add Rule**.



4. Configure the rule.
   1. Select **Rule > Custom TCP Rule**.
   2. Select **Direction > Ingress**.
   3. Select **Port** from the **Open Port** list.
   4. Specify `443` in the **Port** field.
   5. Select **CIDR** from the **Remote** list.
   6. Specify `0.0.0.0/0` in the **CIDR** field.
   7. Click **Add**.

Creating a Custom Flavor

A flavor is a resource allocation profile that specifies, for example, how many virtual CPUs and how much RAM can be allocated to an instance. You can, for example, run Infrastructure Automation on a Red Hat OpenStack m1.large flavor, which specifies a virtual machine with 4 cores, 12 GB RAM, and 80 GB disk space. Creating a flavor to run Infrastructure Automation is optional.

The following procedure demonstrates creating a flavor with the minimum requirements (4 cores, 12 GB RAM, 44 GB disk space) for Infrastructure Automation. For more information about flavors, see the Red Hat OpenStack Platform Administration User Guide.

1. Log in to the OpenStack dashboard as admin.
2. In the **Admin** tab, browse to menu: **System > Flavors**.

3. Click **Create Flavor** to display the **Create Flavor** dialog.
4. Configure the settings to define a flavor that meets Infrastructure Automation system requirements.
    1. Enter a name for the flavor.
    2. Enter the following settings:
        - **VCPUs**: 4
        - **RAM MB**: 12288
        - **Root Disk GB**: 70
        - **Ephemeral Disk GB**: 0
        - **Swap Disk MB**: 0
5. Click **Create Flavor**.

A new flavor specific to Infrastructure Automation is created.

Launching the Instance

1. From the OpenStack dashboard, browse to menu: **Project > Compute > Instances**.
2. Click **Launch Instance**.
3. Enter a name for the instance.
4. Select the custom flavor for your instance. The flavor selection determines the computing resources available to your instance. The resources used by the flavor are displayed in the **Flavor Details** pane.
5. Enter **1** in the **Instance Count** field.
6. Select a boot option from the **Instance Boot Source** list:
    - **Boot from image** - displays a new field for **Image Name**. Select the image from the drop-down list.
    - **Boot from snapshot** - displays a new field for **Instance Snapshot**. Select the snapshot from the drop-down list.
    - **Boot from volume** - displays a new field for **Volume**. Select the volume from the drop-down list.
    - **Boot from image (creates a new volume) -** boot from an image and create a volume by choosing **Device Size** and **Device Name** for your volume. Some volumes can be persistent. To ensure the volume is deleted when the instance is deleted, select **Delete on Terminate**.
    - **Boot from volume snapshot (creates a new volume) -** boot from volume snapshot and create a new volume by choosing **Volume Snapshot** from the drop-down list and adding a **Device Name** for your volume. Some volumes can be persistent. To ensure the volume is deleted when the instance is deleted, select **Delete on Terminate**.
7. Click **Networking** and select a network for the instance by clicking the **+** (plus) button for the network from **Available Networks**.
8. Click **Launch**.

Adding a Floating IP Address

When you create an instance, Red Hat OpenStack Platform automatically assigns it a fixed IP address in the network to which the instance belongs. This IP address is permanently associated with the instance until the instance is terminated.

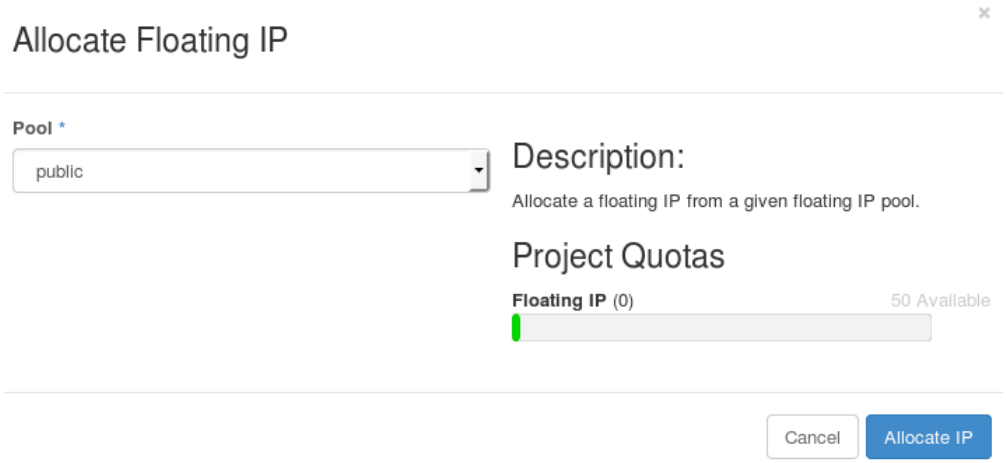
In addition to the fixed address, you can also assign a floating IP address to an instance. Unlike fixed IP addresses, you can modify floating IP addresses associations at any time, regardless of the state of the instances involved.

1. At the command-line on your RHEL OpenStack Platform host, create a pool of floating IP addresses using the **`nova-manage floating create`** command. Replace **IP_BLOCK** with the desired block of IP addresses expressed in CIDR notation.

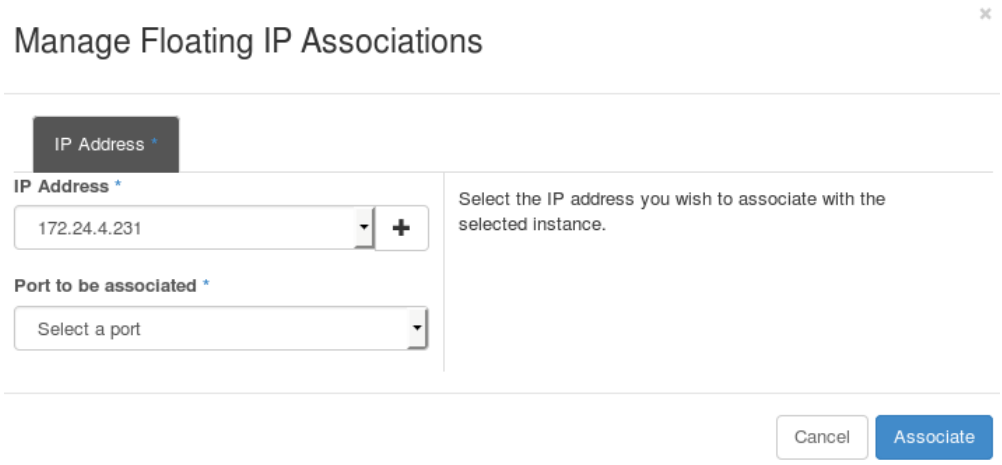
    ```
    $ nova-manage floating create IP_BLOCK
    ```

2. In the **Project** tab, browse to **Compute > Access & Security**.

3. Click **Floating IPs > Allocate IP To Project**. The **Allocate Floating IP** window is displayed.



4. Click **Allocate IP** to allocate a floating IP from the pool. The allocated IP address appears in the **Floating IPs** table.
5. Select the newly allocated IP address from the **Floating IPs** table. Click **Associate** to assign the IP address to a specific instance.



6. Select an instance with which to associate the floating IP Address.
7. Click **Associate** to associate the IP address with the selected instance.

Note:

To disassociate a floating IP address from an instance when it is no longer required, click **Release Floating IPs**.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **`Enter`** to manually configure settings.
4. Press the number for the item that you want to change, and press **`Enter`**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **`Enter`** to accept a setting where applicable.

Note: The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
    - If this is the first Infrastructure Automation appliance, choose **Create key**.
    - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

      Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

    - Enter **1** to choose **`/dev/vdb`** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **`/var/lib/pgsql`**, which can be found in the environment variable **`$APPLIANCE_PG_MOUNT_POINT`**.
    - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).
8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
    - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
    - Select **N** to configure the appliance with the full administrative user interface.
9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you need to configure the appliance to use an external PostgreSQL database. For example, you can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **`postgresql.conf`** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **`postgresql.conf`** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **`postgresql.conf`** to suit your system. For example, customize the **`shared_buffers`** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances that are expected to connect to the PostgreSQL instance, it might be necessary to alter the **`max_connections`** setting.

Note:

```
- Infrastructure Automation requires PostgreSQL version 13.
```

```
- `postgresql.conf` controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.
```

Use the following steps to configure an external PostgreSQL database:

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The appliance console summary screen is displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**`vmdb_production`**).
8. Enter the database username or leave blank for the default (**`root`**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation configures an external database.

Configure Messaging

Configuring messaging is required for the appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one Kafka broker per region.

1. You can either configure the current appliance as a Kafka broker, or point the appliance to an existing external Kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external Kafka broker. Fill in the required **Message Client Parameters**, such as IP address, username and password.
2. Select **Proceed**. **`appliance_console`** applies the configuration that you requested.
3. Restart **`evmserverd`** to pick up the changes.

Note: Use your Fully Qualified Domain Name (FQDN) as the messaging hostname rather than **`localhost`**, ensure that a resolvable and reachable non-localhost name entry is present in **`/etc/hosts`**.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**`5432`**).
9. Enter the database name or leave blank for the default (**`vmdb_production`**).
10. Enter the database username or leave blank for the default (**`root`**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region
14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**
2. Enter the default credentials as listed for the initial login.
   - **Username**: **admin**
   - **Password**: **smartvm**
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**.
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **`appliance_console_cli`** feature is a subset of the full functionality of the **`appliance_console`** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **`root`** and the default password of **`smartvm`**. This displays the Bash prompt for the root user.
2. Enter the **`appliance_console_cli`** or **`appliance_console_cli --help`** command to see a list of options available with the command, or simply enter **`appliance_console_cli --option <argument>`** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **`5432`**)
–username (-U)	database username (defaults to **`root`**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **`vmdb_production`**)

Messaging Configuration Options

| | | | ———————— | ———————————— | | Option | Description | | –message-server-config | configure appliance as a messaging server (boolean) | | –message-server-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-client-config | configure appliance as a messaging server (boolean) | | –message-client-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-keystore-username | messaging keystore username (defaults to admin) | | –message-keystore-password | messaging keystore password | | –message-server-username | messaging server username (defaults to admin) | | –message-server-password | messaging server password | | –message-server-port | messaging server port (defaults to 9093) | | –message-server-use-ipaddr | use ipaddress not hostname for messaging server | | –message-server-host | messaging server hostname or ipaddress | | –message-truststore-path-src | path for messaging server truststore | | –message-ca-cert-path-src | path to a certificate authority for messaging SSL | | –message-persistent-disk | configure a new volume for storing messaging data |

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **`root`**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

Set up an all-in-one appliance as a local database and messaging appliance and start the MIQ Server:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm --message-server-config --message-keystore-password smartvm --server=start
```

Join a worker appliance to an existing region:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm --server start
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword
```

To configure an appliance as a messaging server:

```
# appliance_console_cli --message-server-config --message-keystore-password="smartvm"
```

To configure an appliance as a messaging client:

```
# appliance_console_cli --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
```

```
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on Red Hat Virtualization
  - Installing Infrastructure Automation
    - Obtaining the Appliance
    - Uploading the Appliance to Red Hat Virtualization
    - Creating a Infrastructure Automation Virtual Machine
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database
      - Configuring an Internal Database
      - Configuring an External Database
    - Configure Messaging
    - Configuring a Worker Appliance
  - Logging In After Installing Infrastructure Automation
    - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
  - Database Configuration Options
  - Messaging Configuration Options
  - v2_key Options
  - IPA Server Options
  - Certificate Options
  - Other Options

Installing on Red Hat Virtualization

Installing Infrastructure Automation

Installing Infrastructure Automation on Red Hat Virtualization consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Uploading the appliance image to the Red Hat Virtualization data storage domain.
3. Creating a virtual machine based on the appliance.

After you have completed all the procedures in this chapter, you will have a working environment which you can further configure and customize.

Requirements.

Uploading the Infrastructure Automation appliance file to Red Hat Virtualization requires:

- 44 GB of storage space on both the export domain and the local partition where **/tmp** resides, as the **OVF** archive is locally expanded into that directory.
- 12 GB RAM.
- 4 vCPUs.

Note:

See Creating a Virtual Machine in the Red Hat Virtualization *Virtual Machine Management Guide* for information on specifying memory for a new virtual machine.

Obtaining the Appliance

After the image completes downloading, you are ready to upload the Infrastructure Automation appliance to your Red Hat Virtualization environment and create a virtual machine.

Uploading the Appliance to Red Hat Virtualization

Upload the **qcow2** appliance image to your Red Hat Virtualization data storage domain from the Red Hat Virtualization Administration Portal.

- You must configure the Image I/O Proxy when running **engine-setup**. See Configuring the Red Hat Virtualization Manager in the *Red Hat Virtualization Installation Guide* for more information.
- Internet Explorer 10, Firefox 35, or Chrome 13 or greater is required to perform this upload procedure. Previous browser versions do not support the required HTML5 APIs.
- You must import the required certificate authority into the web browser used to access the Administration Portal.

Note:

To import the certificate authority, browse to **`https://<engine_address>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA`** and select all the trust settings. Refer to the instructions to install the certificate authority in Firefox, Internet Explorer, or Google Chrome.

To upload the appliance:

1. Log in to the Red Hat Virtualization Administration Portal.
2. Click **Storage > Disks**.
3. Select **Upload > Start**.
4. Click **Choose File**, and select the appliance image to upload.
5. Under **Disk Options**, enter an **Alias** for the image.
6. (Optional) Edit other fields in **Disk Options** if desired.
7. Click **OK**.

A progress bar in the **Status** column indicates the status of the upload. You can also pause, cancel, or resume uploads from the **Upload** menu.

The status shows **OK** when the image has completed uploading.

Note:

See the Uploading Images to a Data Storage Domain in the *Red Hat Virtualization Administration Guide* for more information.

1. If the upload times out and you see the message, **`Reason: timeout due to transfer inactivity`**, increase the timeout value:

   ```
   # engine-config -s TransferImageClientInactivityTimeoutInSeconds=6000
   ```

2. Restart the ovirt-engine service:

   ```
   # systemctl restart ovirt-engine
   ```

Creating a Infrastructure Automation Virtual Machine

After uploading the appliance to the data domain, create a virtual machine containing the Infrastructure Automation image and a second disk for the database:

1. In **Compute > Virtual Machines**, click **New** to open the **New Virtual Machine** dialog.
2. From the **General** tab, specify a name for the virtual machine and any other details. Ensure that **`Q35 Chipset with BIOS`** is selected for **Chipset/Firmware Type**.
3. Click **Attach**.
4. Select the Infrastructure Automation appliance you uploaded from the list of images and click **OK**.
5. Click the VM to open its details screen, and click the **Disks** tab.
6. Click **Edit** to the disk, and check **Bootable** if it is not marked already.
7. Click **OK**.
8. Add a second disk for the database:
   1. Click the **General** tab and click the **Edit** button to edit the virtual machine.
   2. Click **+** and **Create** to create a disk for the database (VMDB).
   3. Specify a **Size** in GB for the disk that allows sufficient space for your database. Red Hat recommends allocating the virtual machine disk fully at the time of creation. Three main factors affect the size of your database over time:
      - Virtual Machine Count: the most important factor in the calculation of virtual machine database (VMDB) size over time.
      - Host Count: the number of hosts associated with the provider.
      - Storage Count: the number of individual storage elements as seen from the perspective of the provider or host. It is not the total number of virtual disks for all virtual machines.
   4. Set the **Allocation Policy** to **`Preallocated`** (thick provisioning) for best performance.
   5. Specify any other values as desired.
9. Click **OK** to create the disk and return to the **New Virtual Machine** window.
10. Add a network interface to the virtual machine by selecting a vNIC profile from the **nic1** list.
11. Click **OK** to save your changes to the virtual machine.
12. Start the Infrastructure Automation appliance by clicking the **Run** button.

Your Red Hat Virtualization environment now contains a running Infrastructure Automation appliance.

Next, configure the database on the disk you created by following the steps in Configuring an Internal Database.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **`Enter`** to manually configure settings.
4. Press the number for the item that you want to change, and press **`Enter`**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **`Enter`** to accept a setting where applicable.

Note: The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
   - If this is the first Infrastructure Automation appliance, choose **Create key**.
   - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

     Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

   - Enter **1** to choose **`/dev/vdb`** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **`/var/lib/pgsql`**, which can be found in the environment variable **`$APPLIANCE_PG_MOUNT_POINT`**.
   - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).
8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
   - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.

- Select **N** to configure the appliance with the full administrative user interface.

9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you need to configure the appliance to use an external PostgreSQL database. For example, you can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **`postgresql.conf`** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **`postgresql.conf`** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **`postgresql.conf`** to suit your system. For example, customize the **`shared_buffers`** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances that are expected to connect to the PostgreSQL instance, it might be necessary to alter the **`max_connections`** setting.

Note:

`- Infrastructure Automation requires PostgreSQL version 13.`

`- `postgresql.conf` controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.`

Use the following steps to configure an external PostgreSQL database:

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The appliance console summary screen is displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**`vmdb_production`**).
8. Enter the database username or leave blank for the default (**`root`**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation configures an external database.

Configure Messaging

Configuring messaging is required for the appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one Kafka broker per region.

1. You can either configure the current appliance as a Kafka broker, or point the appliance to an existing external Kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external Kafka broker. Fill in the required **Message Client Parameters**, such as IP address, username and password.
2. Select **Proceed**. **`appliance_console`** applies the configuration that you requested.
3. Restart **`evmserverd`** to pick up the changes.

Note: Use your Fully Qualified Domain Name (FQDN) as the messaging hostname rather than **`localhost`**, ensure that a resolvable and reachable non-localhost name entry is present in **`/etc/hosts`**.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.

4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (`5432`).
9. Enter the database name or leave blank for the default (`vmdb_production`).
10. Enter the database username or leave blank for the default (`root`).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region
14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

1. Browse to the URL for the login screen. For example, URL should be `https://xx.xx.xx.xx`
2. Enter the default credentials as listed for the initial login.
   - **Username**: **admin**
   - **Password**: **smartvm**
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. For example, URL should be `https://xx.xx.xx.xx`.
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the `appliance_console_cli` feature is a subset of the full functionality of the `appliance_console` itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of `root` and the default password of `smartvm`. This displays the Bash prompt for the root user.
2. Enter the `appliance_console_cli` or `appliance_console_cli --help` command to see a list of options available with the command, or simply enter `appliance_console_cli --option <argument>` directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)

–hostname (-h)	database hostname
–port	database port (defaults to **5432**)
–username (-U)	database username (defaults to **root**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **vmdb_production**)

Messaging Configuration Options

| | | | ———————— | ———————————————— | | Option | Description | | –message-server-config | configure appliance as a messaging server (boolean) | | –message-server-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-client-config | configure appliance as a messaging server (boolean) | | –message-client-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-keystore-username | messaging keystore username (defaults to admin) | | –message-keystore-password | messaging keystore password | | –message-server-username | messaging server username (defaults to admin) | | –message-server-password | messaging server password | | –message-server-port | messaging server port (defaults to 9093) | | –message-server-use-ipaddr | use ipaddress not hostname for messaging server | | –message-server-host | messaging server hostname or ipaddress | | –message-truststore-path-src | path for messaging server truststore | | –message-ca-cert-path-src | path to a certificate authority for messaging SSL | | –message-persistent-disk | configure a new volume for storing messaging data |

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **root**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

Set up an all-in-one appliance as a local database and messaging appliance and start the MIQ Server:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm --message-server-config --message-keystore-password smartvm --server=start
```

Join a worker appliance to an existing region:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm --server start
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword
```

To configure an appliance as a messaging server:

```
# appliance_console_cli --message-server-config --message-keystore-password="smartvm"
```

To configure an appliance as a messaging client:

```
# appliance_console_cli --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on Google Compute Engine
  - Installing Infrastructure Automation
    - Obtaining the Appliance
    - Uploading the Appliance on Google Compute Engine
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database
      - Configuring an Internal Database
      - Configuring an External Database
    - Configure Messaging
    - Configuring a Worker Appliance
  - Logging In After Installing Infrastructure Automation
    - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
  - Database Configuration Options
  - Messaging Configuration Options
  - v2_key Options
  - IPA Server Options
  - Certificate Options
  - Other Options

Installing on Google Compute Engine

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you complete all the listed preceding steps, you have a working environment on which additional customizations and configurations can be performed.

Obtaining the Appliance

Uploading the Appliance on Google Compute Engine

When the Infrastructure Automation Google Compute Engine appliance is downloaded to your system, upload the appliance to Google Compute Engine.

To upload the Infrastructure Automation Google Compute Engine appliance file, you need to satisfy the following requirements:

- 44 GB of storage space
- 12 GB RAM
- 4 vCPUs.

You can upload the appliance with the following steps:

1. Log in to the Google Cloud Platform dashboard.
2. Click **Home** located on the left of the screen.
3. Click  to show the **Products and Services** menu. Click **Storage**.
4. Create a bucket by clicking **Create Bucket**, and configure the following details:
    1. Enter a unique **Name** for the bucket. Valid characters include lowercase alphanumeric characters, hyphens, and underscores.
    2. Configure your location from the dropdown list.
    3. Click **Create**.
5. Click **Upload Files** and browse to the location of the Infrastructure Automation Google Compute Engine appliance you downloaded. Select the **`tar.gz`** file on your local machine, and click **Open** to begin the upload.
6. When the upload is complete, click  **Products & services > Compute Engine** on the left menu.
7. Create an image by clicking **Images** from the left menu, then **Create Image**. Enter the following details about the image:
    1. Enter a unique **Name** for the image. Valid characters include lowercase alphanumeric characters and hyphens.
    2. Optionally, add a **Description**.
    3. Configure **Encryption** if needed. The default value is set to **`Automatic (recommended)`**.
    4. In **Source**, use the dropdown to select **`Cloud Storage file`**. This shows the **Cloud Storage file** field.
    5. In **Cloud Storage file**, click **Browse** to open the **Select object** window. Select the bucket containing the image that you uploaded and click the **>** symbol to locate the **`tar.gz`** image inside the bucket. Select the image and click **Select**.
    6. Click **Create**. Creating an image can take a few minutes. When the image is created, the screen refreshes, and the new image appears in the **Images** list.
8. Create a virtual machine instance by navigating to **VM instances > Create Instance**, and configure the following fields:
    1. Enter a unique **Name** for the virtual machine instance. Valid characters include lowercase alphanumeric characters and hyphens.
    2. Select the **Zone** closest to your location.
    3. Under **Boot disk**, click **Change** to open the **Boot disk** window. Click the **Your Image** tab and select the disk that you previously created. Click **Select**.
    4. Under **Firewall**, select the checkbox for **Allow HTTPS traffic**.
    5. Click **Management, disk, networking, SSH keys**. Then, click the **SSH Keys** tab, add your entire SSH public key data in the **Username** box.
    6. Configure any other fields as needed, and click **Create**.

Your new virtual machine instance and the external IP address for accessing the Infrastructure Automation interface appears under **VM instances**.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.

3. Press **Enter** to manually configure settings.
4. Press the number for the item that you want to change, and press **Enter**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **Enter** to accept a setting where applicable.

Note: The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
   - If this is the first Infrastructure Automation appliance, choose **Create key**.
   - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

     Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.** message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

   - Enter **1** to choose **/dev/vdb** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **/var/lib/pgsql**, which can be found in the environment variable **$APPLIANCE_PG_MOUNT_POINT**.
   - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).
8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
   - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
   - Select **N** to configure the appliance with the full administrative user interface.
9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you need to configure the appliance to use an external PostgreSQL database. For example, you can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **postgresql.conf** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **postgresql.conf** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **postgresql.conf** to suit your system. For example, customize the **shared_buffers** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances that are expected to connect to the PostgreSQL instance, it might be necessary to alter the **max_connections** setting.

Note:

- Infrastructure Automation requires PostgreSQL version 13.

- `postgresql.conf` controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.

Use the following steps to configure an external PostgreSQL database:

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The appliance console summary screen is displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**vmdb_production**).
8. Enter the database username or leave blank for the default (**root**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation configures an external database.

Configure Messaging

Configuring messaging is required for the appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one Kafka broker per region.

1. You can either configure the current appliance as a Kafka broker, or point the appliance to an existing external Kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external Kafka broker. Fill in the required **Message Client Parameters**, such as IP address, username and password.

2. Select **Proceed**. **appliance_console** applies the configuration that you requested.
3. Restart **evmserverd** to pick up the changes.

Note: Use your Fully Qualified Domain Name (FQDN) as the messaging hostname rather than **localhost**, ensure that a resolvable and reachable non-localhost name entry is present in **/etc/hosts**.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note: All Infrastructure Automation appliances in a multi-region deployment must use the same key.

6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**5432**).
9. Enter the database name or leave blank for the default (**vmdb_production**).
10. Enter the database username or leave blank for the default (**root**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.

13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region

14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password

15. Confirm the configuration if prompted.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**
2. Enter the default credentials as listed for the initial login.
   - **Username**: **admin**
   - **Password**: **smartvm**
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. For example, URL should be **`https://xx.xx.xx.xx`**.
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **`appliance_console_cli`** feature is a subset of the full functionality of the **`appliance_console`** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **`root`** and the default password of **`smartvm`**. This displays the Bash prompt for the root user.
2. Enter the **`appliance_console_cli`** or **`appliance_console_cli --help`** command to see a list of options available with the command, or simply enter **`appliance_console_cli --option <argument>`** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **`5432`**)
–username (-U)	database username (defaults to **`root`**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **`vmdb_production`**)

Messaging Configuration Options

| | | –––––––––– | ––––––––––––––––––– | | Option | Description | | –message-server-config | configure appliance as a messaging server (boolean) | | –message-server-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-client-config | configure appliance as a messaging server (boolean) | | –message-client-unconfig | unconfigure appliance as a messaging server (boolean) | | –message-keystore-username | messaging keystore username (defaults to admin) | | –message-keystore-password | messaging keystore password | | –message-server-username | messaging server username (defaults to admin) | | –message-server-password | messaging server password | | –message-server-port | messaging server port (defaults to 9093) | | –message-server-use-ipaddr | use ipaddress not hostname for messaging server | | –message-server-host | messaging server hostname or ipaddress | | –message-truststore-path-src | path for messaging server truststore | | –message-ca-cert-path-src | path to a certificate authority for messaging SSL | | –message-persistent-disk | configure a new volume for storing messaging data |

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **root**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

Set up an all-in-one appliance as a local database and messaging appliance and start the MIQ Server:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm --message-server-config --message-keystore-password smartvm --server=start
```

Join a worker appliance to an existing region:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword --message-client-config --message-server-host db.example.com --message-server-password smartvm --message-keystore-password smartvm --server start
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword
```

To configure an appliance as a messaging server:

```
# appliance_console_cli --message-server-config --message-keystore-password="smartvm"
```

To configure an appliance as a messaging client:

```
# appliance_console_cli --message-client-config --message-server-host db.example.com --message-server-password smartvm --
message-keystore-password smartvm
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```

OIDC Configuration steps for Infrastructure Management appliance only

Complete the following steps to configure Infrastructure Management virtual machine appliance with OIDC (OpenID Connect).

- Prerequisites
- Step 1. Register Infrastructure Management instance with IAM as an OIDC client
- Step 2. Import the Root CA certificate to Infrastructure Management
- Step 3. Configure the Administrative UI
- Step 4. Integrating Infrastructure Management with IBM Cloud Pak for AIOps

Prerequisites

- Before you configure OIDC make sure you install the Infrastructure Management virtual machine appliance. For more information, see Installation and Upgrade.
- Single sign-on with Infrastructure Management and IBM Cloud Pak for AIOps requires an LDAP server connection.

  For more information about adding an LDAP connection, see Configuring LDAP connection.

Step 1. Register Infrastructure Management instance with IAM as an OIDC client

In order to enable single sign-on (SSO) for Infrastructure Management with OIDC, the Infrastructure Management instance needs to register as an OIDC client with Identity and Access Management (IAM). Complete these steps on the IBM Cloud Pak for AIOps cluster.

You can register Infrastructure Management as an OIDC client with IAM using the **cloudctl** command.

The registration method requires the following registration payload in a file "registration.json":

```
{
  "token_endpoint_auth_method":"client_secret_basic",
  "client_id": "<YOUR_CLIENT_ID>",
  "client_secret": "<YOUR_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://<YOUR_CLOUD_PAK_ROUTE>"   ],
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":[
     "https://<YOUR_CLOUD_PAK_ROUTE>/"    ],
  "redirect_uris":
["https://<YOUR_CLOUD_PAK_ROUTE>/auth/liberty/callback","https://<INFRA_MGMT_URL>/oidc_login/redirect_uri"]
}
```

Example registration payload (for reference only):

```
{
  "token_endpoint_auth_method":"client_secret_basic",
```

```
  "client_id": "AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK",
  "client_secret": "AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK",
  "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://cp-console-cp4aiops.apps.mycluster.mydomain.com "],
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":["https://cp-console-cp4aiops.apps.mycluster.mydomain.com /"],
  "redirect_uris":["https://cp-console-cp4aiops.apps.mycluster.mydomain.com /auth/liberty/callback","https://im-
mycluster.mydomain.com/oidc_login/redirect_uri"]
}
```

1. Create a file named **registration.json** from the example template. Replace the values in the example template payload registration with the actual values based on your installation.

   - **YOUR_CLIENT_ID** Your base64 encoded character string.
   - **YOUR_CLIENT_SECRET** Your base64 encoded character string.

     Note: The **<YOUR_CLIENT_ID>** and **<YOUR_CLIENT_SECRET>** need to be generated. The values can be any string, but normally a 32 character string that is base64 encoded is used. You can use BASE64 to encode your character string. For more information, see: BASE64. Make a note of the values you generate for **<YOUR_CLIENT_ID>** and **<YOUR_CLIENT_SECRET>**. You will use the values in the next section to update the Apache configuration file.

     Example command that uses base64 to encode a character string:

     ```
     #
     # Generate two encrypted streams from some longer-than-32-characters strings
     #
     echo There is a huge white elephant in LA zoo |base64
     echo 12345678901234567890123456789012345 |base64
     ```

   - **YOUR_CLOUD_PAK_ROUTE** The URL of the IBM Cloud Pak for AIOps console.
   - **INFRA_MGMT_URL** The URL of the Infrastructure Management virtual machine appliance.
   - **post_logout_redirect_uris** The URL of the IBM Cloud Pak for AIOps console.
   - **trusted_uri_prefixes** The URL of the IBM Cloud Pak for AIOps console with "forward slash" /.
   - **redirect_uris** The URL of the IBM Cloud Pak for AIOps console with the path to call back and the URL of the Infrastructure Management host with the path to the redirect_uri.

   Note: You can run the following command on the IBM Cloud Pak for AIOps cluster to determine the URL of the IBM Cloud Pak for AIOps console:

   ```
   oc get routes cp-console -o=jsonpath='{.spec.host}' -n <IA_installation_namespace>
   ```

   Where **<IA_installation_namespace>** is the namespace that Infrastructure Automation is installed in.

2. After the file **registration.json** is completed, log in and run the command to register Infrastructure Management as an OIDC client.

   Note: Include the **-n kube-system** to specify this project, or the **cloudctl iam** command can fail.

   ```
   cloudctl login -a https://<YOUR_CLOUD_PAK_ROUTE> -n kube-system
   ```

   Example **cloudctl iam** command:

   ```
   cloudctl iam oauth-client-register -f registration.json
   ```

Step 2. Import the Root CA certificate to the Infrastructure Management appliance

1. Retrieve the cluster CA cert by running this command on the cluster:

   ```
   oc get secret -n <IA_installation_namespace> ibmcloud-cluster-ca-cert -o jsonpath='{.data.ca\.crt}' | base64 --decode
   ```

   Where **<IA_installation_namespace>** is the namespace that Infrastructure Automation is installed in.

   Note: When copying this command there can be some additional characters added that can cause incorrect command syntax. You can copy the first portion **oc get secret -n <IA_installation_namespace> ibmcloud-cluster-ca-cert -o jsonpath=** and manually add the remaining syntax **'{.data.ca\.crt}'| base64 --decode**.

2. Copy and paste the output to a file, for example **ibm_cp_im.crt**

3. Edit the **ibm_cp_im.crt** file, and change:

   - **BEGIN CERTIFICATE** to **BEGIN TRUSTED CERTIFICATE**
   - **END CERTIFICATE** to **END TRUSTED CERTIFICATE**

   Note: The following steps must be completed by logging in to the Infrastructure Management appliance system as root user:

4. Copy the updated **ibm_cp_im.crt** file to the Infrastructure Management appliance and save it in the directory: **/etc/pki/ca-trust/source/anchors**

5. Run the command:

```
update-ca-trust
```

6. Restart the evm server by running the command:

```
systemctl restart evmserverd
```

7. Copy the Apache OIDC template configuration file with these steps:

```
export TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"

cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user-openidc.conf /etc/httpd/conf.d/

cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth-openidc.conf.erb  /etc/httpd/conf.d/manageiq-external-
auth-openidc.conf
```

8. The Apache **/etc/httpd/conf.d/manageiq-external-auth-openidc.conf** configuration file must be updated with installation-specific values. Replace the contents of the file with the actual values based on the installation.

   Sample configuration file:

```
LoadModule          auth_openidc_module modules/mod_auth_openidc.so
ServerName          https://<YOUR_IM_APPLIANCE_HOSTNAME>
LogLevel            warn

OIDCClientID                    <YOUR_CLIENT_ID>
OIDCClientSecret                <YOUR_CLIENT_SECRET>
OIDCRedirectURI                 https://<YOUR_IM_APPLIANCE_HOSTNAME>/oidc_login/redirect_uri
OIDCCryptoPassphrase            <PASSPHRASE>
OIDCOAuthRemoteUserClaim        sub
OIDCRemoteUserClaim             name

OIDCOAuthClientID                  <YOUR_CLIENT_ID>
OIDCOAuthClientSecret              <YOUR_CLIENT_SECRET>
OIDCOAuthIntrospectionEndpoint     https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/introspect
OIDCOAuthIntrospectionEndpointAuth client_secret_basic

OIDCProviderIssuer                  https://127.0.0.1:443/idauth/oidc/endpoint/OP
OIDCProviderAuthorizationEndpoint   https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/authorize
OIDCProviderTokenEndpoint           https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/token
OIDCProviderJwksUri                 https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/jwk
OIDCProviderEndSessionEndpoint      https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/logout

OIDCScope                         "openid email profile"
OIDCResponseMode                  "query"
OIDCProviderTokenEndpointAuth     client_secret_post

OIDCPassUserInfoAs json
OIDCSSLValidateServer off
OIDCOAuthSSLValidateServer off
OIDCHTTPTimeoutShort 10

OIDCCacheEncrypt On
<Location /oidc_login>
  AuthType  openid-connect
  Require   valid-user
</Location>

<Location /ui/service/oidc_login>
  AuthType openid-connect
  Require valid-user
  Header set Set-Cookie "miq_oidc_access_token=%{OIDC_access_token}e; Max-Age=10; Path=/ui/service"
</Location>

<LocationMatch ^/api(?!\/(v[\d\.]+\/)?product_info$)>
  SetEnvIf X-Auth-Token  '^.+$'                let_api_token_in
  SetEnvIf X-MIQ-Token   '^.+$'                let_sys_token_in
  SetEnvIf X-CSRF-Token  '^.+$'                let_csrf_token_in

  AuthType  oauth20
  AuthName  "External Authentication (oidc) for API"

  Require   valid-user
  Order     Allow,Deny
  Allow from env=let_api_token_in
  Allow from env=let_sys_token_in
  Allow from env=let_csrf_token_in
  Satisfy   Any
</LocationMatch>

RequestHeader unset X-REMOTE-USER
```

```
RequestHeader unset X-REMOTE_USER
RequestHeader unset X_REMOTE-USER
RequestHeader unset X_REMOTE_USER
RequestHeader set X_REMOTE_USER %{OIDC_CLAIM_PREFERRED_USERNAME}e env=OIDC_CLAIM_PREFERRED_USERNAME
RequestHeader set X_EXTERNAL_AUTH_ERROR %{EXTERNAL_AUTH_ERROR}e env=EXTERNAL_AUTH_ERROR
RequestHeader set X_REMOTE_USER_EMAIL %{OIDC_CLAIM_EMAIL}e env=OIDC_CLAIM_EMAIL
RequestHeader set X_REMOTE_USER_FIRSTNAME %{OIDC_CLAIM_GIVEN_NAME}e env=OIDC_CLAIM_GIVEN_NAME
RequestHeader set X_REMOTE_USER_LASTNAME %{OIDC_CLAIM_FAMILY_NAME}e env=OIDC_CLAIM_FAMILY_NAME
RequestHeader set X_REMOTE_USER_FULLNAME %{OIDC_CLAIM_NAME}e env=OIDC_CLAIM_NAME
RequestHeader set X_REMOTE_USER_GROUPS %{OIDC_CLAIM_GROUPS}e env=OIDC_CLAIM_GROUPS
RequestHeader set X_REMOTE_USER_DOMAIN %{OIDC_CLAIM_DOMAIN}e env=OIDC_CLAIM_DOMAIN
```

- **YOUR_IM_APPLIANCE_HOSTNAME** Specifies the hostname of the Infrastructure management appliance server.
- **YOUR_CLIENT_ID** The client ID used for registering Infrastructure management as an OIDC client with IAM.
- **YOUR_CLIENT_SECRET** The client secret that is used for registering Infrastructure management as an OIDC client with IAM.
- **YOUR_CLOUD_PAK_ROUTE** The URL of the IBM Cloud Pak UI console.
- **OIDCCryptoPassphrase** Can be any arbitrary alpha-numeric string.

Note: **YOUR_CLIENT_ID** and **YOUR_CLIENT_SECRET** values are generated when you register Infrastructure Management as an OIDC client.

Example configuration file (for reference only):

```
LoadModule          auth_openidc_module modules/mod_auth_openidc.so
ServerName          https://im-mycluster.mydomain.com
LogLevel            warn

OIDCCLientID                        AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK
OIDCClientSecret                    AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK
OIDCRedirectURI                     https://im-mycluster.mydomain.com/oidc_login/redirect_uri
OIDCCryptoPassphrase                alphabeta
OIDCOAuthRemoteUserClaim            sub
OIDCRemoteUserClaim                 name

OIDCOAuthClientID                   AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK
OIDCOAuthClientSecret
AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK
OIDCOAuthIntrospectionEndpoint      https://cp-console-cp4aiops.apps.mycluster.mydomain.com
/idprovider/v1/auth/introspect
OIDCOAuthIntrospectionEndpointAuth client_secret_basic

OIDCProviderIssuer                  https://127.0.0.1:443/idauth/oidc/endpoint/OP
OIDCProviderAuthorizationEndpoint   https://cp-console-cp4aiops.apps.mycluster.mydomain.com
/idprovider/v1/auth/authorize
OIDCProviderTokenEndpoint           https://cp-console-cp4aiops.apps.mycluster.mydomain.com /idprovider/v1/auth/token
OIDCProviderJwksUri                 https://cp-console-cp4aiops.apps.mycluster.mydomain.com /idprovider/v1/auth/jwk
OIDCProviderEndSessionEndpoint      https://cp-console-cp4aiops.apps.mycluster.mydomain.com /idprovider/v1/auth/logout

OIDCScope                           "openid email profile"
OIDCResponseMode                    "query"
OIDCProviderTokenEndpointAuth       client_secret_post

OIDCPassUserInfoAs json
OIDCSSLValidateServer off
OIDCOAuthSSLValidateServer off
OIDCHTTPTimeoutShort 10

OIDCCacheEncrypt On
<Location /oidc_login>
  AuthType  openid-connect
  Require   valid-user
</Location>

<Location /ui/service/oidc_login>
  AuthType openid-connect
  Require valid-user
  Header set Set-Cookie "miq_oidc_access_token=%{OIDC_access_token}e; Max-Age=10; Path=/ui/service"
</Location>

<LocationMatch ^/api(?!\/(v[\d\.]+\/)?product_info$)>
  SetEnvIf X-Auth-Token  '^.+$'                 let_api_token_in
  SetEnvIf X-MIQ-Token   '^.+$'                 let_sys_token_in
  SetEnvIf X-CSRF-Token  '^.+$'                 let_csrf_token_in

  AuthType  oauth20
  AuthName  "External Authentication (oidc) for API"

  Require   valid-user
  Order     Allow,Deny
  Allow from env=let_api_token_in
  Allow from env=let_sys_token_in
  Allow from env=let_csrf_token_in
  Satisfy   Any
</LocationMatch>

RequestHeader unset X-REMOTE-USER
RequestHeader unset X-REMOTE_USER
RequestHeader unset X_REMOTE-USER
RequestHeader unset X_REMOTE_USER
RequestHeader set X_REMOTE_USER %{OIDC_CLAIM_PREFERRED_USERNAME}e env=OIDC_CLAIM_PREFERRED_USERNAME
RequestHeader set X_EXTERNAL_AUTH_ERROR %{EXTERNAL_AUTH_ERROR}e env=EXTERNAL_AUTH_ERROR
RequestHeader set X_REMOTE_USER_EMAIL %{OIDC_CLAIM_EMAIL}e env=OIDC_CLAIM_EMAIL
```

```
RequestHeader set X_REMOTE_USER_FIRSTNAME %{OIDC_CLAIM_GIVEN_NAME}e env=OIDC_CLAIM_GIVEN_NAME
RequestHeader set X_REMOTE_USER_LASTNAME %{OIDC_CLAIM_FAMILY_NAME}e env=OIDC_CLAIM_FAMILY_NAME
RequestHeader set X_REMOTE_USER_FULLNAME %{OIDC_CLAIM_NAME}e env=OIDC_CLAIM_NAME
RequestHeader set X_REMOTE_USER_GROUPS %{OIDC_CLAIM_GROUPS}e env=OIDC_CLAIM_GROUPS
RequestHeader set X_REMOTE_USER_DOMAIN %{OIDC_CLAIM_DOMAIN}e env=OIDC_CLAIM_DOMAIN
```

9. Restart Apache on the appliance.

```
systemctl restart httpd
```

Step 3. Configure the Administrative UI

Update the Appliance Administrative UI to be OIDC aware and function. Complete these steps on each UI-enabled Infrastructure Management appliance.

1. Log in as **admin**.
2. Select the **Settings** > **Application Settings**, select **"Server "EVM [1]" (current)"** under Zones, then select the **Authentication** tab.
3. In the **Authentication** section, set the **Mode** to **External (httpd)**
4. In the **External Authentication (HTTPd) Settings** section, set **Provider Type** to **Enable OpenID-Connect**.
   - **Note:** This setting enables the OIDC login button on the login screen that redirects to the OIDC protected page for authentication, and supports the OIDC logout process.
5. Optional: In the **External Authentication (HTTPd) Settings** section, select **Enable Single Sign-On**.
   - **Note:** If you select this option, the initial access to the Appliance Administrative UI will redirect to the OIDC Identity Provider authentication screen.
6. In the **Role Settings** section, select the **Get User Groups from External Authentication (HTTPd)** setting.
7. Click Save.
8. Select **Access Control**, click **Groups** and make sure the user's groups are created on the Appliance and appropriate roles are assigned to those groups. The user's groups to be added in Infrastructure Management should have the same names as the groups defined in the LDAP server that is configured in the IBM Cloud Pak console.
   1. Under the **Access Control**, click **Groups**.
   2. Click **Configuration**, then **Add a new Group**.
   3. Enter your existing LDAP group name in the **Description** field. For example, **im_ldap_group**.
   4. Select the **Role** for this LDAP group. For example, select the **EvmRole-super_administrator** to map to this group. The Infrastructure Management administrator can map to the roles that make sense for the user group.
   5. Select **My Company** for **Project/Tenant**.
   6. Click Add.

   Note: Access control in Infrastructure Management is based on group membership as roles are assigned to groups. When Infrastructure Management is integrated with IBM Cloud Pak for AIOps using single sign-on (SSO), it looks at the user's group membership in the identity token and checks if that group exists in Infrastructure Management. If the group doesn't exist, then access is denied. You must create at least one group with the role **EvmRole-super_administrator** and assign one user to this group. This user will be the Admin user for Infrastructure Management and referred to as **IM_ADMIN_LDAP_USER**. The **IM_ADMIN_LDAP_USER** must be a member of the LDAP group with the role **EvmRole-super_administrator** in Infrastructure Management.

   Important: You must create the groups in Infrastructure Management that match your existing LDAP groups by name, and assign the groups account roles. At least one group to which the user belongs in LDAP that IBM Cloud Pak for AIOps is configured to use must also be created in Infrastructure Management. You must assign a proper role to this group in Infrastructure Management.

   Example: In LDAP a group that is named **imgroup100** exists and a user with username **imuser100** is a member of the group. The user **imuser100** and the group **imgroup100** must be created in Infrastructure Management.

Step 4. Integrating Infrastructure Management with IBM Cloud Pak for AIOps

Enable navigation to Infrastructure Management within the IBM Cloud Pak console.

Complete the following steps on a Linux system. You can use the boot node from the HUB cluster where IBM Cloud Pak for AIOps is installed. These steps enable navigation to Infrastructure Management from the IBM Cloud Pak console:

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository to obtain the menu customization script. You must run the script on a Linux operating system.

   Download the supporting subdirectories to make the script run successfully. The files and subdirectories that are required to run the script are

   ```
   im-appliance-link-install.sh
   nav/automate.yml
   nav_template/automate.sh
   ```

2. Install and authenticate **kubectl**. For more information, see Installing the Kubernetes CLI (kubectl).
3. Navigate to the directoy **cp4waiops-samples/infrastructure-automation/infrastructure-management/appliance**. Set the file permissions on the script and run **im-appliance-link-install.sh** to enable navigation to your Infrastructure management instance:

   ```
   chmod 755 ./im-appliance-link-install.sh
   ```

   ```
   ./im-appliance-link-install.sh -i <Infrastructure management URL>
   ```

- `-i` Is a required parameter that refers to the URL for the Infrastructure Management console. For example, `./im-appliance-link-install.sh -i https://vm17-im-mycluster.mydomain.com/`

4. Verify that the Infrastructure Management instance is in the IBM Cloud Pak console navigation menu. From the IBM Cloud Pak navigation menu, click **Automate Infrastructure** > **Infrastructure management**.

Infrastructure Management is integrated with the IBM Cloud Pak console.

Deploying Infrastructure Management as a containerized deployment (podified)

Complete these steps to install Infrastructure Management as a containerized deployment.

Note: Before you proceed further, make sure that Cloud Pak for AIOps is installed.

- Prerequisites
- Deploy the ibm-management-im-install operand

Prerequisites

- Ensure the following operators for **`Infrastructure management`** have been installed.
  - **`ibm-management-im-install`** for Infrastructure Management has been installed. For more information, see Infrastructure management.
- Ensure IBM Cloud Pak® is setup for LDAP authentication. You need an existing user-group from this LDAP repository to use for the Infrastructure Management deployment.

Deploy the ibm-management-im-install operand

Install an instance of the Infrastructure Management using the Red Hat® OpenShift® Container Platform console or the CLI (command-line tools).

1. Create an installation instance CR file **`<im-install-cr.yaml>`** using the following yaml file and update the **`applicationDomain`**, **`imagePullSecret`**, **`initialAdminGroupName`** and accept license. Add **`storageClassName: <storage-Class-Name>`** in the create an installation instance CR file **`<im-install-cr.yaml>`**.

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  labels:
    app.kubernetes.io/instance: ibm-infra-management-install-operator
    app.kubernetes.io/managed-by: ibm-infra-management-install-operator
    app.kubernetes.io/name: ibm-infra-management-install-operator
  name: im-iminstall
  namespace: cp4aiops
spec:
  applicationDomain: <YOUR_IM_HTTPD_ROUTE>
  imagePullPolicy: Always
  imagePullSecret: ibm-entitlement-key
  initialAdminGroupName: <YOUR_LDAP_USER_GROUP>
  storageClassName: <RWO-storage-class-name>
  license:
    accept: true
```

   Where:

   - **YOUR_IM_HTTPD_ROUTE** is a user-defined route, which must include a name for your installation plus part of your IBM Cloud Pak for AIOps console console's route.
   - **YOUR_LDAP_USER_GROUP** is an existing user-group defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure Management to match your existing LDAP group by name, and assigned an account role which facilitates SSO login.
   - **RWO-storage-class-name** is a storage class name.

     Important: You must specify an LDAP user-group and it must contain at least one user that is able to login to the IBM Cloud Pak for AIOps console. For example, you have an existing LDAP group that is named group100 and a user with the username user100 is a member of the group. You enter group100 for the value of **`<YOUR_LDAP_USER_GROUP>`**.
   - **ACCEPT LICENSE** set the accept value to **`true`** to accept the license.

2. Use the following oc command to obtain the console route and modify it to derive your Infrastructure management route.

   ```
   oc get ingress.config.openshift.io -o=jsonpath='{.items[0].spec.domain}'
   ```

   Example output can resemble the following example:

   ```
   apps.mycluster.myibm.com
   ```

3. Add **`inframgmtinstall`** to the output to create **`<YOUR_IM_HTTPD_ROUTE>`**. For example, **`inframgmtinstall.apps.mycluster.myibm.com`**
4. Deploy the installation instance CR YAML using one of the following methods:
   - Using the CLI login to your OpenShift cluster where Infrastructure Automation is installed. For example,

```
oc login --token=`<sha256~EUe-BThLn-qRm32K16QDqLMCQZz5VKlM42JKIj2-U7M>` --
server=https://`<api.mycluster.myibm.com>`:6443

oc apply -f im-install-cr.yaml
```

- Using the OpenShift Console.

  Create an IMInstall instance to deploy Infrastructure management.

  - Navigate to **Operators > Installed Operators.**
  - Switch the project to cp4aiops.
  - Click IBM Infrastructure Management Install.
  - Click Create Instance in the IMInstall tile.
  - Switch to YAML view.
  - Copy and paste the installation instance yaml created in step 1.
  - Click Create.

5. Verify the pods are running (It can take a few minutes for the pods to start) with the command:

   ```
   oc get pods -n cp4aiops
   ```

   The following Infrastructure management pods should be running.

   - 1-event-handle-*
   - 1-generic-*
   - 1-priority-*
   - 1-remote-console-*
   - 1-reporting-*
   - 1-schedule-*
   - 1-ui-*
   - 1-web-service-*
   - httpd-*
   - ibm-infra-management-application-*
   - memcached-*

Custom resource definitions

Use the following information to learn about the custom resource definitions (CRDs) used by Infrastructure Automation.

For more information about API conventions and definitions, see API Conventions in the Kubernetes documentation.

- Infrastructure Automation custom resource definitions

Infrastructure Automation custom resource definitions

Use this information to learn about the custom resource definitions (CRDs) used by Infrastructure Automation.

In the following table, CRDs that do not have a hyperlink must not be manually edited unless otherwise instructed, and are manipulated only by the operator.

Table 1. Infrastructure Automation CRDs

Kind	API version	Operator name	Description
IAConfig	v1alpha1	ibm-infrastructure-automation-operator	Build, deploy, and manage VMs and Kubernetes clusters. Enable self-service orchestration of resources across clouds. Learn More.
ManageService	v1alpha1	ibm-management-cam-install	ManageService is the Schema for the manageservices API
ManageIQ	v1alpha1	ibm-management-im-install	ManageIQ is the Schema for the manageiqs API
IMInstall	v1alpha1	ibm-management-im-install	IMInstall is the Schema for the iminstalls API

IAConfig/v1alpha1

Build, deploy, and manage VMs and Kubernetes clusters. Enable self-service orchestration of resources across clouds. Learn More.

spec

Description: IAConfigSpec defines the desired state of IAConfig

Type: object

Table 1. Description of IAConfig/v1alpha1 spec

Property	Type	Description

Property	Type	Description
spec	object	IAConfigSpec defines the desired state of IAConfig
spec.imagePullSecret	string	Select a secret to serve as the ImagePullSecret
spec.infraAutoComposableComponents	array	Enable Infrastructure management (im) and/or Manage services (cam).
spec.license	object	License acceptance is required to create resources.
spec.license.accept	boolean	I accept the product license after reading it at https://ibm.biz/cp4aiops-481-license
spec.skipAutomationBaseStatus	boolean	This field is reserved for support person to use.
spec.skipAutomationUiConfigStatus	boolean	This field is reserved for support person to use.
spec.storageClass	string	Select a storage class that supports ReadWriteMany access mode.
spec.storageClassLargeBlock	string	Select a storage class with a large block size (for example, 64k).

Important: The following fields are deprecated and should not be used in Infrastructure Automation 4.8.1:

- skipAutomationBaseStatus
- skipAutomationUiConfigStatus

status

Description: IAConfigStatus defines the observed state of IAConfig

Type: object

Table 2. Description of IAConfig/v1alpha1 status

Property	Type	Description
status	object	IAConfigStatus defines the observed state of IAConfig
status.failureMessage	string	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
status.phase	string	InstallationPhase describes the phase of the overall Installation
status.storageclass	string	(No Description)
status.storageclasslargeblock	string	(No Description)
status.version	string	(No Description)

ManageService/v1alpha1

ManageService is the Schema for the manageservices API.

spec

Description: ManageServiceSpec defines the desired state of ManageService.

Type: object

Table 1. Description of IAManageService spec

Property	Type	Description	Default
camAPI	object	Configuration for Cloud Automation Manager API microservices	`{"camComposerApi":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camComposerUi":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camIaas":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camMongo":{"resources":{"limits":{"cpu":"500m","memory":"4Gi"},"requests":{"cpu":"100m","memory":"2Gi"}}},"camOrchestration":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camPortalUi":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camProviderTerraformApi":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}},"terraformSSHDir":{"terraformSSHDirSecret":""}},"camSecret":{"secretName":"cam-api-secret"},"camServiceLibraryUi":{"resources":{"limits":{"cpu":"300m","memory":"512Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camServiceLibraryUiAPi":{"resources":{"limits":{"cpu":"100m","memory":"512Mi"},"requests":{"cpu":"50m","memory":"256Mi"}}},"camTenantApi":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"camUiBasic":{"resources":{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}},"certificate":{"certName":"cert"},"replicaCount":1}`
camAPI.replicaCount	int	Number of replicas for cam microservices	`1`

Property	Type	Description	Default
camController.replicaCount	int	Number of replicas for Controller	`1`
camLoggingPolicies	object	Configuration for the logging level used with Cloud Automation Manager microservices	`{"logLevel":"info"}`
camLoggingPolicies.logLevel	string	Log Level for logging	`"info"`
camMongoPV	object	MongoDB Persistant Storage	`{"name":"cam-mongo-pv","persistence":{"accessMode":"ReadWriteOnce","enabled":true,"existingClaimName":"","size":"15Gi","storageClassName":"","useDynamicProvisioning":false}}`
camMongoPV.persistence.accessMode	string	Access Mode for Storage	`"ReadWriteOnce"`
camMongoPV.persistence.enabled	bool	Enable persistance	`true`
camMongoPV.persistence.existingClaimName	string	Specify the name of the Existing Claim to be used by your application, empty string means don't use an existClaim	`""`
camMongoPV.persistence.size	string	MongoDB PVC size (in Gi)	`"15Gi"`
camMongoPV.persistence.storageClassName	string	Specify the name of the StorageClass empty string means don't use a StorageClass	`""`
camMongoPV.persistence.useDynamicProvisioning	bool	useDynamicProvisioning for persistence	`false`
camProviderTerraform	object	Configuration for Cloud Automation Manager Provider Terraform	`{"isolateRuntime":false,"replicaCount":1,"runtime":{"replicaCount":0},"terraformSSHDir":{"terraformSSHDirSecret":""}}`
camProviderTerraform.isolateRuntime	bool	Select to enable isolation mode for Provider Terraform runtime. In this mode, the runtime runs in separate pods and have read only access to the cam-terraform-pv.	`false`
camProviderTerraform.replicaCount	int	Number of replicas for camProviderTerraformApi	`1`
camProviderTerraform.runtime	object	Configuration for Cloud Automation Manager Provider Terraform runtime	`{"replicaCount":0}`

Property	Type	Description	Default
camProviderTerraform.runtime.replicaCount	int	Number of replicas for Provider Terraform Runtime. This value is only applicable when camProviderTerraform.isolateRuntime is enabled	`0`
camProviderTerraform.terraformSSHDir	object	Cloud Automation Manager Provider Terraform SSH Directory	`{"terraformSSHDirSecret":""}`
camProviderTerraform.terraformSSHDir.terraformSSHDirSecret	string	Secret name for Cloud Automation Manager Provider Terraform SSH Directory Contents	`""`
camProxy	object	CAM Proxy configuration	`{"customNginxSSLCertificate":{"customNginxSSLCertName":""},"limits":{"cpu":"500m","memory":"1500Mi"},"replicaCount":1,"requests":{"cpu":"100m","memory":"256Mi"},"resources":null}`
camProxy.customNginxSSLCertificate.customNginxSSLCertName	string	Nginx SSL Custom Certificate Secret Name	`""`
camProxy.limits.cpu	string	The CPU limit	`"500m"`
camProxy.limits.memory	string	The memory limit	`"1500Mi"`
camProxy.replicaCount	int	Number of replicas for Proxy	`1`
camProxy.requests.cpu	string	The requested CPU	`"100m"`
camProxy.requests.memory	string	The requested memory	`"256Mi"`
camUI	object	UI Configuration	`{"camUISecret":{"secretName":"cam-ui-secret","sessionKey":"opsConsole.sid","sessionTTL":"43200"},"replicaCount":1}`
camUI.camUISecret	object	User Interface Secret	`{"secretName":"cam-ui-secret","sessionKey":"opsConsole.sid","sessionTTL":"43200"}`
camUI.camUISecret.secretName	string	User interface Secret Name	`"cam-ui-secret"`
camUI.camUISecret.sessionKey	string	Session Key	`"opsConsole.sid"`
camUI.camUISecret.sessionTTL	string	Session timeout in seconds	`"43200"`
camUI.replicaCount	int	Number of replicas for UI	`1`
database	object	Optionally use an external MongoDB	`{"bundled":true}`
database.bundled	bool	Use internal MongoDB. Set to false for external. For more information, see Using an external MongoDB in the IBM Knowledge Center	`true`
license	object	License acceptance	`{"accept":true}`

Property	Type	Description	Default
license.accept	bool	Accept the license to use the product"	`true`
proxy	object	Configuration for a proxy used with Cloud Automation Manager. Supply details in the secure values secret.	`{"useProxy":false}`
proxy.useProxy	bool	Select to use a proxy with CAM and fill out the fields in the Secure Values secret (see the Knowledge Center install instructions for more details)	`false`
resources	object	Amount of memory and CPU to assign to Cloud Automation Manager microservices	`{"limits":{"cpu":"500m","memory":"1500Mi"},"requests":{"cpu":"100m","memory":"256Mi"}}`
resources.limits.cpu	string	The memory limit	`"500m"`
resources.limits.memory	string	The CPU limit	`"1500Mi"`
resources.requests.cpu	string	The requested CPU	`"100m"`
resources.requests.memory	string	The requested memory	`"256Mi"`
roks	bool	Select to indicate it is Red Hat OpenShift on IBM Cloud System	`false`
roksRegion	string	Red Hat OpenShift on IBM Cloud Storage Region selection	`""`
roksZone	string	Red Hat OpenShift on IBM Cloud Storage Zone selection	`""`

IMInstall/v1alpha1

IMInstall is the Schema for the iminstalls API

spec

Description: IMInstallSpec defines the desired state of IMInstall

Type: object

Table 1. Description of IMInstall/v1alpha1 spec

Property	Type	Description
spec	object	IMInstallSpec defines the desired state of IMInstall
spec.appAnnotations	object	Optional Annotations to apply to the Httpd, Kafka, Memcached, Orchestrator and PostgresQL Pods
spec.appName	string	Application name used for deployed objects (default: iminstall)

Property	Type	Description
spec.applicationDomain	string	Domain name for the external route. Used for external authentication configuration
spec.applicationImage	string	Image string used for the infrastructure management application deployment. By default it is based on the olm.relatedImage.application annotation
spec.backupLabelName	string	This label will be applied to essential resources that need to be backed up (default: manageiq.org/backup)
spec.baseWorkerImage	string	Image string used for the base worker deployments By default this is determined by the orchestrator pod
spec.databaseRegion	string	Database region number (default: 0)
spec.databaseSecret	string	Secret containing the database access information, content generated if not provided (default: postgresql-secrets)
spec.databaseVolumeCapacity	string	Database volume size (default: 50Gi)
spec.deployMessagingService	boolean	Deprecated: Flag to indicate if Kafka and Zookeeper should be deployed (default: false)
spec.enableApplicationLocalLogin	boolean	Flag to allow logging into the application without SSO (default: false)
spec.enableSSO	boolean	Flag to enable SSO in the application (default: false)
spec.enforceWorkerResourceConstraints	boolean	Flag to trigger worker resource constraint enforcement (default: true)
spec.httpdAuthConfig	string	Secret containing the httpd configuration files Mutually exclusive with the OIDCClientSecret and OIDCProviderURL if using openid-connect
spec.httpdAuthenticationType	string	Type of httpd authentication (default: internal) Options: internal, external, active-directory, saml, openid-connect Note: external, active-directory, and saml require an httpd container with elevated privileges
spec.httpdCpuLimit	string	Httpd deployment CPU limit (default: 1000m)
spec.httpdCpuRequest	string	Httpd deployment CPU request (default: 50m)
spec.httpdImage	string	Image string used for the httpd deployment (default: os.Getenv("HTTPD_IMAGE"))
spec.httpdImageNamespace	string	Image namespace used for the httpd deployment (default: docker-na-public.artifactory.swg-devops.com/hyc-cp4mcm-team-docker-local/infra-mgmt) Note: the exact image will be determined by the authentication method selected
spec.httpdImageTag	string	Image tag used for the httpd deployment (default: 2-el9)
spec.httpdMemoryLimit	string	Httpd deployment memory limit (default: 200Mi)
spec.httpdMemoryRequest	string	Httpd deployment memory request (default: 100Mi)
spec.imagePullSecret	string	Secret containing the image registry authentication information needed for the manageiq images
spec.initialAdminGroupName	string	Group name to create with the super admin role. This can be used to seed a group when using external authentication
spec.internalCertificatesSecret	string	Secret containing all of the necessary certificates to secure communication between pods (default: internal-certificates-secret)
spec.kafkaCpuRequest	string	Kafka deployment CPU request (default: 250m)
spec.kafkaCpulimit	string	Kafka deployment CPU limit (default: 250m)
spec.kafkaImage	string	Image string used for the kafka deployment (default: os.Getenv("KAFKA_IMAGE"))
spec.kafkaImageName	string	Image used for the kafka deployment (default: docker.io/bitnami/kafka)
spec.kafkaImageTag	string	Image tag used for the kafka deployment (default: latest)
spec.kafkaMemoryLimit	string	Kafka deployment memory limit (default: 1024Mi)
spec.kafkaMemoryRequest	string	Kafka deployment memory request (default: 256Mi)
spec.kafkaSecret	string	Secret containing the kafka access information, content generated if not provided (default: kafka-secrets)
spec.kafkaVolumeCapacity	string	Kafka volume size (default: 1Gi)
spec.license	object	Opt-in license acceptance is required to create resources
spec.license.accept	boolean	Accept is an opt-in license acceptance required to deploy resources
spec.memcachedCpuLimit	string	Memcached deployment CPU limit (default: 1000m)
spec.memcachedCpuRequest	string	Memcached deployment CPU request (default: 50m)
spec.memcachedImage	string	Image string used for the memcached deployment (default: :)

Property	Type	Description
spec.memcachedImageName	string	Image used for the memcached deployment (default: manageiq/memcached)
spec.memcachedImageTag	string	Image tag used for the memcached deployment (default: 1-el9)
spec.memcachedMaxConnection	string	Memcached max simultaneous connections (default: 1024)
spec.memcachedMaxMemory	string	Memcached item memory in megabytes (default: 64)
spec.memcachedMemoryLimit	string	Memcached deployment memory limit (default: 200Mi)
spec.memcachedMemoryRequest	string	Memcached deployment memory request (default: 100Mi)
spec.memcachedSlabPageSize	string	Memcached max item size (default: 1m, min: 1k, max: 1024m)
spec.migrationsRan	array	A list of CR data migrations that have been run
spec.oidcAuthIntrospectionURL	string	URL for OIDC authentication introspection Only used with the openid-connect authentication type. If not specified, the operator will attempt to fetch its value from the "introspection_endpoint" field in the Provider metadata at the OIDCProviderURL provided.
spec.oidcCaCertSecret	string	Secret containing the trusted CA certificate file(s) for the OIDC server. Only used with the openid-connect authentication type
spec.oidcClientSecret	string	Secret name containing the OIDC client id and secret Only used with the openid-connect authentication type
spec.oidcProviderURL	string	URL for the OIDC provider Only used with the openid-connect authentication type
spec.orchestratorCpuLimit	string	Orchestrator deployment CPU limit (default: 1000m)
spec.orchestratorCpuRequest	string	Orchestrator deployment CPU request (default: 200m)
spec.orchestratorImage	string	Image string used for the orchestrator deployment (default: / :)
spec.orchestratorImageName	string	Image name used for the orchestrator deployment (default: infra-mgmt-orchestrator)
spec.orchestratorImageNamespace	string	Image namespace used for the orchestrator and worker deployments (default: manageiq)
spec.orchestratorImageTag	string	Image tag used for the orchestrator and worker deployments (default: latest-18.y)
spec.orchestratorInitialDelay	string	Number of seconds to wait before starting the orchestrator liveness check (default: 2400)
spec.orchestratorMemoryLimit	string	Orchestrator deployment memory limit (default: 2048Mi)
spec.orchestratorMemoryRequest	string	Orchestrator deployment memory request (default: 1024Mi)
spec.postgresqlCpuLimit	string	PostgreSQL deployment CPU limit (default: 10000m)
spec.postgresqlCpuRequest	string	PostgreSQL deployment CPU request (default: 500m)
spec.postgresqlImage	string	Image string used for the postgresql deployment (default: :)
spec.postgresqlImageName	string	Image used for the postgresql deployment (Default: docker.io/manageiq/postgresql)
spec.postgresqlImageTag	string	Image tag used for the postgresql deployment (Default: 13-el9)
spec.postgresqlMaxConnections	string	PostgreSQL maximum connection setting (default: 1000)
spec.postgresqlMemoryLimit	string	PostgreSQL deployment memory limit (default: 8192Mi)
spec.postgresqlMemoryRequest	string	PostgreSQL deployment memory request (default: 2048Mi)
spec.postgresqlSharedBuffers	string	PostgreSQL shared buffers setting (default: 1GB)
spec.serverGuid	string	Server GUID (default: auto-generated)
spec.storageClassName	string	StorageClass name that will be used by manageiq data stores
spec.tlsSecret	string	Secret containing the tls cert and key for the ingress, content generated if not provided (default: tls-secret)
spec.uiWorkerImage	string	Image string used for the UI worker deployments By default this is determined by the orchestrator pod
spec.opentofuRunnerImage	string	Image string used for the Opentofu runner deployments By default this is determined by the orchestrator pod

Property	Type	Description
spec.webserverWorkerImage	string	Image string used for the webserver worker deployments By default this is determined by the orchestrator pod
spec.zookeeperCpuRequest	string	Zookeeper deployment CPU request (default: 250m)
spec.zookeeperCpulimit	string	Zookeeper deployment CPU limit (default: 250m)
spec.zookeeperImage	string	Image string used for the zookeeper deployment (default: :)
spec.zookeeperImageName	string	Image used for the zookeeper deployment (default: docker.io/bitnami/zookeeper)
spec.zookeeperImageTag	string	Image tag used for the zookeeper deployment (default: latest)
spec.zookeeperMemoryLimit	string	Zookeeper deployment memory limit (default: 256Mi)
spec.zookeeperMemoryRequest	string	Zookeeper deployment memory request (default: 256Mi)
spec.zookeeperVolumeCapacity	string	Zookeeper volume size (default: 1Gi)

status

Description: IMInstallStatus defines the observed state of IMInstall

Type: object

Table 1. Description of IMInstall/v1alpha1 status

Property	Type	Description
status	object	IMInstallStatus defines the observed state of IMInstall

Troubleshooting installing and uninstalling Infrastructure Automation

Review frequently encountered issues related to installing, upgrading and uninstalling Infrastructure Automation.

Installation issues

- Database is stuck on retrying on Infrastructure Automation IMInstall Power Environment
- The `cam-tenant-api` pod is not in a ready state after installing the IAConfig CR
- Offline install or upgrade throws 'invalid character' error
- Offline install or upgrade stuck because the 'oc ibm-pak generate mirror-manifests' command fails with 'no space left on device'

Uninstall issues

- Uninstall fails to remove Infrastructure Management cleanly after deleting IAConfig instance
- Uninstall hangs when uninstalling Managed services

Troubleshooting installation

Database is stuck on retrying on Infrastructure Management IMInstall Power Environment

Sometimes when Infrastructure Management is deployed, the orchestrator pod fails to start properly with the following in the oc logs:

```
Cannot connect to the database!
Deployment status is
```

Check the postgresql pod to ensure it's running and listening. Then, verify the **vmdb_production** database was created by rushing into the postresql pod:

```
sh-4.4$ psql -U postgres
psql (10.19)
Type "help" for help.

postgres=# \l
                                  List of databases
      Name       |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------------+----------+----------+------------+------------+
 postgres        | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
                 |          |          |            |            | postgres=CTc/postgres
 template1       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
                 |          |          |            |            | postgres=CTc/postgres
(3 rows)
```

Solution: If the `vmdb_production` database doesn't exist, like above, it is possible that a remnant postgresql.conf in the DATA directory is preventing the postgresql image from initializing the new database on boot.

You will need to check the DATA directory and either clean it manually or recreate the PVC/PV and check that it is empty.

You can always find the location of the DATA directory using the psql command below:

```
postgres=# show data_directory;
        data_directory
------------------------------
 /var/lib/pgsql/data/userdata
(1 row)
```

Ensure that the data directory is empty before the postgresql pod is restarted.

The `cam-tenant-api` pod is not in a ready state after installing the IAConfig CR

After you install Infrastructure Automation, you can encounter an error where the `cam-tenant-api` pod displays as running, but not in a ready state. When this error occurs, you can see the following message:

```
[ERROR] init-platform-security - >>>>>>>>>> Failed to configure Platform Security. Will retry in 60 seconds <<<<<<<<<<<<
OperationalError: [object Object]
```

If this error occurs, delete the `cam-tenant-api` pod to cause the pod to restart and attempt to enter a ready state.

Offline install or upgrade throws 'invalid character' error.

When doing an offline install or upgrade, running the `oc ibm-pak generate mirror-manifests <..>` command throws an error similar to the following:

```
Error: failed to load the catalog FBC at cp.stg.icr.io/cp/ <...> invalid character '<' in string escape code
```

Solution: You must have `IBM Catalog Management Plug-in for IBM Cloud Pak` (`ibm-pak-plugin`) v1.10 or higher installed. Run the following commands to ensure that `ibm-pak-plugin` is at the required level.

1. Check which version of `ibm-pak-plugin` you have installed.

   Run the following command on your bastion host, portable compute device, or connected compute device if you are using a portable storage device.

   ```
   oc ibm-pak --version
   ```

   Example output:

   ```
   oc ibm-pak --version
   v1.11.0
   ```

2. If the `ibm-pak-plugin` version is lower than v1.10.0, then you must download and install the most recent version.

   Follow the steps for your installation approach:

   - Bastion host: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.
   - Portable device: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.

Offline install or upgrade stuck because the 'oc ibm-pak generate mirror-manifests' command fails with 'no space left on device'

The `oc ibm-pak generate mirror-manifests $IA_CASE_NAME $TARGET_REGISTRY --version $IA_CASE_VERSION` command fails with a message similar to the following in `$IBMPAK_HOME/.ibm-pak/logs/oc-ibm_pak.log`:

```
write /tmp/render-unpack-4002583241/var/lib/rpm/Packages: no space left on device
```

Solution: The default temporary directory does not have enough space to run the `ibm-pak` tool. You must set the `TMPDIR` environment variable to a different directory with more space before running the `oc ibm-pak generate mirror-manifests` command.

```
TMPDIR=<new_temp_dir> oc ibm-pak generate mirror-manifests $IA_CASE_NAME $TARGET_REGISTRY --version $IA_CASE_VERSION
```

Where `<new_temp_dir>` is the path for a directory with more space.

Troubleshooting uninstall

Uninstall fails to remove Infrastructure Management cleanly after deleting IAConfig instance

After deleting Infrastructure Management IAConfig CR in the namespace to uninstall Infrastructure Management from Red Hat OpenShift Container Platform console, some of the Infrastructure Management pods were still there in the namespace. You can workaround the problem by deleting IMInstall custom resource from the Red Hat OpenShift Container Platform console. This will remove the pods from the namespace.

You should also delete the `clients.oidc.security.ibm.com` customer resource that was used by Infrastructure Management if it still exists. Then, You can uninstall the operators remaining on the namespace.

To delete IMInstall Custom Resource (CR), you need to edit the CR to remove the Finalizer. Save the CR. Then delete the IMInstall Custom Resource (CR). Run the following commands or perform the same from the Red Hat OpenShift Container Platform console.

1. Identify the CR name, for example, **im-iminstall**

   ```
   oc -n <project> get iminstall
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed.

2. Edit the statement "finalizers:" and the one below it. Save and exit.

   ```
   oc -n <project> edit iminstall im-iminstall
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

3. Delete the CR.

   ```
   oc -n <project> delete iminstall im-iminstall
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

After a few minutes, the Infrastructure Management pods should no longer exist. The clients.oidc.security.ibm.com Custom Resource (CR) could still exist, and to delete the CR, do the following steps,

1. Check if the CR exist.

   ```
   oc -n <project> get clients.oidc.security.ibm.com
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

2. If the CR exists, edit the CR. Delete the statement "finalizers:" and the one below. Save and exit.

   ```
   oc -n <project> edit clients.oidc.security.ibm.com ibm-infra-management-application-client
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

3. Delete the CR.

   ```
   oc -n <project> delete clients.oidc.security.ibm.com ibm-infra-management-application-client
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

Uninstall hangs when uninstalling Managed services

The Infrastructure Automation uninstallation sometimes hangs when trying to delete the manageservice instance.

Describing the Managed services instance shows output similar to the following:

```
# oc describe manageservice cam
Name:         cam
Namespace:    cp4aiops
Labels:       operator.ibm.com/opreq-control=true
Annotations:  operator-sdk/primary-resource: /cam-services-sa-csb-patch-rb-pod
              operator-sdk/primary-resource-type: ClusterRoleBinding.rbac.authorization.k8s.io
API Version:  cam.management.ibm.com/v1alpha1
Kind:         ManageService
Metadata:
  Creation Timestamp:             2024-03-05T21:46:56Z
  Deletion Grace Period Seconds:  0
  Deletion Timestamp:             2024-04-15T01:11:29Z
  Finalizers:
    helm.sdk.operatorframework.io/uninstall-release
  Generation:        2
  Resource Version:  55077252
  UID:               87e6b8cf-3123-4914-bf64-a2b209cbe22d
```

Solution: Edit the Managed services instance to remove the finalizer entry.

1. Run the following command:

   ```
   oc edit ManageService cam
   ```

2. Delete the following lines, and then save your changes.

   ```
   Finalizers:
       helm.sdk.operatorframework.io/uninstall-release
   ```

Upgrading Infrastructure Automation

Upgrade an existing online or offline deployment of Infrastructure Automation to 4.8.1.

Upgrade paths

Infrastructure Automation uses a versioning system of X.Y.Z, where X=version, Y=release, and Z=patch. For example, Infrastructure Automation 4.8.1 is version 4, release 8, patch 1. Infrastructure Automation upgrades can be release upgrades or patch upgrades.

Stand-alone Infrastructure Automation

If you have a stand-alone deployment of Infrastructure Automation 4.4.x or later, then you can upgrade directly to the latest version of Infrastructure Automation.

Deployment of Infrastructure Automation with IBM Cloud Pak® for AIOps

You cannot use these instructions to upgrade deployments of Infrastructure Automation 4.6.1 or earlier.

You can upgrade by a maximum of one release at a time. If the release that you want to upgrade from is more than one release behind the release that you are upgrading to, then you must upgrade sequentially. For example, you cannot upgrade directly from Infrastructure Automation 4.4.0 to v4.8.1, you must use the following sequence:

1. Use the v4.5.1 documentation to upgrade from Infrastructure Automation 4.4.1 to v4.5.1.
2. Use the v4.6.1 documentation to upgrade from Infrastructure Automation 4.5.1 to v4.6.1.
3. Use the v4.7.1 documentation to upgrade from Infrastructure Automation 4.6.1 to v4.7.1.
4. Use this documentation, the v4.8.1 documentation to upgrade from Infrastructure Automation 4.7.1 to v4.8.1.

Controlling upgrade

Patch upgrades

Online deployments

Upgrades of online Infrastructure Automation deployments to a higher patch, for example from Infrastructure Automation 4.8.0 to 4.8.1, complete automatically if catalog polling is enabled. You elect to enable or disable catalog polling when you install Infrastructure Automation, and you can change this setting after installation if necessary. Automatic patch upgrade can also occur if you have not fixed the **`ibm-operator-catalog`** CatalogSource to a specific image and the **`ibm-operator-catalog`** pods are restarted, even if catalog polling is disabled.

If you do not want your Infrastructure Automation deployment to upgrade automatically when newer patches are available, then you must disable catalog polling and pin the image to a specific version. For more information, see Configuring automatic patch upgrades.

Offline deployments

Upgrades of offline Infrastructure Automation deployments to a higher patch do not occur automatically, as the updated installation images must be manually mirrored to the offline repository first.

Release upgrades

Online deployments

Upgrades of online Infrastructure Automation deployments to a higher release, for example from Infrastructure Automation 4.7.0 to 4.8.1 do not complete automatically, even if catalog polling is enabled. For a release upgrade, catalog polling must be enabled and the channel subscription must also be manually edited.

Offline deployments

Upgrades of offline deployments to a higher release version do not occur automatically, as the updated installation images must be manually mirrored to the offline repository first.

Upgrade procedure

Important: If you are planning to upgrade Red Hat OpenShift as part of an upgrade to Infrastructure Automation 4.8.1, then you must complete the Infrastructure Automation upgrade before you upgrade Red Hat OpenShift.

The upgrade procedure depends on whether your deployment is an online or offline (air-gapped) deployment:

- Online upgrade of Infrastructure Automation (CLI method)
- Online upgrade of Infrastructure Automation (console method)
- Offline upgrade of Infrastructure Automation

Upgrading Infrastructure Management appliances

The following topics describe the process of upgrading Infrastructure Management appliances when a new release of Infrastructure Automation is available.

For more information about migrating from IBM Red Hat CloudForms or IBM Cloud Pak for Multicloud Management appliances to Infrastructure Automation, see these instructions to complete a full appliance replacement upgrade.

- Migrating to an Infrastructure management appliance

For more information about upgrading Infrastructure Management appliances, see:

- Upgrading Infrastructure management appliances

For more information about migrating from an appliance installation to a Kubernetes (podified) installation of Infrastructure Management, see the following topic.

- Migrate from an appliance-based to a Kubernetes-based Infrastructure Management installation

Online upgrade of Infrastructure Automation (console)

Follow these steps to use the Red Hat® OpenShift® Container Platform console to upgrade Infrastructure Automation to 4.8.1.

This procedure can be used for an online deployment of Infrastructure Automation, and can still be used if the deployment has had hotfixes applied. If you have an offline deployment, follow the instructions in Offline upgrade of Infrastructure Automation.

Before you begin

- Ensure that your cluster meets all of the prerequisites for deployment. For more information, see Planning.
- Red Hat OpenShift requires a user with `cluster-admin` privileges for the following operations:
  - Configure automatic catalog polling
  - Update the operator subscription

Restrictions

- There are restrictions on which releases you can upgrade from. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your online Infrastructure Automation deployment.

1. Configure automatic catalog polling
2. Update the operator subscription
3. Verify the deployment
4. Post upgrade actions

1. Configure automatic catalog polling

Ensure that your catalog is set to automatically poll for updates.

Your `ibm-operator-catalog` CatalogSource object can be configured to automatically poll for a newer version, and to retrieve it if one is available. This triggers an automatic update of your deployment. Polling for updates is enabled by configuring the following polling attribute:

`spec.updateStrategy.registryPoll`

You might have already elected to automatically accept product updates by adding the polling attribute to your ibm-operator-catalog YAML when you installed Infrastructure Automation, installed an Infrastructure Automation hotfix from IBM Support, or when you installed another IBM Cloud Pak.

Use the following steps to check whether you already have a polling attribute set, and configure it if you do not.

Note: `ibm-operator-catalog` also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster, and you enable the polling attribute, then automatic update is configured for all of them.

1. Log in to your OpenShift cluster console.
2. Go to **Administration** > **Cluster Settings**.
3. Click **Configuration**, and then scroll down and click **OperatorHub**.
4. Select **Sources**, scroll down, and click `ibm-operator-catalog`.
5. Click **YAML** to switch to the YAML view.
6. If there is not a `spec.updateStrategy` section, or `spec.image` is not set to `icr.io/cpopen/ibm-operator-catalog:latest`, then update the YAML to have the following in the `spec` block, and save it.

```
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

2. Update the operator subscription

Skip this section if you are upgrading from Infrastructure Automation v4.8.0 to v4.8.1, and proceed to section 3 Verify the deployment.

Update the `spec.channel` value of the Infrastructure Automation operator's subscription to the release that you want to upgrade to, v4.8.

1. From OpenShift cluster console, select **Home** > **Search**.
2. From the **Project list**, select the **project (namespace)** that your IBM Infrastructure Automation subscription is deployed in. This is your IBM Infrastructure Automation project if your deployment is namespace scoped, or **`openshift-operators`** if your deployment has a cluster wide scope.
3. In the **Resources list**, select **SUB Subscription**. A list of subscriptions is displayed.
4. Click the subscription that has the name of **`ibm-infrastructure-automation-operator`**. A new window with the subscription details for **`ibm-infrastructure-automation-operator`** is displayed.
5. Click the value in the **Update channel** box. A new window called *Change Subscription update channel* is displayed.
6. Change the **channel** to **v4.8** and click **Save**.

Important: You must only update the Infrastructure Automation subscription. You must not update the Infrastructure Management or Managed Services subscriptions (**`ibm-management-im-install`** and **`ibm-management-cam-install`**).

3. Verify the deployment

Check the status of the Infrastructure Automation upgrade to make sure that it shows a status of **`Phase:Installing`** and eventually changes to **`Running`**.

1. Click **Operators** > **Installed Operators**.
2. From the **Project list**, select the **project (namespace)** that IBM Infrastructure Automation is deployed in if your deployment is namespace scoped, or **`openshift-operators`** if your deployment has a cluster-wide scope.
3. Locate **IBM Infrastructure Automation** in the list, and verify that the annotation underneath it shows 4.8.1 as version.
4. Select **IBM Infrastructure Automation** and then click **IBM Infrastructure Automation** under the **Provided APIs** column, and verify that it has a *Status* of **`Phase: Running`**, which means that your deployment is successful.

Note: Do not proceed until Infrastructure Automation has a status of **`Phase: Running`**.

4. Post upgrade actions

1. Clear your browser cache. Failure to do so will result in the error *502 Bad Gateway* when viewing the **`Library`**, **`Instances`**, and **`Manage`** pages.
2. If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading Infrastructure Automation backup and restore artifacts.

Online upgrade of Infrastructure Automation (CLI method)

Use these instructions to upgrade Infrastructure Automation to 4.8.1.

This procedure can be used on an online deployment of Infrastructure Automation, and can still be used if the deployment has had hotfixes applied. If you have an offline deployment, follow the instructions in Offline upgrade of Infrastructure Automation.

Before you begin

- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with **`oc login`** for any steps that use the Red Hat® OpenShift® Container Platform command-line interface (CLI).
- Red Hat OpenShift requires a user with **`cluster-admin`** privileges for the following operations:
  - Configure automatic catalog polling
  - Update the operator subscription

Restrictions

- There are restrictions on which releases you can upgrade from. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your online Infrastructure Automation deployment.

1. Verify cluster readiness
2. Configure automatic catalog polling
3. Update the operator subscription
4. Verify the deployment
5. Post upgrade actions

1. Verify cluster readiness

1. Ensure that your cluster still meets all the prerequisites for deployment.

2. Run the following command to set the environment variable that you need for upgrade.

   ```
   export PROJECT_IA=<project>
   ```

   Where **<project>** is the namespace (project) that Infrastructure Automation is deployed in.

2. Configure automatic catalog polling

Verify that your catalog is set to automatically poll for updates. Your **ibm-operator-catalog** CatalogSource object can be configured to automatically poll for a newer version, and to retrieve it if one is available. This triggers an automatic update of your deployment. Polling for updates is enabled by configuring the following polling attribute:

```
spec.updateStrategy.registryPoll
```

You might have already elected to automatically accept product updates by adding the polling attribute to your ibm-operator-catalog YAML when you installed Infrastructure Automation, installed an Infrastructure Automation hotfix from IBM Support, or when you installed another IBM Cloud Pak.

Use the following steps to check whether you already have a polling attribute set, and to configure it if you do not.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster, and you enable the polling attribute, then automatic update is configured for all of them.

1. Run the following command to view and edit your ibm-operator-catalog CatalogSource instance.

   ```
   oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

2. If there is not a **spec.updateStrategy** section, then update the YAML to have the following contents, and save it.

   ```
   updateStrategy:
     registryPoll:
       interval: 45m
   ```

3. Update the operator subscription

Skip this section if you are upgrading from Infrastructure Automation v4.8.0 to v4.8.1, and proceed to section 4, Verify the deployment.

Update the **spec.channel** value of the Infrastructure Automation operator's subscription to the release that you want to upgrade to, v4.8.

```
oc patch sub ibm-infrastructure-automation-operator -n ${PROJECT_IA} --type=json -p='[{'op': 'replace', 'path':
'/spec/channel', 'value': 'v4.8'}]'
```

Important: You must only update the Infrastructure Automation subscription. You must not update the Infrastructure Management or Managed Services subscriptions (**ibm-management-im-install** and **ibm-management-cam-install**).

4. Verify the deployment

4.1 Check Infrastructure Automation upgrade status

Check the status of the Infrastructure Automation upgrade to make sure the PHASE is Installing and eventually it changes to Running.

```
oc get iaconfigs -n ${PROJECT_IA}
```

Example output:

```
NAME                PHASE         VERSION    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-ia-installer    Running       4.8.1      rook-cephfs     rook-ceph-block           8m44s
```

Important: Do not proceed until the message that Infrastructure Automation upgrade is Running is displayed.

4.2 Check the version

Run the following command and check that the *VERSION* that is returned is **4.8.1**.

```
oc get csv | grep -E "ibm-infrastructure-automation-operator|VERSION"
```

Example ouput:

```
NAME                                          DISPLAY                          VERSION   REPLACES
PHASE
ibm-infrastructure-automation-operator.v4.8.1  IBM Infrastructure Automation  4.8.1     ibm-infrastructure-automation-
operator.v4.8.0  Succeeded
```

4.3 Check the deployment

After a few minutes, run the following command from the CLI to verify that your Infrastructure Automation deployment is running again.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE        VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running      4.8.1     rook-cephfs    rook-ceph-block          8m44s
```

Do not proceed until Infrastructure Automation has a **PHASE** of **`Running`**.

5. Post upgrade actions

1. Clear your browser cache. Failure to do so will result in the error *502 Bad Gateway* when viewing the **`Library`**, **`Instances`**, and **`Manage`** pages.
2. If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading Infrastructure Automation backup and restore artifacts.

Offline upgrade of Infrastructure Automation

Use these instructions to upgrade Infrastructure Automation to 4.8.1.

This procedure can be used on an offline deployment of Infrastructure Automation, and can still be used if the deployment has had hotfixes applied. If you have an online deployment, follow the instructions in Online upgrade of Infrastructure Automation (CLI method).

Before you begin

- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with **`oc login`** for any steps that use the Red Hat® OpenShift® Container Platform command-line interface (CLI).
- Red Hat OpenShift Container Platform requires a user with **`cluster-admin`** privileges for the following operations:
  - Mirror images
  - Update the catalog
  - Update the operator subscription
- Ensure that your cluster meets all of the prerequisites for deployment. For more information, see Planning

Restrictions

- There are restrictions on which releases you can upgrade from. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your offline Infrastructure Automation deployment.

1. Verify prerequisites
2. Download CASE files
3. Mirror images
4. Update the catalog
5. Update the operator subscription
6. Verify the deployment
7. Post upgrade actions

1. Verify prerequisites

1. Ensure that your cluster still meets all the prerequisites for an air-gapped deployment.

   Review the steps for your installation approach:

   - Bastion host
   - Portable device
2. Ensure that the IBM Catalog Management Plug-in for IBM Cloud Pak® (**`ibm-pak-plugin`**) is at the required level.

   Important: **`ibm-pak-plugin`** v1.9 and lower cannot process the Infrastructure Automation 4.8.1 catalog. You must ensure that you have **`ibm-pak-plugin`** v1.10 or higher installed.

   1. Check which version of **`ibm-pak-plugin`** you have installed.

      Run the following command on your bastion host, portable compute device, or connected compute device if you are using a portable storage device.

      ```
      oc ibm-pak --version
      ```

      Example output:

      ```
      oc ibm-pak --version
      v1.13.0
      ```

   2. If the **`ibm-pak-plugin`** version is lower than v1.10.0, then you must download and install the most recent version.

      Follow the steps for your installation approach:

- Bastion host: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.
- Portable device: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.

3. Run the following command to set the environment variable that you need for upgrade.

```
export PROJECT_IA=<project>
```

Where **<project>** is the namespace (project) that Infrastructure Automation is deployed in.

2. Download CASE files

On your OpenShift cluster, rerun step 2 of the air-gap installation procedure **Set environment variables and download CASE files** to download the latest CASE files.

Follow the steps for your installation approach:

- Bastion host
- Portable device

Note: If you have an offline cache from a previous installation, you have two options to ensure that the latest CASEs are saved when you are using a case save command. You must:

- Delete the existing CASE .tgz files from your offline cache.
- Run a case save by using a different offline cache directory, and then copy the contents of that new directory into your existing offline cache.

Once you have the latest files in your offline cache, you can continue the airgap process using the offline cache directory.

3. Mirror images

Rerun step 3 of the air-gap installation procedure to mirror the updated images to the offline registry. Follow the steps for your installation approach:

- Bastion host: 3 Mirror images to your final location
- Portable device: 3 Mirror images

4. Update the catalog

Rerun step 5.1 or 5.2 of the air-gap installation procedure to update your catalog source.

Follow the steps for your installation approach:

- Bastion host: 5.2 Create the catalog source
- Portable device: 5.1 Create the catalog source

5. Update the operator subscription

Skip this section if you are upgrading from Infrastructure Automation v4.8.0 to v4.8.1, and proceed to section 6, Verify the deployment.

Update the **spec.channel** value of the Infrastructure Automation operator to the release that you want to upgrade to, v4.8.

```
oc patch sub ibm-infrastructure-automation-operator -n ${PROJECT_IA} --type=json -p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v4.8'}]'
```

Important: You must only update the Infrastructure Automation subscription. You must not update the Infrastructure Management or Managed Services subscriptions (**ibm-management-im-install** and **ibm-management-cam-install**).

6. Verify the deployment

6.1 Check Infrastructure Automation upgrade status

Check the status of the Infrastructure Automation upgrade to make sure the PHASE is Installing and eventually it changes to Running.

```
oc get iaconfigs -n ${PROJECT_IA}
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running       4.8.1     rook-cephfs    rook-ceph-block          8m44s
```

Important: Do not proceed until the message that Infrastructure Automation upgrade is Running is displayed.

6.2 Check the version

Run the following command and check that the *VERSION* that is returned is **4.8.1**.

```
oc get csv | grep -E "ibm-infrastructure-automation-operator|VERSION"
```

Example ouput:

```
NAME                                          DISPLAY                          VERSION   REPLACES
PHASE
```

```
ibm-infrastructure-automation-operator.v4.8.1  IBM Infrastructure Automation  4.8.1    ibm-infrastructure-automation-
operator.v4.8.0  Succeeded
```

6.3 Check the deployment

After a few minutes, run the following command from the CLI to verify that your Infrastructure Automation deployment is running again.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE        VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running      4.8.1     rook-cephfs    rook-ceph-block          8m44s
```

Important: Do not proceed until the message that Infrastructure Automation upgrade is Running is displayed.

7. Post upgrade actions

1. Clear your browser cache. Failure to do so will result in the error *502 Bad Gateway* when viewing the **Library**, **Instances**, and **Manage** pages.
2. If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading Infrastructure Automation backup and restore artifacts.

Configuring automatic patch upgrades

Disable or enable the automatic application of patch upgrades on your online IBM Infrastructure Automation deployment.

If you have an online IBM Infrastructure Automation deployment with catalog polling enabled and a new patch is available, then your deployment is automatically upgraded. If you do not want patch upgrades to occur automatically, then catalog polling must be disabled, and the image must be pinned to a specific version. For more information, see Controlling upgrade.

Disabling automatic patch upgrade

You can disable automatic patch upgrade at any time, using the Red Hat® OpenShift® Container Platform CLI or console.

Note: The **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks®. If you have multiple IBM Cloud Paks installed on your cluster and you alter the catalog polling attribute, then automatic update is altered for all of your IBM Cloud Paks.

Disabling automatic patch upgrades with the OpenShift CLI

1. Run the following command to view and edit your **ibm-operator-catalog** CatalogSource instance.

   ```
   oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

2. If there is a **spec.updateStrategy** section in the YAML like in the following example, then remove it and save your changes.

   ```
   updateStrategy:
     registryPoll:
       interval: 45m
   ```

3. Update the **ibm-operator-catalog** CatalogSource to fix it to use the current image digest, instead of **icr.io/cpopen/ibm-operator-catalog:latest**. This ensures that the **ibm-operator-catalog** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart.

   Run the following commands:

   ```
   IMGDIGEST=`oc get pods -n openshift-marketplace -l=olm.catalogSource=ibm-operator-catalog --no-headers -o=jsonpath="
   {.items[0].status.containerStatuses[0].imageID}" -n openshift-marketplace`
   CATALOGIMG=`oc get catalogsource ibm-operator-catalog -n openshift-marketplace -o jsonpath='{.spec.image}'`
   if [ "${CATALOGIMG}" = "icr.io/cpopen/ibm-operator-catalog:latest" ]; then
     oc patch catalogsource ibm-operator-catalog -n openshift-marketplace --type=json -p "[{ "op": "replace", "path":
   "/spec/image", "value": "\"$IMGDIGEST\"" }]"
   fi
   ```

Disabling automatic patch upgrades with the OpenShift console

1. Find the current image digest.
   1. Go to **Home** > **Projects**, and select **openshift-marketplace**.
   2. Click **Workloads** > **Pods**, and then search for **ibm-operator-catalog**.
   3. Click on the returned **ibm-operator-catalog-<...>** pod.
   4. Click **YAML** to switch to the YAML view.

5. Search for `imageID` in the YAML, and copy down the value of `spec.containerStatuses.imageID`. The value will be in a format similar to the following:

   ```
   icr.io/cpopen/ibm-operator-catalog@sha256:<...>
   ```

2. Log in to your OpenShift cluster's console.
3. Go to **Administration** > **Cluster Settings**.
4. Click **Configuration**, and then scroll down and click **OperatorHub**.
5. Select **Sources**, and then scroll down and click `ibm-operator-catalog`.
6. Click **YAML** to switch to the YAML view.
7. If there is a `spec.updateStrategy` section in the YAML, as in the following example, then remove it and save your changes.

   ```
   updateStrategy:
     registryPoll:
       interval: 45m
   ```

8. Set the value of `spec.image` to the value of the current image digest that you found in step 1, instead of to `icr.io/cpopen/ibm-operator-catalog:latest`, and save your changes. This ensures that the `ibm-operator-catalog` CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart.

Enabling automatic patch upgrade

You can enable automatic patch upgrade at any time, using the OpenShift CLI or console.

Use the following steps to configure your catalog to automatically poll for the latest images. This will cause patch upgrades to occur automatically, if a newer patch is available.

Note: `ibm-operator-catalog` also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster and you enable the polling attribute, then automatic update is configured for all of them.

Enabling automatic patch upgrades with the OpenShift CLI

1. Run the following command to view and edit your `ibm-operator-catalog` CatalogSource instance.

   ```
   oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

2. If there is not a `spec.updateStrategy` section, or `spec.image` is not set to `icr.io/cpopen/ibm-operator-catalog:latest`, then update the YAML to have the following contents, and save it.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
     updateStrategy:
       registryPoll:
         interval: 45m
   ```

Enabling automatic patch upgrades with the OpenShift console

1. Log in to your OpenShift cluster's console.
2. Go to **Administration** > **Cluster Settings**.
3. Click **Configuration**, and then scroll down and click **OperatorHub**.
4. Select **Sources**, and then scroll down and click `ibm-operator-catalog`.
5. Click **YAML** to switch to the YAML view.
6. If there is not a `spec.updateStrategy` section, or `spec.image` is not set to `icr.io/cpopen/ibm-operator-catalog:latest`, then update the YAML to have the following contents, and save it.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
     updateStrategy:
   ```

```
    registryPoll:
      interval: 45m
```

Migrating IBM Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances to Infrastructure Automation

This topic describes the process of doing a full appliance replacement upgrade when a new release of Infrastructure Automation – Infrastructure Management is available.

The following migration paths are supported:

- IBM Cloud Pak for Multicloud Management 2.3.x - Infrastructure Management to IBM Cloud Pak for AIOps 4.8.1 - Infrastructure Automation
- IBM Red Hat CloudForms 4.7 through 5.0 to IBM Cloud Pak for AIOps 4.8.1 - Infrastructure Automation

Ordering migrations

You must complete all migrations of the databases before you configure replication and high availability. The reason is because you deploy new appliances for migrating to Infrastructure Management.

Important: Do not run **evmserverd** if a database is in a partially migrated state.

Modifying the PostgreSQL configuration

This procedure modifies the strategy that is used for updating the PostgreSQL configuration. Previously, changes were made directly to the **postgresql.conf** file in the data directory. After you update the new configuration, changes will be made by editing a separate file in the **/etc/manageiq/postgresql.conf.d** directory. This modification allows future PostgreSQL configuration changes to be made more easily.

Note: The **01_miq_overrides.conf** file in the **/etc/manageiq/postgresql.conf.d** directory is reserved for application overrides and is re-created on application updates. **Important:** This file must not be edited. More files are read by the PostgreSQL server process according to the documentation here: PostgreSQL - Managing Configuration File Contents

1. Back up the **postgresql.conf** file so that you have any configuration changes:

   ```
   cp $APPLIANCE_PG_DATA/postgresql.conf <file_location>
   ```

2. Copy the updated **postgresql.conf** file to the **pg_data** directory:

   ```
   cp $APPLIANCE_SOURCE_DIRECTORY/TEMPLATE/var/opt/rh/rh-postgresql95/lib/pgsql/data/postgresql.conf
   $APPLIANCE_PG_DATA/postgresql.conf
   ```

3. Optional: If you updated the **postgresql.conf** file before this procedure, create a new **02_user_overrides.conf** file, and add any modifications that you have from the original **postgresql.conf** file. Example layout of the **02_user_overrides.conf** file:

   ```
   #--------------------------------------------
   # WRITE AHEAD LOG
   #--------------------------------------------

   wal_level = 'logical'
   wal_log_hints = on
   wal_buffers = 16MB
   min_wal_size = 1GB
   max_wal_size = 2GB
   checkpoint_completion_target = 0.9
   ```

   After the **02_user_overrides.conf** file is modified and saved, restart the postgresql service:

   ```
   systemctl restart $APPLIANCE_PG_SERVICE
   ```

All **postgresql.conf** customizations are now made to files in the **/etc/manageiq/postgresql.conf.d** directory due to the following **include_dir** line in the default configuration. The settings in these new files override ones in the **postgresql.conf** file.

```
include_dir = '/etc/manageiq/postgresql.conf.d`
```

Migrating to the Infrastructure Management appliance

Overview

This procedure describes the process of migrating to the Infrastructure Management appliance. This procedure does not necessarily include migration of all possible customer modifications, so it is recommended that you fully test any modifications before you migrate your environment.

Important:

- Read through all of the steps in this procedure before you begin the migration process.
- Infrastructure Management appliances require 12 GB memory. However, before you migrate your appliances, adjust resources in your environment to avoid performance issues.

You can classify the migration into three groups of appliances:

- VMDB appliance - An appliance with running workers that also hosts its own database that other appliances can also connect to.
- Non-VMDB appliance - An appliance with running workers that does not host a database. The appliance is connected to an external database.
- Dedicated database appliance - A Infrastructure Management appliance or non-Infrastructure Management virtual machine with no running workers on it: the appliance contains only a database for other appliances to connect to.

Migration workflow summary

In summary, the migration process follows this workflow:

1. Prepare appliances:

   Note: This step applies only for a multi-region Infrastructure Management environment. Remove all the subscriptions from the global region before you migrate the global database since the subscriptions must be dropped and re-created.

2. Back up the database of your appliance.

3. Migrate appliances:

   1. Provision new appliances to replace each appliance.

   2. On the database and nondatabase appliances, copy the GUID file from the version of the same appliance to the same location.

   3. On the database and nondatabase appliances, use the appliance console menu to stop the evmserverd service, manually remove the pre-configured database and key, and fetch the v2_key from an appliance.

   4. Copy the appliance database backup file and restore the database on each VMDB and dedicated database appliance.

   5. Load the new version of the pglogical library on the VMDB and dedicated database appliances.

      Note: This step applies only for a multi-region Infrastructure Management environment.

   6. Migrate the VMDB appliance databases and update the Automate Model.

   7. Restart **evmserverd** on the VMDB appliances.

4. Configure replication after the migration process is complete and appliances are running again.

Backing up current appliances

These steps do not affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

   ```
   psql -d vmdb_production -c "SELECT pg_size_pretty(pg_database_size('vmdb_production'));"
   ```

2. If it's a multi-region environment, ensure you removed all subscriptions from the global database appliance.

3. Stop the **evmserverd** process.

   ```
   systemctl stop evmserverd
   ```

4. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

   ```
   pg_dumpall -c --if-exists | gzip > vmdb.pg.gz
   ```

   Notes:

   - This command creates a compressed text-based backup. **pg_dumpall** is a utility for writing out ("dumping") all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
   - When you run a **pg_dumpall**, the **pg_replication_slots** is not copied over causing the slots to become inactive. Also, the size of the global database appliance is reduced allowing for quicker backup and restore.
   - Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your file system available space by using the df command.
   - You must also provide backup for configuration files such as **/var/www/miq/vmdb/certs/v2_key**, **/var/www/miq/vmdb/config/database.yml**, and **/var/www/miq/vmdb/GUID**.

5. Back up the following files and take note of the appliance that each file comes from. These files are used to replace the old appliances in the database and for disaster recovery.

   - /var/www/miq/vmdb/GUID

6. During the upgrade, the iptables configuration file **/etc/sysconfig/iptables** is removed. If you changed the iptables configuration from the default, run the **iptables --list -n** command to see the current configuration. Run the following command to back up the iptables configuration:

   ```
   iptables-save > /etc/iptables.conf
   ```

   You can restore your iptables configuration file with the following command:

   ```
   iptables-restore < /etc/iptables.conf
   ```

   You can also add this command to **/etc/rc.local** to reload the rules at every restart.

Steps to follow on newly deployed Infrastructure Management 4.8.1 Appliances

Complete the following steps on your appliances as specified to migrate to Infrastructure Management.

Note: Some steps are run on certain appliances. Ensure you wait for each command to finish before you move to the next step.

1. Provision and start the new appliances.
2. On the VMDB and non-VMDB appliances, run the following commands as the root user:
    1. Copy the GUID file from the appliance.

        ```
        scp root@$IP_of_old_appliance:/var/www/miq/vmdb/GUID /var/www/miq/vmdb/
        ```

        Notes:

        - Complete these steps on VMDB appliances first. The non-VMDB appliances don't have a database to connect to if they're done first.
        - Complete this step only if you want to keep the same appliance/miq_servers records in the database and inherit that appliance's existing settings. If you plan on deploying new appliances with new IP addresses and hostname and not inherit the settings, you might prefer creating new records by not copying the GUID.
    2. Stop the evmserverd process.

        ```
        systemctl stop evmserverd
        ```

    3. Remove the previously generated encryption key.

        ```
        rm /var/www/miq/vmdb/certs/v2_key
        ```

    4. Copy the default database configuration file, ovewriting the previously generated one that used the deleted encryption key.

        ```
        scp root@<IP_of_old_appliance>:/var/www/miq/vmdb/config/database.yml /var/www/miq/vmdb/config/database.yml
        ```

    5. Drop the previously generated database.

        ```
        dropdb vmdb_production
        ```

    6. For VMDB appliances, configure the application, fetch the key from remote system, and Reset Configured Database.
    7. For VMDB appliances, select the same region number and database password as the old installation. You can find the region number that is used for an installation that is listed as **Database/Region:** in the `appliance_console` summary information.
    8. For non-VMDB appliances, configure the application by using the join region in external database option. Note as part of the database configuration step, fetch the `v2_key` from the appliances.
    9. Allow the appliance to start.
3. On the new VMDB and new dedicated database appliances:
    1. Stop the evmserverd process.

        ```
        systemctl stop evmserverd
        ```

    2. Drop the default database.

        ```
        dropdb vmdb_production
        ```

    3. Copy the database dump from the appliance. Update the following location to reflect the source file location of the database dump and copy it to a location with enough available space.

        ```
        scp root@$IP_of_old_appliance:/var/tmp/vmdb.pg.gz /var/tmp
        ```

    4. Decompress and restore the database dump backup file.

        ```
        gunzip < vmdb.pg.gz | psql postgres
        ```

    5. Change to the vmdb directory.

        ```
        cd /var/www/miq/vmdb/
        ```

    6. Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you run into an error, contact IBM Support.

        ```
        rake db:migrate
        ```

4. Update the Automate Model to the latest version. This resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance.

    ```
    rake evm:automate:reset
    ```

5. Start the evmserverd process. In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance:

    ```
    systemctl start evmserverd
    ```

The migration process is now complete. Once appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrading Infrastructure Management appliances

The following steps describe the process of upgrading Infrastructure Management appliances when a new release of IBM Cloud Pak for AIOps Infrastructure Automation is available.

Important: You can upgrade by a maximum of one release at a time. If the release that you want to upgrade from is more than one release behind the release that you are upgrading to, then you must upgrade sequentially. For more information, see Upgrade paths.

Before you begin

- When you are upgrading, Kafka messaging configuration is required for the appliance setup. You can configure the broker on the same appliance where your database is configured. You can only have one Kafka broker per region.You can either configure the current appliance as a Kafka broker, or point the appliance to an existing external Kafka broker.

    1. Select between **Configure this appliance as a messaging server** or **Connect to an external messaging system** (to connect to an external Kafka broker). Fill in the necessary *Message Client Parameters* fields such as *IP address* and *username/password*.
    2. Click **Proceed** and the *appliance_console* will apply the configuration that you selected, and then restart *evmserverd* to apply the changes.

Notes:

- You can do a normal appliance rpm upgrade from 4.5.0 to 4.8.1.
- When you are upgrading to version 4.8.1 from a release that is older than 4.5.0, you must do a full appliance replacement. For more information, see Upgrade an Infrastructure Management appliance from 3.6.x.

About this task

Use the following steps to begin the appliance upgrade process.

Infrastructure Management has three types of appliances:

- VMDB appliance - An Infrastructure Management appliance with running workers, which also hosts its own database that other appliances can connect to.

  Upgrade an Infrastructure Management VMDB appliance

- Non-VMDB appliance - An Infrastructure Management appliance with running workers that does not host a database. The appliance is connected to an external database.

  Upgrade an Infrastructure Management non-VMDB appliance

- Dedicated database appliance - An Infrastructure Management appliance or non-Infrastructure Management virtual machine with no running workers on it. The appliance contains only a database for other appliances to connect to.

  Upgrade an Infrastructure Management dedicated database appliance

Upgrade an Infrastructure Management appliance from 3.6.x

The following steps describe the process of upgrading Infrastructure Management appliances from 3.6.x.

Important:

- Do not run **evmserverd** if a database is in a partially migrated state.
- As part of upgrading to version 4.8.1, you must do a full appliance replacement. For more information, see Upgrade an Infrastructure Management appliance from 3.6.x.

  The reason for this replacement is that Infrastructure Management 4.8.1 does not support a rpm upgrade when deployed as an appliance. Rpm upgrade instructions are only applicable for earlier versions.

Procedure

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Stop the **evmserverd** process.

   Skip this step in dedicated database appliances:

   ```
   systemctl stop evmserverd
   ```

3. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For more information about downloading the image, see Passport Advantage part numbers. Download the RPM updates image and copy the image into the temporary upgrade directory.

4. Extract the .tgz on the target appliance.

Run the following commands:

```
cd /tmp/upgrade
tar -zxvf infrastructure-management-upgrade-15.y*.tgz
```

5. Upgrade the RPM packages.

   Run the following commands:

   ```
   dnf -y upgrade --allowerasing --refresh *.rpm
   dnf -y autoremove
   ```

6. Migrate the database.

   Important: In a single (stand-alone database) or replicated environment, run the following command only on the first Non-VMDB appliance.

   Change to the vmdb directory.

   ```
   cd /var/www/miq/vmdb/
   ```

   Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you run into an error, contact IBM Support.

   ```
   rake db:migrate
   ```

7. Update the Automate Model to the latest version.

   This command resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. Run the command appropriate to your environment to update the Automate Model:

   Important: In a single (stand-alone database) or replicated environment, run the following command only on the first Non-VMDB appliance.

   ```
   rake evm:automate:reset
   ```

8. Start the evmserverd process.

   Skip this step in dedicated database appliances:

   ```
   systemctl start evmserverd
   ```

9. Remove the temporary upgrade directory.

   Remove the upgrade directory by running the following command:

   ```
   rm -rf /tmp/upgrade
   ```

   The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrade an Infrastructure Management VMDB appliance

Important: Do not run `evmserverd` if a database is in a partially migrated state.

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Back up the current appliance database, v2_key, database.yml, and GUID file.

   Important: Some of these substeps in step 2 are only required if you are upgrading from Infrastructure Management 3.5.x and older.

   These steps don't affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

   1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

      Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

      ```
      psql -d vmdb_production -c "SELECT pg_size_pretty(pg_database_size('vmdb_production'));"
      ```

   2. If it's a multi-region environment, make sure that you removed all subscriptions from the global database appliance.

      Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   3. Stop the `evmserverd` process.

      ```
      systemctl stop evmserverd
      ```

   4. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

      Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

```
pg_dumpall -c --if-exists | gzip > vmdb.pg.gz
```

Notes:

- If you changed your postgresql database configuration files, **pg_hba.conf**, and **postgresql.conf**, include these files when you back up the v2_key, database.yml, GUID files. These files need to be manually updated to include any changes you made previously.
- The **pg_dumpall** command creates a compressed text-based backup. **pg_dumpall** is a utility for writing out (“dumping”) all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
- Running a **pg_dumpall** command the **pg_replication_slots** doesn't get copied over causing the slots to become inactive. This action helps to reduce the size of the global database appliance and allows for quicker backup and restore.
- Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your available file system space by using the **df** command.
- You must also provide backup for configuration files such as **/var/www/miq/vmdb/certs/v2_key**, **/var/www/miq/vmdb/config/database.yml**, and **/var/www/miq/vmdb/GUID**.

5. Back up the following files and make a note of the appliance that each file comes from. This step is to replace the old appliances in the database and for disaster recovery.

   Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   Note: The REGION file might not exist.

   - **/var/www/miq/vmdb/GUID**
   - **/var/www/miq/vmdb/REGION**

3. Clear the data directory.

   Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   After the database and postgresql are safely backed up, clear the data directory by running the following commands:

   ```
   systemctl stop postgresql
   rm -rf $APPLIANCE_PG_DATA
   ```

4. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For more information about downloading the image, see Passport Advantage part numbers. Download the RPM updates image and copy the image into the temporary upgrade directory.

5. Create the **/root/upgrade** directory by running the following commands:

   ```
   rm -fr /root/upgrade
   mkdir /root/upgrade
   ```

6. Use the **scp** command to copy the rpm tarball to the appliance server.

   ```
   scp ~/Downloads/infrastructure-management-upgrade-18.y-x86_64-20240521161926.tgz
   root@<appliance_hostname_or_ip>:/root/upgrade
   ```

   Where **<appliance_hostname_or_ip>** is the IP address of the appliance or the hostname of the appliance.

   For example, the scp command copies the **infrastructure-management-upgrade-18.y-x86_64-20240521161926.tgz** rpm tarball to the server using **root** as the username.

7. SSH to the appliance or use the VM console to extract the tarball:

   ```
   tar -zxvf /root/upgrade/infrastructure-management-upgrade-*.tgz
   ```

8. Run the upgrade script:

   ```
   /root/upgrade/im-upgrade.sh
   ```

9. Remove the temporary upgrade directory:

   ```
   rm -rf /tmp/upgrade
   ```

   The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrade an Infrastructure Management Non-VMDB appliance

Important: Do not run **evmserverd** if a database is in a partially migrated state.

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Stop the evmserverd process.

   Run the following command:

   ```
   systemctl stop evmserverd
   ```

3. Back up important files.

   Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   Back up the following files and make a note of the appliance that each file comes from. This step is to replace the old appliances in the database and for disaster recovery.

   Note: The REGION file might not exist.

   - /var/www/miq/vmdb/GUID
   - /var/www/miq/vmdb/REGION
4. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For information on downloading the image, see Passport Advantage part numbers. Download the RPM updates image and copy the image into the temporary upgrade directory.

5. Create the **/root/upgrade** directory by running the following commands:

   ```
   rm -fr /root/upgrade
   mkdir /root/upgrade
   ```

6. Use the **scp** command to copy the rpm tarball to the appliance server.

   ```
   scp ~/Downloads/infrastructure-management-upgrade-18.y-x86_64-20240521161926.tgz
   root@<appliance_hostname_or_ip>:/root/upgrade
   ```

   Where **<appliance_hostname_or_ip>** is the IP address of the appliance or the hostname of the appliance.

   For example, the scp command copies the **infrastructure-management-upgrade-18.y-x86_64-20240521161926.tgz** rpm tarball to the server using **root** as the username.

7. SSH to the appliance or use the VM console to extract the tarball:

   ```
   tar -zxvf /root/upgrade/infrastructure-management-upgrade-*.tgz
   ```

8. Run the upgrade script:

   ```
   /root/upgrade/im-upgrade.sh
   ```

9. Remove the temporary upgrade directory:

   ```
   rm -rf /tmp/upgrade
   ```

   The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrade an Infrastructure Management Dedicated database appliance

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Back up the current appliance database.

   Important: All of these substeps in step 2 are only required if you are upgrading from Infrastructure Management 3.5.x and older.

   These steps don't affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

   1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

      Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

      ```
      psql -d vmdb_production -c "SELECT pg_size_pretty(pg_database_size('vmdb_production'));"
      ```

   2. If it's a multi-region environment, make sure that you removed all subscriptions from the global database appliance.

      Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   3. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

      Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

      ```
      pg_dumpall -c --if-exists | gzip > vmdb.pg.gz
      ```

      Notes:

      - If you changed your postgresql database configuration files, such as the pg_hba.conf and postgresql.conf then update these files manually to include any changes you made previously.

- The `pg_dumpall` command creates a compressed text-based backup. `pg_dumpall` is a utility for writing out (“dumping”) all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
- Running a `pg_dumpall` command the `pg_replication_slots` doesn't get copied over causing the slots to become inactive. This action helps to reduce the size of the global database appliance and allows for quicker backup and restore.
- Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your file system available space by using the `df` command.

3. Clear the data directory.

   Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   After the database and postgresql are safely backed up, clear the data directory by running the following commands:

   ```
   systemctl stop postgresql
   rm -rf $APPLIANCE_PG_DATA
   ```

4. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For more information, see Passport Advantage part numbers. Download the IBM Cloud Pak for AIOps 4.8.1 - Infrastructure Automation RPM updates image.

5. Extract the .tgz on the target appliance.

   Run the following commands:

   ```
   cd /tmp/upgrade
   tar -zxvf infrastructure-management-upgrade-15.y*.tgz
   ```

6. Enable new stream 8 modules.

   Important: This step is only required if you are upgrading from Infrastructure Management 3.5.x and older.

   Enable the stream 8 modules by running the following commands:

   ```
   dnf -y module reset postgresql
   dnf -y module enable postgresql:13
   dnf -y module reset ruby
   dnf -y module enable ruby:3.0
   ```

7. Upgrade the RPM packages.

   Run the following commands:

   ```
   dnf -y upgrade --allowerasing --refresh *.rpm
   dnf -y autoremove
   ```

8. Verify that the PostgreSQL version is 13.

   Important: This step is required only if you are upgrading from Infrastructure Management 3.5.x and older.

   Run the following command:

   ```
   cat $APPLIANCE_PG_DATA/PG_VERSION
   ```

   This command checks the PostgreSQL version and returns: 13

9. Complete the steps in Upgrading Non-VMDB appliances.

   Complete the steps in Upgrade an Infrastructure Management non-VMDB appliance

10. Remove the temporary upgrade directory.

    Remove the upgrade directory by running the following command:

    ```
    rm -rf /tmp/upgrade
    ```

    The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Migrating from an Infrastructure Management appliance to a Kubernetes (podified) installation

Follow these steps to migrate from an appliance installation to a Kubernetes (podified) installation of Infrastructure Management.

Before you begin

You must have an Infrastructure Management appliance that is installed and a containerized (podified) installation of Infrastructure Management.

Collect data from the Infrastructure Management appliance

1. Take a backup of the database from the appliance.

```
pg_dump -Fc -d <database_name> > /root/pg_dump
```

Where **<database_name>** is **vmdb_production** by default. Find your database name by running **appliance_console** on your appliance.

2. Export the encryption key and Base64 encode it for the Kubernetes Secret.

```
vmdb && rails r "puts Base64.encode64(ManageIQ::Password.key.to_s)"
```

3. Get the region number.

```
vmdb && rails r "puts MiqRegion.my_region.region"
```

4. Get the GUID of the server that you want to run as.

```
vmdb && cat GUID
```

Restore the backup into the Kubernetes environment

Important:

Two options are available. You can restore the backup of the appliance into a new installation or you can restore the backup into an existing installation.

- Restore the appliance backup into a new Kubernetes based installation
- Restore the appliance backup into an existing Kubernetes based installation

Restore the backup into a new Kubernetes based installation

For a new installation, you need to make sure you configured OIDC for integration with IBM Cloud Paks®. You must do so before you create the yaml file and continue with this procedure. Follow the steps in OIDC Configuration Steps for Infrastructure Management.

1. Create a YAML file to define the Custom Resource (CR). Minimally you need the following.

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  name: im-iminstall
spec:
  applicationDomain: inframgmtinstall.apps.mycluster.com
  databaseRegion: <region number from the appliance>
  serverGuid: <GUID value from appliance>
```

Example CR

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  labels:
    app.kubernetes.io/instance: ibm-infra-management-install-operator
    app.kubernetes.io/managed-by: ibm-infra-management-install-operator
    app.kubernetes.io/name: ibm-infra-management-install-operator
  name: im-iminstall
  namespace: c4waiops
spec:
  applicationDomain: inframgmtinstall.apps.mycluster.com
  databaseRegion: '99'
  serverGuid: <GUID value from appliance above>
  imagePullSecret: my-pull-secret
  enableSSO: true
  initialAdminGroupName: inframgmtusers
  license:
    accept: true
```

2. Create the CR in your namespace. The operator creates several more resources and starts deploying the app.

```
oc create -f <CR file name>
```

3. Edit the secret and insert the encryption key from the appliance. Replace the "encryption-key" value with the value exported from the appliance.

```
oc edit secret app-secrets -n c4waiops
```

4. Temporarily hijack the orchestrator by adding the following to the deployment:

```
oc patch deployment orchestrator -p '{"spec":{"template":{"spec":{"containers":[{"name":"orchestrator","command":
["sleep", "1d"]}]}}}}' -n cp4aiops
```

Note:

The default timeout for **oc debug** is 1 minute of inactivity. Increase the idle timeout values for the client and server. Otherwise, you might have difficulty in completing the remaining steps without the debug pod timing out and forcing a restart of the entire process. For more information, see How to change idle-timeout for oc commands like oc rsh, oc logs, and oc port-forward in OCP4.

5. Shell into the orchestrator container:

```
oc rsh deploy/orchestrator
```

Clear postgres database:

```
cd /var/www/miq/vmdb
source ./container_env
DISABLE_DATABASE_ENVIRONMENT_CHECK=1 rake db:drop db:create
```

6. Copy your appliance database file into the postgresql pod. You must run the command from your cluster vm, not from within the pod.

```
oc cp <your_backup_file> <postgresql pod>:/var/lib/pgsql
```

7. Run the command `oc rsh` into the database pod and restore the database backup.

```
cd /var/lib/pgsql
# --- you copied the backup here in step 6. ---
pg_restore -v -d vmdb_production <your_backup_file>
rm -f <your_backup_file>
exit
```

8. Back in the debug pod from step 5:

```
rake db:migrate
exit
```

9. Delete the orchestrator deployment to remove the command override that was previously added:

```
oc delete deployment orchestrator -n cp4aiops
```

The orchestrator starts deploying the rest of the pods that are required to run Infrastructure Management.

Restore the backup into an existing Kubernetes based installation

1. For an existing installation, find the existing CR by running the command,

```
oc get IMInstall -n cp4aiops
```

2. Edit the `IMInstall` resource from the CLI with the details in the previous step, for example, `oc edit IMInstall im-install` and add the following lines under `spec:`

```
  databaseRegion: <region number from the appliance>
  serverGuid: <GUID value from appliance>
```

OR

Log in to Red Hat OpenShift Container Platform for your cluster. Click **Home > Search**. In the **Resources** list menu, find **IMInstall**. Click the `IMInstall` resource; then, click the **YAML** tab. Add the following lines under `spec:`

```
  databaseRegion: <region number from the appliance>
  serverGuid: <GUID value from appliance>
```

Click **Save**

3. Edit the secret and insert the encryption key from the appliance. Replace the "encryption-key" value with the value exported from the appliance.

```
oc edit secret app-secrets -n cp4aiops
```

4. Temporarily hijack the orchestrator by adding the following to the deployment:

```
oc patch deployment orchestrator -p '{"spec":{"template":{"spec":{"containers":[{"name":"orchestrator","command":
["sleep", "1d"]}]}}}}' -n cp4aiops
```

Note: The default timeout for `oc debug` is 1 minute of inactivity. Increase the idle timeout values for the client and server. Otherwise, you might have difficulty in completing the remaining steps without the debug pod timing out and forcing a restart of the entire process. For more information, see How to change idle-timeout for oc commands like oc rsh, oc logs, and oc port-forward in OCP4.

5. Shell into the orchestrator container:

```
oc rsh deploy/orchestrator
```

Clear postgres database:

```
cd /var/www/miq/vmdb
source ./container_env
DISABLE_DATABASE_ENVIRONMENT_CHECK=1 rake db:drop db:create
```

6. Copy your appliance database file into the postgresql pod. You must run the command from your cluster vm, not from within the pod.

```
oc cp <your_backup_file> <postgresql pod>:/var/lib/pgsql
```

7. Run the command `oc rsh` into the database pod and restore the database backup.

```
cd /var/lib/pgsql
# --- you copied the backup here in step 6. ---
pg_restore -v -d vmdb_production <your_backup_file>
rm -f <your_backup_file>
exit
```

8. Back in the debug pod from step 5:

```
rake db:migrate
exit
```

9. Delete the orchestrator deployment to remove the command override that was previously added:

```
oc delete deployment orchestrator -n cp4aiops
```

Important:

If your Infrastructure Management instance on your cluster was previously configured for OIDC/SSO, you need to complete these additional steps.

1. Log in to IBM Cloud Paks®, and from the menu, click **Automate infrastructure > Infrastructure Management**. SSO to Infrastructure Management is expected to fail.
2. On the Infrastructure Management login screen, log in with the default credentials, user **admin**, and password **smartvm**.
3. In Infrastructure Management, click **Settings > Application Settings**. Expand **Access Control** and click **Groups**.
4. Click **Configuration > Add a new Group**, and create a group with the following fields:
   - *Description:* **<LDAP_group_name>**
   - *Role:* **EvmRole-super-administrator**
   - *Project/Tenant:* **<Your_tenant>**

   Where **<LDAP_group_name>** is the value for **initialAdminGroupName:** found in the installation CR.
5. When finished, click **Add**.
6. Expand **Settings** and click **Zones > Server: EVM**. Click the **Authentication** tab, and ensure that the following settings are displayed:
   - *Mode*: External (**httpd**)
   - *Enable Single Sign-On*: checked
   - *Provider Type*: Enable OpenID-Connect
   - *Get User Groups from External Authentication (httpd)*: checked
7. Log out of Infrastructure Management and IBM Cloud Paks®. Log in to IBM Cloud Paks® by using **Enterprise LDAP** authentication. From the menu, click **Automate infrastructure > Infrastructure Management**. SSO to Infrastructure Management can now succeed.

Customizing the installation

Configuring the application domain name

It is necessary to set the **applicationDomain** in the CR if you are running in a production cluster. For a Code Ready Containers cluster, it might be something like **miqproject.apps-crc.testing**

External postgresql

Running with an external Postgres server is an option, if you want the default internal Postgres you can skip this step. Additionally, if you want to secure the connection, you need to include the optional parameters **sslmode=verify-full** and **rootcertificate** when you create the secret. To complete this process, manually create the secret and substitute your values before you create the CR.

```
oc create secret generic postgresql-secrets \
  --from-literal=dbname=vmdb_production \
  --from-literal=hostname=<your hostname> \
  --from-literal=port=5432 \
  --from-literal=password=<your password> \
  --from-literal=username=<your username> \
  --from-literal=sslmode=verify-full \ # optional
  --from-file=rootcertificate=path/to/optional/certificate.pem # optional
```

Note: If wanted, the secret name is also customizable by setting **databaseSecret** in the CR.

Using TLS to encrypt connections between pods inside the cluster:

Create a secret that contains all of the certificates

Certificates must be signed by a CA and that CA certificate must be in uploaded as **root_crt** so that it can be used to verify connection validity. If the secret is named **internal-certificates-secret**, no changes are needed in the CR, if you choose a different name that must be set in the **internalCertificatesSecret** field of the CR.

```
    oc create secret generic internal-certificates-secret \
      --from-file=root_crt=./certs/root.crt \
      --from-file=httpd_crt=./certs/httpd.crt \
      --from-file=httpd_key=./certs/httpd.key \
      --from-file=kafka_crt=./certs/kafka.crt \
      --from-file=kafka_key=./certs/kafka.key \
      --from-file=memcached_crt=./certs/memcached.crt \
      --from-file=memcached_key=./certs/memcached.key \
      --from-file=postgresql_crt=./certs/postgresql.crt \
      --from-file=postgresql_key=./certs/postgresql.key \
      --from-file=api_crt=./certs/api.crt \
      --from-file=api_key=./certs/api.key \
      --from-file=remote-console_crt=./certs/remote-console.crt \
      --from-file=remote-console_key=./certs/remote-console.key \
      --from-file=ui_crt=./certs/ui.crt \
      --from-file=ui_key=./certs/ui.key
```

Generating certificates:

A certificate authority (CA) and certificates are needed for each service. These certificates need to be valid for the internal Kubernetes service name, such as **httpd** or **postgres**. The services (UI, api) that back the route require the certificate to include a SAN with the application domain name. For example, the certificate for the

UI needs to be valid for the hostname `ui` and `your_application.apps.example.com`.

Configuring external messaging

It is possible to run with an external Kafka messaging server. To do so, create the required secret with the correct parameters by using the following example and provide that secret name as `kafkaSecret` in the CR.

```
kubectl create secret generic kafka-secrets --from-literal=hostname=<your fqdn> --from-literal=username=<your username> --from-literal=password=<your password>
```

If you decide to use a secret name other than `kafka-secrets`, you need to specify that in the CR.

```
kafkaSecret: <name of your secret>
```

Creating a custom TLS secret

You can use a custom TLS certificate. To use a custom certificate create the required secret with the correct parameters by using the folowing example and provide that secret name as `tlsSecret` in the CR.

```
oc create secret tls tls-secret --cert=tls.crt --key=tls.key`
```

If you decide to use a secret name other than `tls-secret`, you need to specify that in the CR.

```
tlsSecret: <name of your tls secret>
```

Configuring an image pull secret

If authentication is required to pull the images, a secret that contains the credentials needs to be created. The following script is an example of creating the secret.

```
kubectl create secret docker-registry image-pull-secret --docker-username=<your registry username> --docker-password=<your registry password> --docker-server=<your registry server>
```

If you decide to use a secret name other than `image-pull-secret`, you need to specify that in the CR.

```
imagePullSecret: <name of your pull secret>
```

Configuring OpenID-Connect Authentication

To run with OpenID-Connect Authentication, customize the following example as required to fit your environment. For the following example, we tested with Keycloak version 11.0

Create a secret that contains the OpenID-Connect's `Client ID` and `Client Secret`. The values for `CLIENT_ID` and `CLIENT_SECRET` come from your authentication provider's client definition.

```
oc create secret generic <name of your kubernetes secret> --from-literal=CLIENT_ID=<your auth provider client ID> --from-literal=CLIENT_SECRET=<your auth provider client secret>
```

Modify the CR with the following values:

```
httpdAuthenticationType: openid-connect
oidcProviderURL: https://<your keycloak FQDN>/auth/realms/<your realm>/.well-known/openid-configuration
oidcClientSecret: <name of your kubernetes secret>
```

Configuring OpenID-Connect with a CA Certificate

To configure OpenID-Connect with a CA certificate, follow these steps:

Acquire your CA certificate (Review the documentation for your certificate provider for instructions) and store it in a secret by using the following example.

```
oc create secret generic <name of your kubernetes OIDC CA cert> --from-file=<path to your OIDC CA cert file>
```

Modify the CR to identify the secret created.

```
oidcCaCertSecret: <name of your kubernetes OIDC CA cert>
```

Using your own images

If you built your own custom application images and want to deploy them, you can specify the image names in the CR by using the following example.

```
applicationDomain: miqproject.apps-crc.testing
orchestratorImage: docker.io/<your_username_or_organization>/<your_prefix>-orchestrator:latest
baseWorkerImage: docker.io/<your_username_or_organization>/<your_prefix>-base-worker:latest
uiWorkerImage: docker.io/<your_username_or_organization>/<your_prefix>-ui-worker:latest
webserverWorkerImage: docker.io/<your_username_or_organization>/<your_prefix>-webserver-worker:latest
```

If you built your own operator image, you need to edit the operator deployment to specify the image.

Infrastructure Automation - Administration

- User interface
- Roles and permissions
- Backup and restore
  - Installing the backup and restore tools
  - Backing up Infrastructure Automation

    - Restoring Infrastructure Automation
    - Upgrading Infrastructure Automation backup and restore artifacts
    - Backup and restore Infrastructure management appliances
- Migrating to IBM Cloud Pak for AIOps
    - Migrating IBM Cloud Private Managed services
    - Migrating IBM Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances

Managed services UI view

Use the Infrastructure Automation UI console to view, create, and automate the management of your infrastructure.

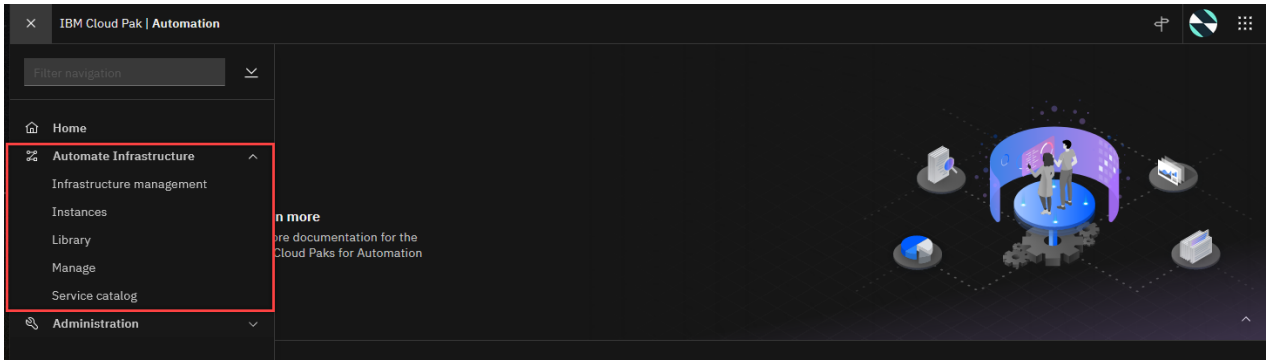


Figure. Main page

Instances

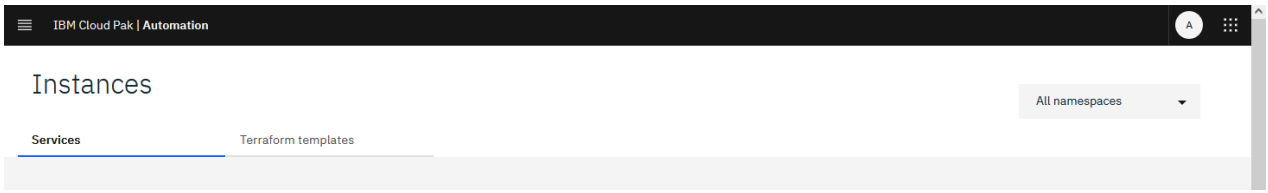


Figure. Instances

Services

On the *Services* tab, you can view service instances. If you need to create a service instance, you are redirected to the **Library** > **Services** page, where you can create the service instance.

For more information, see Services.

Terraform templates

On the *Services* tab, you can view Terraform instances. If you need to create a Terraform instance, you are redirected to the **Library** > **Terraform templates** page, where you can create the Terraform instance.

For more information, see Templates.

Library

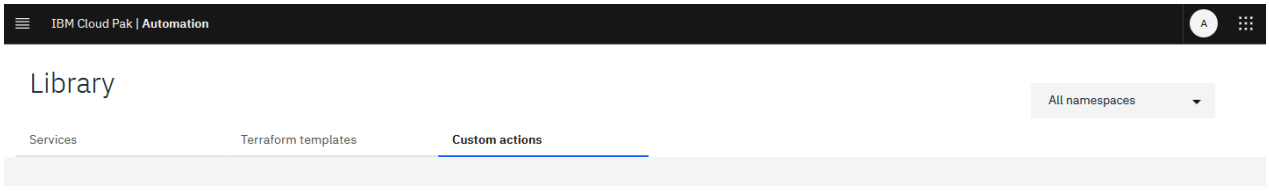


Figure. Library page

Services

On the *Services* tab, you can view all services, which can be sorted by category. You can also choose to create or import services.

For more information, see Services.

Terraform templates

On the *Terraform templates* tab, you can view all templates. You can also choose to import a template.

For more information, see Templates.

Custom actions

On the *Custom actions* tab, you can view all custom actions, which can be sorted by category. You can also choose to create different custom actions.

For more information, see Creating and managing custom actions.

Manage

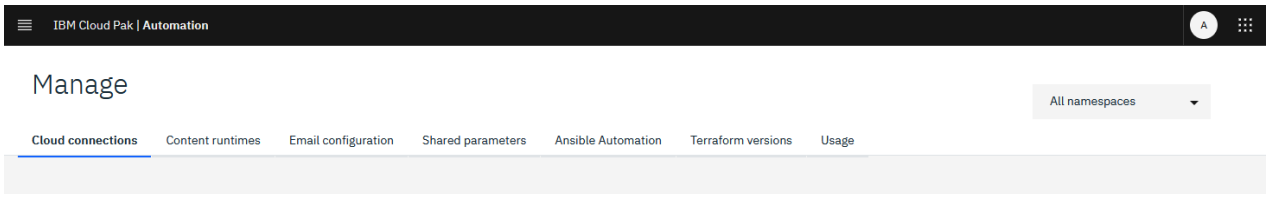


Figure. Manage page

Cloud connections

On the *Cloud connections* tab, you can view and create cloud connections. Cloud connections allow you to create connections to each of your target cloud providers that are used when you deploy instances.

For more information, see Managing connections

Content runtimes

On the *Content runtimes* tab, you can view and create advanced content runtimes.

For more information about creating advanced content runtimes, see Managing content runtime

Email configuration

On the *Email configuration* tab, you can view and create email server configurations. To use email-related features within Manage services, you need to configure an email server to send these emails.

For more information about configuring and managing SMTP server connection for email notifications, see Managing Email Configuration

Shared parameters

On the *Shared parameters* tab, you can view and create data types and data objects that can be used as shared parameters. Shared parameters define shared configuration to store data and are used by templates. You can create reusable data objects using shared parameters. This helps save time when you are deploying new instances and creating services.

For more information, see Shared parameters.

Ansible Automation

On the *Ansible Automation* tab, you can create and view Ansible connections. By creating a globally accessible connection to Ansible Tower, you can include Ansible Automation within your services.

For more information about configuring an Ansible Tower connection, see Configuring the Ansible Tower connection.

Terraform versions

On the *Terraform versions* tab, you can view versions of Terraform that can be offered when you deploy templates and services. You can also create versions, if needed. A default version can be set, which is used when the template metadata does not specify a version.

For more information, see Managing Terraform Versions.

Usage

On the *Usage* tab, you can view the highest number of virtual machines (VMs) managed by Manage services each month. You can view the current and monthly usage, and download the usage data.

For more information, see Managing the usage of virtual machines.

Access Control

In the *Access Control* tab, you can provide access for users to certain namespaces. An administrator user can assign namespaces to the user or user groups by using this feature.

For more information, see Managing Access Control.

Service catalog

The Service catalog view displays the deployable services and applications and related details. It also displays the deployed instances and related details. With the service catalog you can complete the following tasks:

- As an IT administrator, you can define a new service by using the service composer, test it and when it is ready, publish it to the Service catalog. You can decide which parameters to expose to users, such as to hide the complexity around provisioning from users.
- As a user, such as a developer, you can access the Service catalog to provision infrastructure and applications.

The Service catalog view can be refreshed at a particular interval by using the **Autorefresh** menu. The default is 10 seconds. You can also disable the autorefresh.

To access the Service catalog, go to IBM Cloud Pak for AIOps console and click **Automate Infrastructure** > **Service catalog**.

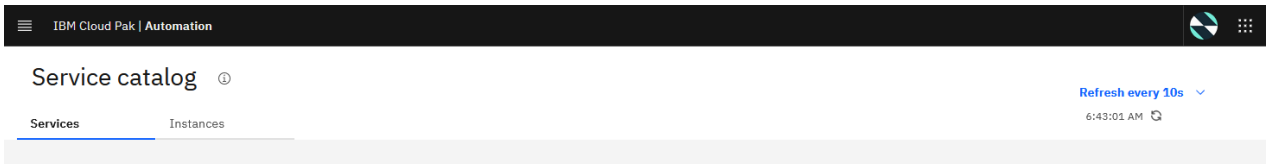


Figure. Service catalog page

- Services
- Instances

Services

On the *Services* tab, a tiled list of published services and applications is displayed. The list of services and applications is sorted according to categories. Clicking the service deploys that service to your environment. You can search a service by using **Search services** option or search by category.

For more information about services, see the following topics:

- Creating a service
- Publishing a service
- Deploying a service

Instances

On the *Instances* tab, a list of all deployed service and application instances is displayed. If needed, you can use the search box to filter the listed instances.

For each instance, the following information is displayed:

- The name of the instance
- The name of the service
- The status of the instance, Executed or Terminated
- The date and time when the service was deployed

A typical operation user can access all the following instance actions from the overflow menu:

- **View**: For more information, see Viewing service instance details.
- **Edit**: For more information, see Editing a service instance.
- **Custom actions**: For more information, see Custom actions.
- **Retry/resume** : For more information, see Resuming or retrying a service instance.
  - Applicable only for **Error** state instances.
  - **Retry** tries to deploy the instance from the beginning of the execution.
  - **Resume** tries to resume deploying the instance from the point of failure.
- **Upgrade/downgrade**: For more information, see Upgrading or downgrading a service instance
- **Terminate**/**Delete**: For more information, see Terminating and deleting a service instance.
  - Only **Active** instances can be terminated.
  - Only **Terminated** instances can be deleted.

Roles and permissions for Infrastructure Automation

For accessing and completing actions with IBM Cloud Pak® for AIOps Infrastructure Automation, users require specific roles and permissions. Review the different available roles, permissions, and the associated actions that users can complete with the roles and permissions. These roles and permissions help you get your users set up to begin day to day operations.

- Predefined roles
- Creating a role
- Permissions

Predefined roles

A role defines the permissions that a user or group has. You can edit the default roles or create new roles if the default set of permissions in a role does not align with your business needs. For more information, see Managing roles.

The following roles are available by default:

- Administrator
- Automation Administrator
- Automation Analyst
- Automation Developer (AIOps Developer role)

- Automation Operator
- User

Notes:

- The User predefined role is listed as available by default but this role does not include any permissions for completing actions within IBM Cloud Pak for AIOps. This default role is used within the IBM® Automation family of offerings, which includes IBM Cloud Pak for AIOps, however, this roles is not used within IBM Cloud Pak for AIOps.

  As this role does not include or provide any permissions within IBM Cloud Pak for AIOps, do not assign the role to users within IBM Cloud Pak for AIOps. If more IBM Cloud Paks are installed, this roles can have permissions for completing actions within tools for those other IBM Cloud Paks.

- Not all permissions are associated with a predefined role. A user with with a permission to manage roles, such as the Administer platform permission, can assign permissions to a role when needed.

Default user

The default user (`admin`) is automatically assigned the following roles when the roles are added to the platform:

- Administrator

Creating a role

When adding a user, you can choose to create a role instead of assigning an existing role. You can also choose to directly create a role. The options for creating a role are the same regardless of the path you select to create the role.

To directly create a role, complete the following steps:

1. Log in to the console as an administrator with permissions to manage users.
2. From the main navigation, click **Administration** > **Access control**.
3. Select the **Roles** tab and click **New role**.
4. Enter the details for the new role:
   - Name - Enter a distinctive and descriptive name for the role.
   - Description - Enter an optional description for the role to define the role, such as to identify the objective or responsibilities for users with the role.
5. Click **Next**.
6. Select the permissions for the role. You can use the *Find permissions* search field to find and filter the list of permissions.
7. Click **Next**.
8. Review the details for the role and click **Create**.

Permissions

The following table describes the actions that are associated with each permission.

Infrastructure Automation

Table: User administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Automation Analyst	Automation Administrator	Administrator
View Kubernetes resources	(1)			X		
Edit Kubernetes resources	(2)	X				
Manage Kubernetes resources	(3)		X			
Administer Kubernetes resources	(4)				X	
Administer Kubernetes namespace	(5)					X

1. View Kubernetes resources in the Cloud Pak namespaces.
   - View resources
2. Edit Kubernetes resources in the Cloud Pak namespaces.
   - View resources
   - Edit resources
3. Manage Kubernetes resources in the Cloud Pak namespaces.
   - View resources
   - Edit resources
   - Create resources
4. Administer Kubernetes resources in the Cloud Pak namespaces.
   - View resources

- Edit resources
- Create resources
- Delete resources

5. Administer Kubernetes namespace

    - Administer Kubernetes namespace (administrator ClusterRole)
    - Access Administration Hub
    - Manage identity providers
    - Manage teams
    - Manage service IDs

Platform administration

Permissions in this category enable an administrator to configure, customize, monitor, and manage the platform.

Only the Administer platform permission is associated with a predefined role (Adminstrator). This permission includes all other platform administration permissions.

Table: Platform administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Automation Administrator	Administrator
Administer platform	(1)				X
Manage configurations	(2)				
Manage platform health	(3)				
View platform health	(4)				

1. Administer platform

    This permission offers the most comprehensive set of actions for managing and monitoring the platform. Users with this permission have elevated privileges and can grant or revoke all permissions, including other administrative permissions. Users with this permission can complete the same actions as users that have the following permissions:

    - Manage configurations (Administer platform)
    - Manage platform health (Administer platform)
    - Manage platform roles (User administration)
    - Manage users (User administration)
    - Manage user groups (User administration)
    - Manage service instances (Service instances)

2. Manage configurations

    Users with this permission can customize the platform, integrate the platform with other applications, and enable connections to unsupported data sources. Users with this permission can access the Customizations page, the Configurations page, and the JDBC drivers tab on the Platform connections page. Some actions require specific services to be installed. Users with this permission can complete the following actions:

    - Configure connection to SMTP server

      Note: Requires Watson Studio or Watson Knowledge Catalog. An SMTP connection enables the platform to send emails.

    - Configure integration with IBM Guardium appliances

      Note: Requires Watson Knowledge Catalog. Use IBM Guardium to audit access to sensitive data on remote databases.

    - Configure connections to Hadoop clusters

      Note: Requires Execution Engine for Apache Hadoop.

    - Customize branding
    - Enable and disable home page cards
    - Enable and disable default support links
    - Add and delete custom support links
    - Enable and disable guided tours
    - Import JDBC drivers

      Note: Requires the common core services. JDBC drivers enable users to connect to unsupported data sources.

3. Manage platform health

    Users with this permission can monitor resource use, set quotas and alerts, manage workloads to maintain the health of the platform, and gather diagnostic data when problems occur. Users with this permission can access the Monitoring page and the Diagnostics page. Users with this permission can complete the following actions:

    - Monitor workloads and resource use

      Note: Workloads include active runtimes and jobs. Resources include memory and vCPU use across services, instances, and environments.

    - Stop any runtime environment

      Note: Environments are hardware and software configurations defined for running analytical assets or jobs.

- View pod status, details, and logs

  Note: Services are composed of Kubernetes pods. A pod is an instance of a process that runs on the cluster.

- Restart pods
- View platform quotas and service quotas

  Note: A quota specifies the maximum amount of memory or vCPU that the platform or a service should use.

- View event history and alerts
- Set and edit platform resource quotas
- Set and edit individual service resource quotas
- Create and run diagnostics jobs

  Note: A diagnostic job gathers information that can be used for troubleshooting problems.

- Delete diagnostics jobs

4. View platform health

   Users with this permission can monitor resource use and workloads across the platform to gauge the health of the platform. Users with this permission have read-only access to the Monitoring page. Users with this permission can complete the following actions:

   - Monitor workloads and resource use

     Note: Workloads include active runtimes and jobs. Resources include memory and vCPU use across services, instances, and environments.

   - View pod status, details, and logs

     Note: Services are composed of Kubernetes pods. A pod is an instance of a process that runs on the cluster.

   - View platform quotas and service quotas

     Note: A quota specifies the maximum amount of memory or vCPU that the platform or a service should use.

   - View event history and alerts

Service instances

A service instance is a specific deployment of a service. Some services can be deployed more than once. Some service instances have their own access controls.

Table: Service instances permissions

Permissions	Actions	Automation Developer	Automation Operator	Automation Administrator	Administrator
Create service instances	(1)				X
Manage service instances	(2)				

1. Create service instances

   Users with this permission can create service instances and storage volumes. The types of service instances depend on the services that are installed. Users with this permission can complete the following actions:

   - Create service instances

     Note: If an instance has access controls, the user who creates the instance is an instance administrator.

2. Manage service instances

   Users with this permission can manage access to any service instance or delete any service instance from the Instances page. Users with this permission can complete the following actions:

   - Create service instances
   - View all service instances
   - Add users to any service instance
   - Assign an instance role to instance users
   - Remove users from a service instance
   - Delete any service instance

What permissions do I have?

You can see what permissions you have from your profile. Your permissions are determined by the roles that are assigned to you.

To see what roles are assigned to you:

1. Log in to the IBM Cloud Pak for AIOps console and click your avatar in the upper right of the toolbar.
2. Click **Profile and settings**.
3. Click the **Roles** tab.

The permissions that are associated with your role (or roles) are listed in the **Enabled permissions** column.

Adding users

To add users, you need to have an external LDAP server set up for creating user profiles and authentication. When a user record is included within the LDAP server, you can add the user account within the IBM Cloud Pak for AIOps console. You can add a user directly or as part of an LDAP group. For more information, see Adding users.

Assigning roles and permissions

When you add a user or group, you must specify the role and permissions that they have. These roles and permissions can be granted to users through a combination of the IBM Cloud Pak for AIOps console and an external LDAP server for managing user authentication.

Red Hat OpenShift Container Platform roles

To add and manage users and roles for accessing and using the Red Hat OpenShift Container Platform web console, you need to use the Red Hat role-based access control (RBAC). For more information, see the Red Hat OpenShift topic Using RBAC to define and apply permissions.

Infrastructure Automation backup and restore

Learn how to back up and restore IBM Cloud Pak® for AIOps Infrastructure Automation, such as for recovering from a disaster like a complete data center outage.

- Overview
- Planning
- Backup process
- Restore process
- Components

Overview

Infrastructure Automation backup and restore is based on the storage and database backup of critical datastores, cluster resources, and component data and settings. With the backup and restore feature, you can recover your services to a previous point in time backup to recover from data corruption, system failures, or user errors.

You can also use the backup and restore feature as a means of copying data from one cluster to another, or from one environment into another, such as for disaster recovery purposes.

If you are using backup data to restore environments on a new cluster in another data center, your recovery time objective (RTO) and recovery point objective (RPO) must be sufficient for this usage.

Infrastructure Automation uses the *OpenShift APIs for Data Protection* (OADP) and the open source tool Velero to backup data to object storage. In the case of a failure, the APIs and Velero are used to restore data from the backups.

For more information about these tools, see:

- OpenShift APIs for Data Protection (OADP) features
- Velero documentation

Planning for backup and restore

When you are planning your Infrastructure Automation backup and restore strategy, consider what hardware requirements are needed for backing up your environment:

- You need to install the backup and restore tools in the Red Hat OpenShift Container Platform cluster. These tools include the OpenShift APIs for Data Protection (OADP), Velero, and Restic. You need to configure the OpenShift APIs and Velero with the appropriate object storage for storing your backups. This object storage must support RWX (read-write-execute) mode.
- You can backup both smaller starter and larger production sized cluster deployments.
- You can use backed up data to restore data to the existing cluster where the data was backed up, or to a new cluster.
- A restored cluster should have the same size and high availability level as the original cluster.
- Data is copied and stored in object Storage as part of the backup process. During a restoration, the same data is fetched and restored.

Notes:

- The backup and restore processes can take time. Using a backup to restore data or configurations is most useful when your recover time objective (RTO) and recovery point objectives (RPO) are met.
- As your data grows, the size of your backup storage might need to grow.
- A backup and restore of Red Hat OpenShift Container Platform or ETCD is not completed as part of the Infrastructure Automation backup and restore feature.
- Configure the duration between your backups to be less than your RPO. By default, the RPO is 12 hours, which requires backups to run once every 12 hours. The default RTO is 4 hours.
- Custom configuration settings for the Infrastructure Automation - Managed services component, such as Ansible replica count, extra variables, might not be backed up and restored. If you need this data and resources to be included in restored clusters, you need to directly add the data or resources to the restored cluster.
- If you upgrade your deployment, you must setup back and restore again. For more information, see Upgrading Infrastructure Automation backup and restore artifacts.

Backup process

The IBM Cloud Pak® for AIOps backup process is capable of backing up data for both Infrastructure Automation and IBM Cloud Pak for AIOps. The backup of data is completed by backing up the datastores that include Infrastructure Automation data, and if available, IBM Cloud Pak for AIOps data. Some datastores are backed up and restored in their entirety. Other datastores are backed up and restored depending on a particular component need.

The following table shows the Infrastructure Automation datastores that are backed up:

Table: Backed up datastores

Datastores	Backup type
`IM-Postgres`	Volume backup
`CS-edbPostgres`	Data export
`CAM-mongodb`	Volume backup
`metastore`	Data export

The following table shows the Infrastructure Automation resources that are backed up:

Table: Backed up resources

Backup resource name	Resource type
`backup-metastore`	pod
`backup-postgres`	pod
`backup-cam`	pod
`zen-secrets-aes-key`	secret
`im-iminstall`	ManageIQ
`im-iminstall`	IMInstall
`ibm-infra-management-application`	secret
`postgresql`	secret
`backup-imedb`	pod

The backup process is run by `custom backup-job(s)`. When the backup job runs, pods can be scaled down to ensure consistent backups are created. The backup job calls Velero to back up Kubernetes resources and volumes. Native backup scripts are run to back up datastores. The backup job then scales up pods to match their state before the backup process began. Datastores that have a selective backup types are backed up only when the datastore is updated since the previous backup.

For more information about the backup process, see:

- Backing up Infrastructure Automation data.
- Backing up IBM Cloud Pak for AIOps data.

Restore process

To run the restore process, you first need to create a new cluster and install the required prerequisites on the cluster. The prerequisites include installing the required CLI tools, creating the storage classes, and updating the `bcdr/common/aiops-config.json` and `bcdr/restore/restore-data.json` files with the required configuration values. Then you need to configure Velero to point to the same s3 storage where the backup is located.

You then need to create the required namespaces for IBM Cloud Pak® for AIOps by running the `restore-namespace.sh` job. This script creates only the namespaces and namespace metadata, not the contents of the namespace. This namespace restore is needed because the metadata contains SELinux settings that must match the settings from the corresponding namespaces on the cluster where you took the backup.

Then, you need to complete the installation of Infrastructure Automation. With the Infrastructure Automation instance created and Velero configured, you can run the Velero restore jobs to populate the instance. When the jobs complete you need to run any other post restore tasks that you need to complete.

Note: Velero does not support the overriding of resources during the restore. The restoration of resources alread existing in the cluster are skipped. If this occurs, sync the existing resources and create only the resources that were not restored instead of creating a new resources Operator.

For more information about the restore process, see Restoring Infrastructure Automation.

Procedures for backing up and restoring Infrastructure Automation

- Installing the backup and restore tools
- Backing up Infrastructure Automation
- Restoring Infrastructure Automation
- Upgrading Infrastructure Automation backup and restore artifacts
- Backup and Restore Infrastructure management appliances

Installing the backup and restore tools

Learn how to install the tools that are required for the backup and restore of IBM Cloud Pak® for AIOps Infrastructure Automation.

You need to install the Red Hat OpenShift APIs for Data Protection (OADP). OADP provides default Velero plug-ins that are integrated with storage providers to support backup operations.

Note: The backup and restore tools for Infrastructure Automation are the same as the tools for backing up and restoring IBM Cloud Pak for AIOps. If you have the tools installed for IBM Cloud Pak for AIOps, you do not need to complete the following procedure.

Prerequisites

- Provision the object storage (S3 bucket) for storing your backups. The backup process moves backup files to an S3 bucket, which must be in an S3 compatible location, such as Amazon Web Services (AWS).
- Ensure that the following command line tools are installed on your workstation:
  - `oc`
  - `jq`
  - `git`
- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).

Procedure

1. Install the Velero client CLI
2. Install OpenShift API for Data Protection (OADP)
3. (Offline deployments only) Mirror the backup and restore images

1. Install the Velero client CLI

Install the Velero client CLI on the workstation that you plan to use to run the scripts for backing up and restoring the cluster.

1. Run the following command to download the Velero CLI client:

   ```
   curl -LO https://github.com/vmware-tanzu/velero/releases/download/v1.12.3/velero-v1.12.3-linux-amd64.tar.gz
   ```

2. Extract the tarball:

   ```
   tar -xzf velero-v1.12.3-linux-amd64.tar.gz
   ```

3. Move the extracted velero binary to `/usr/local/bin`:

   ```
   mv velero-v1.12.3-linux-amd64/velero /usr/local/bin/velero
   ```

4. Run the following command to verify the installation.

   ```
   velero version
   ```

   Example output:

   ```
   Client:
    Version: v1.12.3
    Git commit: 06d12dec4030703cb46b2699ca434cdabd9edbb0
   velero version
   ```

2. Install Red Hat OpenShift APIs for Data Protection (OADP)

Follow the Red Hat OpenShift documentation to install Red Hat OpenShift API for Data Protection: About installing OADP.

You must install the Red Hat OpenShift APIs for Data Protection (OADP) Operator.

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository that includes the backup and restore scripts. Clone this repository by running the following command:

   ```
   git clone https://github.com/IBM/cp4waiops-samples.git
   ```

   If you have an offline (air-gapped) deployment of Infrastructure Automation, then copy the cloned scripts to your air-gapped environment.

2. Update the installation files.

   1. Change to the `install` directory where the backup and restore scripts are downloaded.

      ```
      cd <Path>/bcdr/4.8.1/install
      ```

      Where `<Path>` is the path to where you cloned the IBM Cloud Pak for AIOps backup and restore scripts.

   2. Edit the `install-oadp-config.json` script. Configure the following settings:

      - `aws_access_key_id` - Access key ID to connect to your object storage S3 bucket.
      - `aws_secret_access_key` - Secret access key to connect to your S3 bucket.
      - `bucket_name` - Name of the S3 bucket where backup data will be stored.
      - `bucket_region` - Region where the S3 bucket is deployed.
      - `namespace` - Namespace name in which you want to deploy OADP. If the namespace does not exist, it is created when you run the OADP installation script.
      - `backup_label` - Label for creating a folder structure to organize sets of backups together.

   3. Edit the `install-oadp.sh` script.

If your version of Red Hat OpenShift is higher than v4.15, update `channel: stable-1.3` to `channel: stable-1.4`.

Example excerpt:

```
spec:
  channel: stable-1.4
  installPlanApproval: Automatic
  name: redhat-oadp-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
```

3. Run the OADP installation script:

   ```
   nohup ./install-oadp.sh > install-oadp.log &
   ```

4. Check the logs by running the following command:

   ```
   tail -f install-oadp.log
   ```

5. Export an environment variable with the Velero namespace.

   ```
   export OADP_NAMESPACE=<oadpNamespace>
   ```

   Where `<oadpNamespace>` is the namespace where OADP is installed.

6. Check the Velero pods status by running the following command:

   ```
   oc get pods -n ${OADP_NAMESPACE}
   ```

   The Velero and Restic pods should be in a running state.

7. Check the status of the `backupStorageLocation` by running the following command:

   ```
   oc get backupStorageLocation -n ${OADP_NAMESPACE}
   ```

   The `backupStorageLocation` needs to be in an available status.

3. (Offline deployments only) Mirror the backup and restore image

If the deployment that you are backing up or restoring to is an offline (air-gapped) deployment, then you must copy the backup and restore image to a registry that is available to your air-gapped environment. You can do this by using either a bastion host or a portable device. For background information on offline (air-gapped) deployments, see Installing Infrastructure Automation in an air-gapped environment (offline).

Prerequisites:

- The host that you are running the mirror from must allow access to `icr.io` on port 443.
- You must have a target registry available to mirror the backup and restore image into. The target registry must adhere to the requirements in Set up a target registry.

Follow the instructions in Mirroring with a bastion host or Mirroring with a portable device.

Mirroring with a bastion host

1. Set environment variables.

   ```
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export BCDR_IMAGE=<bcdr_image>
   export BCDR_UTILS_IMAGE=cp4waiops-bcdr-utils@sha256:1db594ad10da9c7049367b5dd4373e86af7e6bcb875758022e2e743c83cf486f
   export IA_NAMESPACE=<ia_namespace>
   ```

   Where

   - `<bcdr_image>` is the name of the backup and restore image, as given in the backup helm chart `values.yaml` file, in the form `cp4waiops-bcdr@{digest}`. An example value for BCDR_IMAGE is `cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355`.
   - `<ia_namespace>` is the namespace that Infrastructure Automation is deployed in.

   If your bastion host must connect to the internet through a proxy, then also set the following environment variables:

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Authenticate with the target registry.

   You do not need to authenticate with the IBM Entitled Registry, as the image that you require for backup and restore is freely available.

   You need to authenticate with the target registry to enable the `oc image mirror` command to push the image to your target registry. Log in to your target registry to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file.

   Run the following commands:

```
podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

3. Mirror the image.

   Connect the bastion host to the internet and disconnect if from the air-gapped environment. Then run the following command to mirror the backup and restore image to the target registry:

```
oc image mirror icr.io/cpopen/cp4waiops/${BCDR_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_IMAGE} -a
${REGISTRY_AUTH_FILE} --insecure
oc image mirror icr.io/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} -
a ${REGISTRY_AUTH_FILE} --insecure
```

4. Create the `ImageContentSoucePolicy`.

   1. Switch to the namespace that Infrastructure Automation is deployed in.

```
oc project ${IA_NAMESPACE}
```

   2. Run the following command to create the `ImageContentSourcePolicy`.

```
cat << EOF | oc apply -f -
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
 name: bcdr-mirrors
spec:
 repositoryDigestMirrors:
  - mirrors:
    - ${TARGET_REGISTRY}/cpopen/cp4waiops/
    source: icr.io/cpopen/cp4waiops/
EOF
```

Mirroring with a portable device

If you are using a portable storage device, then connect it to a host with access to the internet. If you are using a portable compute device, then connect it to the internet. Run the following steps.

1. Set environment variables.

```
export IMAGE_PATH=<image-path>
export BCDR_IMAGE=<bcdr_image>
export BCDR_UTILS_IMAGE=cp4waiops-bcdr-utils@sha256:1db594ad10da9c7049367b5dd4373e86af7e6bcb875758022e2e743c83cf486f
```

   Where

   - `<image_path>` is the directory where you want the image to be stored.
   - `<bcdr_image>` is the name of the backup and restore image, as given in the backup helm chart `values.yaml` file, in the form `cp4waiops-bcdr@{digest}`. An example value for BCDR_IMAGE is `cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355`.

   If your device must connect to the internet through a proxy, then also set the following environment variables:

```
export https_proxy=http://proxy-server-hostname:port
export http_proxy=http://proxy-server-hostname:port
```

2. Mirror the images to the file system on your portable device.

   You do not need to authenticate with the IBM Entitled Registry, as the image that you require for backup and restore is freely available. Run the following command to mirror the image from the IBM Entitled Registry to the file system on your portable device.

```
oc image mirror \
icr.io/cpopen/cp4waiops/${BCDR_IMAGE} file://cpopen/cp4waiops/${BCDR_IMAGE} \
--filter-by-os '.*' \
--insecure \
--skip-multiple-scopes \
--dir "${IMAGE_PATH}" \
--max-per-registry=1

oc image mirror \
icr.io/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} file://cpopen/cp4waiops/${BCDR_UTILS_IMAGE} \
--filter-by-os '.*' \
--insecure \
--skip-multiple-scopes \
--dir "${IMAGE_PATH}" \
--max-per-registry=1
```

3. Setup the file system in the air-gapped environment.

   1. (Portable storage device only) Copy files to the air-gapped environment

      If you are using a portable storage device, you must copy the files at $IMAGE_PATH on the portable storage device to a local compute device in the air-gapped environment that has access to the target registry. If you are using a portable compute device, then these files are already present, and you can proceed to the next step.

2. Disconnect the device that has your file system (the portable compute device or the local compute device) from the internet and connect it to the air-gapped environment.

3. Set environment variables on the device in the air-gapped environment that has access to the target registry.

   If you are using a portable compute device, then set the following environment variables on it. If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

   ```
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export IMAGE_PATH=<image_path>
   export BCDR_IMAGE=<bcdr_image>
   export BCDR_UTILS_IMAGE=cp4waiops-bcdr-
   utils@sha256:1db594ad10da9c7049367b5dd4373e86af7e6bcb875758022e2e743c83cf486f
   export IA_NAMESPACE=<ia_namespace>
   ```

   Where

   - `<image_path>` is the directory where you want the image to be stored.
   - `<bcdr_image>` is the name of the backup and restore image, as given in the backup helm chart `values.yaml` file, in the form `cp4waiops-bcdr@{digest}`. An example value for BCDR_IMAGE is `cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355`.
   - `<ia_namespace>` is the namespace that Infrastructure Automation is deployed in.

4. Authenticate with the target registry.

   Authenticate with the target registry in the air-gapped environment that you are mirroring the image into. You need to authenticate with the target registry to enable the `oc image mirror` command to push the image to your target registry. Log in to your target registry to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file.

   ```
   podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
   export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
   ```

   Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

5. Mirror the images to the target registry from the file system.

   Run the following commands on the device that has your file system (the portable compute device or the local compute device) to copy the image from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat® OpenShift® Container Platform cluster that you want to backup or restore Infrastructure Automation on.

   ```
   oc image mirror \
   file://cpopen/cp4waiops/${BCDR_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_IMAGE} \
   --from-dir "${IMAGE_PATH}" \
   -a ${REGISTRY_AUTH_FILE} \
   --filter-by-os '.*' \
   --insecure  \
   --skip-multiple-scopes \
   --max-per-registry=1

   oc image mirror \
   file://cpopen/cp4waiops/${BCDR_UTILS_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} \
   --from-dir "${IMAGE_PATH}" \
   -a ${REGISTRY_AUTH_FILE} \
   --filter-by-os '.*' \
   --insecure  \
   --skip-multiple-scopes \
   --max-per-registry=1
   ```

6. Create the `ImageContentSoucePolicy`.

   1. Switch to the namespace that Infrastructure Automation is deployed in.

      ```
      oc project ${IA_NAMESPACE}
      ```

   2. Run the following command to create the `ImageContentSourcePolicy`.

      ```
      cat << EOF | oc apply -f -
      apiVersion: operator.openshift.io/v1alpha1
      kind: ImageContentSourcePolicy
      metadata:
       name: bcdr-mirrors
      spec:
       repositoryDigestMirrors:
        - mirrors:
          - ${TARGET_REGISTRY}/cpopen/cp4waiops/
          source: icr.io/cpopen/cp4waiops/
      EOF
      ```

Troubleshooting

- Red Hat OpenShift APIs for Data Protection (OADP) installation failed
- Backups are not showing after installation

Red Hat OpenShift APIs for Data Protection (OADP) installation failed

If your installation of OADP fails for any reason, delete any related files and reattempt the installation.

For example, the installation script `install-oadp.sh` might not complete successfully. If this occurs an error similar to the following message can display:

```
error: unable to recognize "STDIN": no matches for kind "DataProtectionApplication" in version
"oadp.openshift.io/v1alpha1"
```

This can occur if the network or cluster is running slow and taking too long to pull the required images. This delay causes the OADP operator to take too long to create the required custom resource.

If the preceding issue occurs, or the installation fails for another reason, run the following `oc` commands to delete any existing OADP installation files before attempting to install again.

1. Delete any cluster service versions (CSV) in the project:

   ```
   oc delete csv --all -n ${OADP_NAMESPACE}
   ```

2. Delete any subscriptions in the project:

   ```
   oc delete subscription --all -n ${OADP_NAMESPACE}
   ```

3. Delete any deployments in the project:

   ```
   oc delete deploy --all -n ${OADP_NAMESPACE}
   ```

4. Delete any daemonsets in the project:

   ```
   oc delete ds --all -n ${OADP_NAMESPACE}
   ```

5. Delete the project (namespace):

   ```
   oc delete project ${OADP_NAMESPACE}
   ```

Any existing installation is deleted and you can reattempt the installation.

Backups are not showing after installation

If your backups do not show, complete the following steps:

1. Correct the `bucket` and `prefix` property within the `backupstoragelocation` resource that is named `bcdr-s3-location`. The `bucket` and `prefix` name should match the value that is defined within the `install-oadp-config.json` file during the OADP installation.

   Run the `oc edit` command to edit the `backupstoragelocation` resource:

   ```
   oc edit backupstoragelocation bcdr-s3-location -n ${OADP_NAMESPACE}
   ```

2. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n ${OADP_NAMESPACE}
   ```

   The backups should now be available.

3. Run the following command to list the backups:

   ```
   velero get backup
   ```

Backing up Infrastructure Automation data

Learn how to back up data for Infrastructure Automation components.

The backup process uses OADP for cluster object backups and volume backups. The backup script must be the same version as your deployment.

Note: If you also need to back up data for IBM Cloud Pak for AIOps components, the process and tools to use are the same. The only change that is required is when you are packaging the Helm chart. During this step you need to configure the value of the `enabledComponents` parameter to include `'"AIOPS"'` as well as `'"IA"'`. All other steps remain the same.

Prerequisites

- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If they are not already installed, install the backup and restore tools. You need to install the Red Hat® OpenShift® Container Platform APIs for Data Protection (OADP) in the Red Hat OpenShift Container Platform cluster. For more information, see Installing the backup and restore tools.
- Provision S3 compliant object storage for your backups. You must have sufficient storage to backup the data that is contained in the datastores and persistent volume claims (PVCs) of the components that are backed up, as detailed in Backup process. The storage must also be accessible by any environment that a backup might need to be restored to. The backup process moves backup files to an S3 bucket, which must be in an S3 compatible location. For instance, you can use the following public or private cloud options for provisioning storage for your backups:
  - MinIO
  - Red Hat OpenShift Data Foundation object store

- IBM Cloud Object Storage
- AWS Cloud

Important: Ensure that the OADP is configured to point to the same object storage (S3 bucket) that includes the backup that you plan to use.

Note: As your data grows, the capacity of your backup storage might need to grow, and you must ensure that sufficient storage for backups is provisioned to accommodate any growth in backup data.

Backup procedure

Follow these steps to backup Infrastructure Automation.

1. Prepare to backup Infrastructure Automation
2. Backup Infrastructure Automation
3. Setup on-demand backups of Infrastructure Automation

1. Prepare to backup Infrastructure Automation

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository to retrieve the latest Infrastructure Automation backup and restore scripts by running the following command:

   ```
   git clone https://github.com/IBM/cp4waiops-samples.git
   ```

   If you have an offline (air-gapped) deployment of Infrastructure Automation, then copy the cloned scripts to your air-gapped environment.

2. Export the environment variables that you will need for the backup procedure.

   If you are backing up an online deployment, set the following:

   ```
   export OADP_NAMESPACE=<oadpNamespace>
   export PATH=<path>
   export IA_NAMESPACE=<ia_namespace>
   ```

   If you are backing up an offline deployment, set the following:

   ```
   export TARGET_REGISTRY_HOST=<target_registry_host>
   export TARGET_REGISTRY_PORT=<port>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export EMAIL=<email>
   export PATH=<path>
   export OADP_NAMESPACE=<oadpNamespace>
   export IA_NAMESPACE=<ia_namespace>
   ```

   Where:

   - **<target_registry_host>** is the IP address or FQDN of the target registry that holds the backup and restore images, from Offline deployments only: Mirror the backup and restore images
   - **<port>** is the port_number of the target registry
   - **<username>** is the username for the target registry
   - **<password>** is the password for the target registry
   - **<email>** is the email for the target registry
   - **<path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.
   - **<oadpNamespace>** is the OADP namespace
   - **<ia_namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.

3. Package the Helm Chart

   1. Change to the directory where you have the backup and recovery files downloaded.

      ```
      cd ${PATH}/bcdr/4.8.1/backup
      ```

   2. Update the following parameters in the **values.yaml** file. The file is located in the **./helm** directory.

      - **repository** - Do not edit this parameter.
      - **pullPolicy** - The policy for determining when to pull the image from the image registry server. For example, to force pull the image, use the **Always** policy.
      - **schedule** - The Cron expression for the automated backup. For example, to take a backup once a day, use the **0 0 * * *** Cron expression.
      - **backupStorageLocation** - The storage location where backed up data is stored. For example, **bcdr-s3-location**.

        Use the **oc get backupstoragelocation -n ${OADP_NAMESPACE}** command to get the **backupStorageLocation** on the OpenShift cluster.
      - **backupNameSuffix** - The prefix for the backup name when the backup is created by using the job. Generally, it can be name of source cluster itself. For example, **aiops-cluster-backup-106**.
      - **aiopsNamespace** - The namespace where Infrastructure Automation is installed.
      - **csNamespace** - The namespace where IBM Cloud Pak foundational services is installed. From Infrastructure Automation v4.4.0 this is the same as the namespace where Infrastructure Automation is installed.

- **oadpNamespace** - The namespace where OADP is installed.
- **ttl** - The time-to-live setting for the backup. The backed up data is retained until the **ttl** value is reached (expires). For example **720h0m0s**.
- **enabledNamespaces** The namespaces of the components that need to be backed up. You can delete any unused namespaces from the list to reduce the time that is required for the backup process. If you have installed only Infrastructure Automation, then you require the namespaces that Infrastructure Automation and IBM Cloud Pak foundational services are installed in. From Infrastructure Automation v4.4.0, the IBM Cloud Pak foundational services namespace is the same as the Infrastructure Automation namespace, but this namespace must be listed twice, as in the following example.

  ```
  enabledNamespaces:
  - '"cp4aiops"'
  - '"cp4aiops"'
  ```

- **enabledComponents** - Required. The list of components to back up.

  The backup and restore processes support backing up and restoring both the IBM Cloud Pak for AIOps and Infrastructure Automation components. These components can be installed together or independtly. You can specify to back up both, or only one of these components. Any other value is ignored and a corresponding error message is generated.

  To specify Infrastructure Automation, set the value **'"IA"'**.

  To specify IBM Cloud Pak for AIOps, set the value **'"AIOPS"'**.

  The following configuration shows both components listed:

  ```
  enabledComponents:
  - '"IA"'
  - '"AIOPS"'
  ```

3. Package the Helm Chart.

   ```
   helm package ./helm
   ```

2. Setup an automated backup of Infrastructure Automation

Run the following steps to trigger an automated backup.

1. Change to the **backup** directory where the script is located:

   ```
   cd ${PATH}/bcdr/4.8.1/backup
   ```

2. Deploy the backup job by running the following command:

   ```
   helm install backup-job clusterbackup-0.1.0.tgz
   ```

3. Optional - update the **schedule** parameter in the **values.yaml** file (located in the **./helm** directory). This uses a cron expression to schedule the automated backup. For example, to take a backup once a day, use the **0 0 * * *** Cron expression.

3. Setup on-demand backups of Infrastructure Automation

This step is optional. Use only when you do not want to wait untill the next scheduled backup job.

Prerequisite: The deployment of an automated backup job is a prerequisite for the on-demand job. Only after you initiate an automated backup job can you then trigger an on-demand backup.

1. Deploy the on-demand backup job by running the following command:

   ```
   oc create job --from=cronjob/backup-job on-demand-backup-job -n ${OADP_NAMESPACE}
   ```

2. Check the on-demand backup pods status by running the following command:

   ```
   oc get pods -n ${OADP_NAMESPACE}
   ```

3. Check the on-demand backup job logs by running the following command:

   ```
   oc logs -f <on-demand-backup-job-***> -n ${OADP_NAMESPACE}
   ```

4. Find and export the name of your backup.

   You can see the backup name after the on-demand backup job is complete. For example, you might see the backup name **aiops-cluster-backup-106-1622193915** in the on-demand backup job log as follows:

   ```
   Waiting for backup aiops-cluster-backup-106-1622193915 to complete
   ```

   Export the name of your backup by running the following command:

   ```
   export BACKUP_NAME=<backup_name>
   ```

   Where **<backup_name>** is the name of the backup.

5. Check the backup status by running the following command:

   ```
   velero get backup ${BACKUP_NAME} -n ${OADP_NAMESPACE}
   ```

When the backup is finished, the status is **Completed**, such as in the following sample output:

```
aiops-backup-1654171690-testfest34   Completed   2022-06-02 05:08:11 -0700 PDT   29d   bcdr-data-protection-app-1
cp4aiops.ibm.com/backup=t
```

To see more details, you can also:

- Check the backup pods' status by running the following command:

  ```
  oc get pods -n ${OADP_NAMESPACE}
  ```

- Check the backup job logs by running the following command:

  ```
  oc logs -f <backup-job-***>
  ```

6. Review the backup result summary that is included within the logs of the backup job.

   Note: A **backup-result-config** ConfigMap is created to include this result. To view this ConfigMap, run the following command:

   ```
   oc get cm backup-result-config -n ${IA_NAMESPACE} -o yaml
   ```

Troubleshooting

- Backup process is stuck in an *In progress* state
- Classifier and layout pods remain not ready after running a backup
- Helm install backup job command failed

Backup process is stuck in an *In progress* state

If your backups process is stuck in an *In progress* state for an unexpected duration, complete the following steps. This procedure stops the backup process so that you can try the backup again.

1. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n <oadpNamespace>
   ```

   Where **<oadpNamespace>** is the namespace where OADP is installed, and **<pod>** is the name of the Velero pod.

2. Run the backup that is stuck in the *In progress* state:

   ```
   velero delete backup <backup>
   ```

   Where **<backup>** is the backup that you want to delete. The backup process should begin again.

3. Wait for the backup to complete and verify that the backup is created.

Classifier and layout pods remain not ready after running a backup

After you complete a backup, you might notice that the classifier (**aiops-ir-analytics-classifier**) and layout (**aiops-topology-layout**) pods do not change to a ready state, which can cause **AIOpsAnalyticsOrchestrator** to be stuck in an updating phase. This issue can occur after most backups. To resolve this issue, you need to either restart the pods or increase the resource limits for the pods.

- For the classifier (**aiops-ir-analytics-classifier**) pod, restart the pod.
- For the layout (**aiops-topology-layout**) pod, restart the pod. If the restart does not result in the pod becoming ready, increase the resource limits for the pod. Doubling the **cpu** and **memory** limits and request allocations for the pods can result in the pod starting and entering a ready state.

Helm install backup job command failed

When you are running the **helm install backup-job clusterbackup-0.1.0.tgz** command, you might encounter the command failing with an error that is similar to the following error:

```
Error: admission webhook "trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the request:
Deny "icr.io/cpopen/cp4waiops/cp4aiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355", no
matching repositories in ClusterImagePolicy and no ImagePolicies in the "velero" namespace
```

If you encounter this error, complete the following steps to resolve the issue:

1. Uninstall the **backup-job** job by running the following command:

   ```
   helm uninstall backup-job -n ${OADP_NAMESPACE}
   ```

2. Export an environment variable for the image.

   For an online deployment:

   ```
   export REGISTRY=icr.io/cpopen/cp4waiops/cp4aiops-bcdr
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   For an offline deployment:

   ```
   export REGISTRY=$TARGET_REGISTRY
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

Where **<bcdr_image>** is the name of the backup and restore image, as given in the backup helm chart **values.yaml** file, in the form **cp4waiops-bcdr@{digest}**. An example value for BCDR_IMAGE is **icr.io/cpopen/cp4waiops/cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355**.

3. Create a **backup-image-policy.yaml** file and add the following content within the file:

```
apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
kind: ClusterImagePolicy
metadata:
  name: backup-image-policy
spec:
 repositories:
  - name: ${BCDR_IMAGE}
    policy:
```

4. Apply the policy by running the following command:

```
oc apply -f backup-image-policy.yaml
```

5. Deploy the backup job by running the following command:

```
helm install backup-job clusterbackup-0.1.0.tgz
```

Restoring Infrastructure Automation data

Learn how to restore data for Infrastructure Automation components to a cluster, such as for disaster recovery.

The following procedure restores all backed up data that exists in the specified backup for Infrastructure Automation components. The steps in the following procedure restore Infrastructure Automation in a new cluster.

Before you begin

- All required storage classes must be created before you run the restore process. The storage classes must have the same name as the backup cluster.
- Custom configuration settings for the Infrastructure Automation - Managed services component, such as Ansible replica count, extra variables, might not be backed up and restored. If you need this data and resources to be included in restored clusters, you need to directly add the data or resources to the restored cluster.
- You can restore a backup only within an environment that has the same version of Infrastructure Automation as the environment where the backup was created. For example, a backup of an Infrastructure Automation 4.8.1 environment must be restored within a cluster that has Infrastructure Automation 4.8.1 installed. If you need to upgrade as well as restore data, complete the restore process before you upgrade.
- If you are also restoring IBM Cloud Pak for AIOps data, the overall procedure is mostly the same, with the following additional steps required:
  - After you install IBM Cloud Pak for AIOps on the cluster where you are restoring data, you need to install Infrastructure Automation.
  - After you restore your IBM Cloud Pak for AIOps data, you need to restore your Infrastructure Automation data.

    For more information, see Restoring IBM Cloud Pak for AIOps.

Restore procedure

Follow the steps to restore Infrastructure Automation from backup.

1. Set up your new cluster for backup and restore.
2. Prepare the backup data for restoring.
3. Restore the cluster namespaces and install Infrastructure Automation
4. Optional. Restore IBM Cloud Pak for AIOps data
5. Restore the Infrastructure Automation data

If you encounter any issues with the restore process, see Troubleshooting

1. Set up your new cluster for backup and restore

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.

   IBM Cloud Pak for AIOps requires OpenShift to be installed and running. You must have administrative access to your OpenShift cluster.

   Important: Ensure that the version of Red Hat OpenShift Container Platform that you install is the same as the version that was installed in the backed-up cluster.

   For information about the supported versions of OpenShift, see Supported Red Hat OpenShift Container Platform versions.

   Note: Infrastructure Automation uses the OpenShift image registry when it builds images in real time. If the OpenShift image registry is not persistent and the registry respawns, then workloads can temporarily fail until respawning is complete. A persistent OpenShift image registry is recommended to avoid this issue. For more information, see Setting up and configuring the registry in the Red Hat OpenShift Container Platform documentation.

2. Install the OpenShift command-line interface (**oc**) on your cluster's boot node and run **oc login**, using the instructions in Getting started with the Red Hat OpenShift CLI.

3. Configure storage

The storage configuration must satisfy your sizing requirements. For more information about the storage classes that are needed for installing IBM Cloud Pak for AIOps, see Storage.

Important: All required storage classes must be created before running the restore process. The storage classes must have the same name as the backup cluster.

4. Install the backup and restore tools

   Install the Red Hat OpenShift APIs for Data Protection (OADP) in the Red Hat OpenShift Container Platform cluster. For more information, see Installing the backup and restore tools.

   Important: Ensure that the OADP is configured to point to the same object storage (S3 bucket) that includes the backup that you plan to use.

5. Export the environment variables that you will need for the restore procedure.

   If you are restoring to an online deployment, set the following:

   ```
   export PATH=<path>
   export OADP_NAMESPACE=<oadpNamespace>
   ```

   If you are restoring to an offline deployment, set the following:

   ```
   export TARGET_REGISTRY_HOST=<target_registry_host>
   export TARGET_REGISTRY_PORT=<port>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export EMAIL=<email>
   export PATH=<path>
   export OADP_NAMESPACE=<oadpNamespace>
   ```

   Where:

   - `<target_registry_host>` is the IP address or FQDN of the target registry that holds the backup and restore images, from Offline deployments only: Mirror the backup and restore images
   - `<port>` is the port_number of the target registry
   - `<username>` is the username for the target registry
   - `<password>` is the password for the target registry
   - `<email>` is the email for the target registry
   - `<path>` is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.
   - `<oadpNamespace>` is the OADP namespace

2. Prepare the backup data for restoring

Verify your backed up data and prepare the data for restoring.

1. Check the backup status

   Check the backup status to ensure that the backup that you want to restore in your cluster is complete. Run the following command to check the contents of the backup:

   ```
   velero describe backup <backup-name> --details
   ```

   The output should list the backed-up data for Infrastructure Automation and for IBM Cloud Pak foundational services. If you also backed up IBM Cloud Pak for AIOps data, this data (`cp4aiops/*`) should also be listed.

2. Package and install the Helm Chart

   1. Change to the `restore` directory where you need to package the Helm Chart:

      ```
      cd ${PATH}/bcdr/4.8.1/restore
      ```

   2. Update the following parameters in the `values.yaml` file. The file is located in the `./helm` directory:

      - `backupName` - The name of the backup that you are restoring.
      - `aiopsNamespace` - The namespace where IBM Cloud Pak for AIOps is installed.
      - `csNamespace` - The namespace where IBM Cloud Pak foundational services is installed. In IBM Cloud Pak for AIOps v4.8.1 this is the same as the namespace where IBM Cloud Pak for AIOps is installed.
      - `oadpNamespace` - The namespace where OADP is installed.

   3. Package the Helm Chart.

      ```
      helm package ./helm
      ```

   4. Install the Helm Chart for restoring data by running the following job:

      ```
      helm install restore-job clusterrestore-0.1.0.tgz
      ```

3. Restore the cluster namespaces and install Infrastructure Automation

Since the restore job does not install Infrastructure Automation, you need to first install Infrastructure Automation before you can run the restore jobs for restoring database and component data. A restore job for restoring the cluster namespaces is available and must be run before you install Infrastructure Automation.

1. Restore the cluster namespaces

   You need to restore the projects (namespaces) of the backed-up cluster so that your new cluster includes the metadata with the SELinux settings that need to match the settings for the backup data that you plan to restore.

These steps only restore the namespaces and namespace metadata. The commands do not restore the contents of the namespaces.

Note: In Infrastructure Automation v4.8.1, the IBM Cloud Pak foundational services namespace is the same as the Infrastructure Automation namespace.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd ${PATH}/bcdr/4.8.1/restore
   ```

2. Optional. Delete any existing namespace restore jobs:

   ```
   oc delete -f ns-restore-job.yaml
   ```

3. Create a job to restore the cluster namespaces:

   ```
   oc create -f ns-restore-job.yaml
   ```

4. Check the restore job logs by running the following command:

   ```
   oc logs -f <ns-restore-job-***>
   ```

5. Check the velero-restore status for the namespace by running the following command:

   ```
   velero get restore <RESTORE_NAME>
   ```

   Where **<RESTORE_NAME>** is the name of the restore for namespace.

   You can view the restore name after the restore job is completed. For example, you might see the restore name **aiops-namespace-restore-20221006054710** within the restore job log as follows:

   ```
   Restore request "aiops-namespace-restore-20221006054710" submitted successfully.
   ```

   Ensure that the projects (namespaces) are restored before you proceed.

2. Optional. Install IBM Cloud Pak for AIOps

   If you also need to restore IBM Cloud Pak for AIOps data, you need to install IBM Cloud Pak for AIOps before you install Infrastructure Automation. For more information, see Installing IBM Cloud Pak for AIOps.

   - For more information about restoring IBM Cloud Pak for AIOps, see Restoring IBM Cloud Pak for AIOps.

   Important: Ensure that the version of IBM Cloud Pak for AIOps that you are installing is the same as the version that was installed in the backed-up cluster.

3. Install Infrastructure Automation.

   For more information, see Installing Infrastructure Automation.

   Note: When you install Infrastructure Automation, the Infrastructure Automation operator and the IAConfig CR are created.

   - If you install the Managed services component, the Managed Services operator is also installed and the corresponding custom resource (CR) is automatically created.
   - If you install the Infrastructure management component, only the Infrastructure management operator is installed. The corresponding CR is not created. You do not need to create this CR as it is created during the restore process in the following steps.

   Important:

   - Ensure that the version of IBM Cloud Pak for AIOps that you are installing is the same as the version that was installed in the backed up cluster.
   - The backup includes keys and certificates from the backed-up cluster. Ensure that your new cluster is configured to support the use of these keys and certificates so that the restored data can be accessed.
   - Wait until the installation is complete and all pods in the IBM Cloud Pak for AIOps project (namespace) are running before you proceed with the following restore steps.

4. Optional. Restore the IBM Cloud Pak for AIOps data

If you are also restoring IBM Cloud Pak for AIOps data, run the commands to restore the IBM Cloud Pak for AIOps data before you restore your Infrastructure Automation data. For more information, see Restoring IBM Cloud Pak for AIOps.

5. Restore the Infrastructure Automation data

This procedure restores the data for all backed up Infrastructure Automation databases and components. Optionally, you can choose to restore only individual components. For details, see Restoring individual components.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd ${PATH}/bcdr/4.8.1/restore
   ```

2. Optional. Delete any existing Infrastructure Automation restore jobs:

   ```
   oc delete -f ia-restore-job.yaml
   ```

3. Create a job to restore Infrastructure Automation:

   ```
   oc create -f ia-restore-job.yaml
   ```

4. Check the restore job logs by running the following command:

   ```
   oc logs -f <ia-restore-job-***>
   ```

5. Check the velero-restore status by running the following command:

```
velero get restore <RESTORE_NAME>
```

Where **<RESTORE_NAME>** is the name of the restore for Infrastructure Automation.

You can view the restore name after the restore job is completed. For example, you might see the restore name **cam-restore-20221006054710** for the Managed services restore in the restore job log as follows:

```
Restore request "cam-restore-20221006054710" submitted successfully.
```

Similarly, the velero restore name for other components can display in the restore job log.

6. When the restore is completed, and all **infrastructure-management** pods are in a running started, restart the **zen-watcher** pod. Run the following command to restart the pod:

   ```
   oc delete pod -l app.kubernetes.io/component=zen-watcher -n <aiopsNamespace>
   ```

   Where **aiopsNamespace** is the namespace where IBM Cloud Pak for AIOps is installed.

Restoring individual components

If needed, you can choose to restore data for specific individual databases and components instead of for all databases and components at the same time.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd ${PATH}/bcdr/4.8.1/restore
   ```

2. Copy the **ia-restore-job.yaml** file. Rename your new file **<component>-restore-job.yaml**, where **<component>** is the name of the component that you are restoring.

   For example, if you are restoring Managed services, rename the file to be **cam-restore-job.yaml**. If you are restoring Infrastructure management, rename the file to be **im-restore-job.yaml**.

3. Open the new **<component>-restore-job.yaml** file for editing. Update the **name** and **command** sections to match the values for the component that you are restoring:

   - Update the **name** of the restore job in the metadata section to be the individual component job, such as **cam-restore-job**.
   - Update the command section **command: ["/bin/bash", "restore.sh","-ia"]** to replace **-ia** with the respective argument for the component that you are restoring. For the list of component arguments, see the table that follows this procedure.

4. Create a job to restore the individual component.

   ```
   oc create -f <component>-restore-job.yaml
   ```

5. Check the restore job logs by running the following command:

   ```
   oc logs -f <cp4waiops-component-restore-job-***>
   ```

6. Check the velero-restore status by running the following command:

   ```
   velero get restore <RESTORE_NAME>
   ```

   Where **<RESTORE_NAME>** is one of the restore names for the component.

   You can view the names for the component in the restore job log when the restore job is completed. For example, the restore name for a Managed services restore can be **cam-restore-20221006054710**, which can display in an entry similar to the following example log entry:

   ```
   Restore request "cam-restore-20221006054710" submitted successfully.
   ```

Component or database arguments for restore job command configuration

Table. Component or database arguments for restore job commands

Component or Database	Argument
Cassandra	**-cassandra**
CouchDB	**-couchdb**
Elasticsearch	**-es**
Metastore	**-metastore**
Minio	**-minio**
Postgres	**-postgres**
IBM Cloud Pak foundational services	**-cs**
Managed services	**-cam**
Infrastructure management	**-im**

Troubleshooting

- LDAP user login is not working after a restore
- Restore process is stuck in an *In progress* state
- Helm install restore job command failed
- Restore process terminated mid-process with partial data available

- Managed services instance deployment fails due to a socket hang up
- Infrastructure management pods are not running after a restore
- Infrastructure management URL in the navigation panel points to the cluster where the backup was taken
- Troubleshooting the IBM Cloud Pak for AIOps restore

LDAP user login is not working after a restore

Follow the steps to solve the problem:

1. Log in to the console as the default admin user.
2. From the main navigation menu, click **Administer** > **Identify and access**.
3. Select the LDAP connection, and click **Edit connection**. Edit the LDAP connection with the correct information.
4. Click **Test connection**.
5. Click **Save** once the connection is success.
6. log in to the console with the LDAP user's credentials.

Restore process is stuck in an *In progress* state

If your backups remain stuck in an *In progress* state for an unexpected duration, complete the following steps. This procedure stops the backup process so that you can try the backup again.

1. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n <oadpNamespace>
   ```

   Where **<oadpNamespace>** is the namespace where OADP is installed, and **<pod>** is the name of the Velero pod.

2. Delete the restore that is stuck in the *In progress* state:

   ```
   velero delete restore <restore>
   ```

   Where **<restore>** is the restore process that you want to delete. The process should begin again.

3. Wait for the process to complete and verify that the restored data is available.

Helm install restore job command failed

When you are running the **helm install restore-job clusterrestore-0.1.0.tgz** command, you might encounter the command failing with an error that is similar to the following error:

```
Error: admission webhook "trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the request:
Deny "icr.io/cpopen/cp4waiops/cp4aiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355", no
matching repositories in ClusterImagePolicy and no ImagePolicies in the "velero" namespace
```

If you encounter this error, complete the following steps to resolve the issue:

1. Uninstall the **restore-job** job by running the following command:

   ```
   helm uninstall restore-job -n $OADP_NAMESPACE
   ```

2. Export an environment variable for the image.

   For an online deployment:

   ```
   export REGISTRY=icr.io/cpopen/cp4waiops/cp4aiops-bcdr
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   For an offline deployment:

   ```
   export REGISTRY=$TARGET_REGISTRY
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   Where **<bcdr_image>** is the name of the backup and restore image, as given in the backup helm chart **values.yaml** file, in the form **cp4waiops-bcdr@{digest}**. An example value for BCDR_IMAGE is **icr.io/cpopen/cp4waiops/cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355**.

3. Create a **restore-image-policy.yaml** file and add the following content within the file:

   ```
   apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
   kind: ClusterImagePolicy
   metadata:
     name: restore-image-policy
   spec:
    repositories:
     - name: ${BCDR_IMAGE}
       policy:
   ```

4. Apply the policy by running the following command:

   ```
   oc apply -f restore-image-policy.yaml
   ```

5. Deploy the backup job by running the following command:

   ```
   helm install restore-job clusterbackup-0.1.0.tgz
   ```

Restore process terminated mid-process with partial data available

If the restore process does not complete as expected for a data store, such as if it is aborted or terminated during its run, data might not be restored correctly. This incomplete restored data neeeds to be removed before you run the restore process again. To remove the data, you need to run a post-restore cleanup script to clean up the data store.

To run a script, complete the following steps:

1. Define the following environment variable on your workstation:

   ```
   export WORKDIR="<Path>/bcdr/4.8.1/"
   ```

   Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

2. Change to the **restore** directory where the post-restore cleanup script is located:

   ```
   cd <Path>/bcdr/4.8.1/restore/<data_store>/
   ```

   Where

   - **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.
   - **<data_store>** is the directory for the data store or custom resource that needs to be cleaned up. For example, **couchdb** is the directory for CouchDb.

3. Run the post-restore cleanup script:

   ```
   nohup ./<data-store>-post-restore.sh.sh > <data-store>-post-restore.log &
   ```

   Where **<data_store>** is the data store or custom resource that needs to be cleaned up.

4. Run the restore process for that data store or resource again.

For example, if the restore job for restoring the Cassandra data store aborted, you need to run the **cassandra-post-restore.sh** post-restore script that is stored in the **bcdr/restore/cassandra** directory to clean the data. Then, run the **cassandra-native-post-restore.sh** post-restore script.

The following table lists the cleanup script to run for each data store:

Table. Cleanup script for components

Component or Database	Cleanup script
Cassandra	Run first: **bcdr/4.8.1/restore/cassandra/cassandra-post-restore.sh** Then, run: **bcdr/4.8.1/restore/cassandra/cassandra-native-post-restore.sh**
CouchDB	**bcdr/4.8.1/restore/couchdb/couchdb-post-restore.sh**
Elasticsearch	**bcdr/4.8.1/restore/elasticsearch/es-post-restore.sh**
Metastore	**bcdr/4.8.1/restore/metastore/metastore-post-restore.sh**
Minio	**bcdr/4.8.1/restore/minio/minio-post-restore.sh**
Postgres	**bcdr/4.8.1/restore/postgres/postgres-post-restore.sh**
IBM Cloud Pak foundational services	**bcdr/4.8.1/restore/common-services/cs-post-restore.sh**
Managed services	**bcdr/4.8.1/restore/cam/cam-post-restore.sh**
Infrastructure Management	**bcdr/4.8.1/restore/infrastructure-management/im-cleanup-restore.sh**

Managed services instance deployment fails due to a socket hang up

After your restore a Managed services (**cam**) backup and Managed services is deployed, the Managed services instance can fail due to a socket hang up issue. If this issue ocurs, restart the **cam-iaas** pod:

```
oc delete pod <cam-iaas-xxxx> -n <aiopsNamespace>
```

Where **aiopsNamespace** is the namespace where IBM Cloud Pak for AIOps is installed.

Infrastructure management pods are not running after a restore

Following a restore, you can notice that the Infrastructure Management pods (prefixed with "1-") are showing as not running.

For example,

```
oc get pod |grep postgresql
postgresql-6cff46dcdc-g5cn8                                       0/1     Running     0                33m
```

This issue can occur when Postgres is not fully initialised. To resolve this issue, if the pod is not showing a ready state (**1/1**), restart the pod manually. This restart enables the Infrastructure Management pods to start.

Infrastructure management URL in the navigation panel points to the cluster where the backup was taken

Following a restore, the Infrastructure Management URL in the navigation panel points to the cluster where the backup was taken. To resolve this issue, restart the zen watcher pod on the restored cluster:

```
oc delete pod -l app.kubernetes.io/component=zen-watcher
```

Troubleshooting the IBM Cloud Pak for AIOps restore

If you are also restoring IBM Cloud Pak for AIOps data and encounter an issue with the restore process for IBM Cloud Pak for AIOps or encounter an issue with data not being available or processed after the restore, see Troubleshooting IBM Cloud Pak for AIOps restore.

Upgrading Infrastructure Automation backup and restore artifacts

Use this procedure if you want to run backup or restore on an upgraded Infrastructure Automation deployment.

If you run a backup or restore on an Infrastructure Automation deployment, the backup and restore scripts must always be the same version as your deployment. For example, if you run backup on an Infrastructure Automation v4.8.1 deployment, you must use a v4.8.1 backup script.

If you previously set up backup or restore on a deployment that you have now upgraded, then you must cleanup any existing backup or restore artifacts. If you then want to run a new backup or restore, you must get the new version of the backup or restore scripts.

Note: If you have never set up backup or restore and you want to run backup or restore, then use the instructions in Infrastructure Automation backup and restore.

Set up backup on an upgraded Infrastructure Automation deployment

Use the following steps to upgrade your backup artifacts.

1. Uninstall the backup helm chart.

   Run the following command in the namespace that **Red Hat OpenShift APIs for Data Protection** (OADP) is installed in.

   ```
   helm uninstall backup-job
   ```

2. Set up backup for Infrastructure Automation v4.8.1.

   If you need to run backup for Infrastructure Automation v4.8.1, follow the instructions in Backing up Infrastructure Automation data to set up backup.

   Important: The **cp4waiops-bcdr** image must be the same version as your Infrastructure Automation deployment. Follow these instructions to ensure that your **cp4waiops-bcdr** image is v4.8.1 for your Infrastructure Automation v4.8.1 deployment.

Set up restore on an upgraded Infrastructure Automation deployment

Use the following steps to upgrade your backup artifacts.

1. Uninstall the restore helm chart.

   Run the following command in the namespace that **Red Hat OpenShift APIs for Data Protection** (OADP) is installed in.

   ```
   helm uninstall restore-job
   ```

2. Set up restore for Infrastructure Automation v4.8.1.

   If you need to run restore for Infrastructure Automation v4.8.1, follow the instructions in Restoring Infrastructure Automation data.

   Important: The **cp4waiops-bcdr** image must be the same version as your Infrastructure Automation deployment. Follow these instructions to ensure that your **cp4waiops-bcdr** image is v4.8.1 for your Infrastructure Automation v4.8.1 deployment.

- Backup
- Scheduling Database Backups
- Restore

Backup

Creating a backup of the appliance will be done using SSH or the console.

1. Launch `appliance_console`
   1. Select **Create Database Backup** and choose a backup location (/tmp/evm_db.backup is the default)
   2. Select **Quit** to return to the shell
2. For convenience, create a TGZ containing all files required to restore the database

   ```
   tar -czvf /root/backup.tgz /tmp/evm_db.backup /var/www/miq/vmdb/GUID /var/www/miq/vmdb/certs/v2_key
   ```

3. Save and/or rename `/root/backup.tgz`

Scheduling Database Backups

Scheduling a database backup using crontab and pg_dump. This example will create a backup file in **/tmp** named something like **db_dump20231009T145028.dump**. Before adding the cron job it would be best practice to mount an external storage and adjust the example accordingly.

1. SSH into the appliance and edit the root user crontab `crontab -e`
2. Press `i` to insert and add a line with the desired schedule to run the backup job. Examples:

   ```
   # Every hour on the hour
   0 * * * * pg_dump -Fc vmdb_production > /tmp/db_dump$(date +\%Y\%m\%dT\%H\%M\%S).dump
   # Every day at midnight
   0 0 * * * pg_dump -Fc vmdb_production > /tmp/db_dump$(date +\%Y\%m\%dT\%H\%M\%S).dump
   # Every week at midnight on Sunday
   0 0 * * 0 pg_dump -Fc vmdb_production > /tmp/db_dump$(date +\%Y\%m\%dT\%H\%M\%S).dump
   ```

3. Press `Esc` to exit insert mode then `:wq` and hit `Enter` to write and quit. If successful, it will output `crontab: installing new crontab`
4. Done. If needed, logs can be found at `/var/log/cron`

Restore

Restoring a backup of the appliance will be done using SSH or the console.

1. Set the hostname on the new appliance. On the command line:

```
hostnamectl hostname your-new-hostname.example.com
echo "your.ip.address your-new-hostname.example.com your-new-hostname" >> /etc/hosts
```

2. Download a backup TGZ and place it in `/`
3. On the command line:

```
cd /
tar -zxf /backup.tgz
rm -f /backup.tgz # if desired to save space
```

4. Launch `appliance_console`
    1. Since the database has not yet been initialized, select `Configure Application`.
        1. When prompted to configure the database, select `Create Internal Database`
        2. When prompted to configure messaging, either connect to an external messaging system or configure this appliance as a messaging server
        3. Choose a database disk
        4. Answer y or n to “Should this appliance run as a standalone database server?”
        5. Enter a region number (0), this is not important and will be overwritten on restore
        6. Set a database password
        7. Wait for database initialization to complete.
    2. Select `Stop EVM Server Processes`
    3. Select `Restore Database From Backup`
        - Decide whether you want to delete the backup file in /tmp after restore. (Yes)
    4. Select `Start EVM Server Processes`
5. After a few minutes, you can log into the web UI and see all of the data contained in the backup.

Migrating infrastructure data to Infrastructure Automation

You can migrate infrastructure data, such as services data and appliances, to Infrastructure Automation from other products and services, such as IBM Cloud Pak for Multicloud Management, IBM Cloud Private, and IBM Red Hat CloudForms.

- Migrating IBM Cloud Private Managed services
- Upgrading IBM Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances

Migrating IBM Cloud Private Managed services to Infrastructure Automation

This topic describes the process of migrating Managed services from IBM Cloud Private to Infrastructure Automation.

Most of the Managed services pods are stateless, except MongoDB pod. The MongoDB pod stores state in external volumes, that is, persistent volumes. You must do the migration for the following resources:

- Logs

  The logs from different MongoDB pods in IBM Cloud Private are stored in persistent volumes, but in Infrastructure Automation, the pods don't use persistent volumes to store logs. The Infrastructure Automation uses `system.out` as default logging mechanism. It is recommended to use an external log aggregator like Kibana for the IBM Cloud Private customers.

- Terraform binaries

  In IBM Cloud Private, the custom terraform binaries are stored in persistent volumes whereas in Infrastructure Automation, the custom terraform binaries are stored in MongoDB. If in case, you have a custom terraform binaries or want to use binary version's that is not shipped anymore as part of the Infrastructure Automation release, then you can upload using Terraform file APIs.

- MongoDB

  The MongoDB pod contains IBM Cloud Private state. This state needs to be backed up and restored in Infrastructure Automation. For more information, see Backing up MongoDB in IBM Cloud Private and Restoring MongoDB.

Backing up MongoDB in IBM Cloud Private

1. Stop the running pods by setting the replica count to 0 and ensure that nothing is written to the MongoDB pod or scale down the MongoDB pod to 0.

2. Start the MongoDB pod and set replica of MongoDB pod count to 1.

3. Switch to the service namespace.

```
kubectl config set-context $(kubectl config current-context) --namespace=services
```

4. Create a new MongoDB pod using the **mongopod.yaml** file and ensure to use the same labels as other pods.

```
kind: Pod
apiVersion: v1
metadata:
  name: cam-mongo-backup
  labels:
    app.kubernetes.io/instance: cam
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: cam-manageservice
    certmanager.k8s.io/time-restarted: 2022-2-23.1227
    helm.sh/chart: manageservice-4.2.0
    name: cam-mongo-backup
    release: cam
  annotations:
    openshift.io/scc: anyuid
spec:
  containers:
    - name: backup-mongo
      image: mongo:4.0.16
```

Note: Get the mongo version of the targeted cluster by executing inside the mongo pod and running **mongo --version**. Use the same tag of mongo image while creating the backup mongo pod.

Run **kubectl apply -f mongopod.yaml** to create the **cam-mongo-backup** pod and wait for the pod to get started.

If you encounter with an error message like **no matching repositories in ClusterImagePolicy** while applying the **kubectl**, then edit the cluster policy and add image policy for the MongoDB pod as following:

```
kubectl  edit ClusterImagePolicy image-policy

-name:docker.io/mongo:*
```

5. Get the IP address of MongoDB pod.

```
MONGO_IP=$(kubectl get -n management-infrastructure-management svc/cam-mongo --no-headers | awk '{print $3}')
```

6. Get password to connect to the MongoDB pod.

```
MONGO_PASSWORD=$(kubectl get -n management-infrastructure-management secret cam-secure-values-secret -o yaml | grep
mongoDbPassword: | awk '{print $2}' | base64 --decode)
```

7. Get the username and port by describing **cam-mongo** pod.

```
kubectl describe pod cam-mongo
```

8. Execute the **mongodump** from **cam-mongo-backup** pod to create a zip file.

```
kubectl -it exec cam-mongo-backup -- bash -c "mongodump --ssl --sslAllowInvalidCertificates --uri
mongodb://MONGODB_USERNAME:$MONGO_PASSWORD@$MONGO_IP:27017/cam?ssl=true --archive=/tmp/mongo_backup.gz --gzip"
```

Note: Verify whether the tar file is present at **/tmp** location in the pod.

9. Run the following command to copy the back up archive to your local set up, for example, the **/tmp/dbbackup** directory

```
kubectl cp cam-mongo-backup:/tmp/mongo_backup.gz /tmp/dbbackup
```

10. Delete **cam-mongo-backup** pod.

```
kubectl delete pod cam-mongo-backup
```

11. Restore the replica count for the pods that you scaled down in a previous step.

Restoring MongoDB in Infrastructure Automation

The MongoDB contains Managed services state and to restore this state, do the following.

1. Stop the running pods by setting the replica count to 0 and ensure that nothing is written to the MongoDB pod or scale down the MongoDB pod to 0.
2. Start the MongoDB pod and set replica of MongoDB pod count to 1.
3. Switch to the **cp4aiops** namespace.

```
oc project cp4aiops
```

4. Create a new MongoDB pod using the **mongopod.yaml** file and ensure to use the same labels as other pods:

```
kind: Pod
apiVersion: v1
metadata:
  name: cam-mongo-backup
  labels:
    app.kubernetes.io/instance: cam
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: cam-manageservice
```

```
    certmanager.k8s.io/time-restarted: 2022-2-23.1227
    helm.sh/chart: manageservice-4.2.0
    name: cam-mongo-backup
    release: cam
  annotations:
    openshift.io/scc: anyuid
spec:
  containers:
    - name: backup-mongo
      image: mongo:4.0.16
```

Note: Get the mongo version of the targeted cluster by executing inside the mongo pod and running `mongo --version`. Use the same tag of mongo image while creating the backup mongo pod.

Run `oc apply -f mongopod.yaml` to create the `cam-mongo-backup` pod and wait for the pod to get started.

5. Copy the desired back up archive to restore into the `cam-mongo-backup` container.

```
oc rsync /tmp/dbbackup/ cam-mongo-backup:/tmp/
```

6. Get the IP address of MongoDB pod.

```
MONGO_IP=$(kubectl get -n cp4aiops svc/cam-mongo --no-headers | awk '{print $3}')
```

7. Get password to connect to the MongoDB pod.

```
MONGO_PASSWORD=$(kubectl get -n cp4aiops secret cam-secure-values-secret -o yaml | grep mongoDbPassword: | awk
'{print $2}' | base64 --decode)
```

8. Get the username and port by describing `cam-mongo pod`.

```
oc describe pod cam-momgo
```

9. Restore the data by running the following command:

```
oc -it exec cam-mongo-backup -- bash -c "mongorestore --ssl --sslAllowInvalidCertificates --uri
mongodb://camuser:$MONGO_PASSWORD@$MONGO_IP:27017/cam?ssl=true --archive=/tmp/mongo_backup.gz --gzip --drop"
```

10. Delete `cam-mongo-backup` pod.

```
oc delete pod cam-mongo-backup
```

11. Restore the replica count for the pods that you scaled down in a previous step.

12. For the services that are not Globally accessible, you need to create the namespaces if they do not already exist. Create the namespaces in the cluster where you are restoring data so that the services can be accessed.

Migrating IBM Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances to Infrastructure Automation

This topic describes the process of doing a full appliance replacement upgrade when a new release of Infrastructure Automation – Infrastructure Management is available.

The following migration paths are supported:

- IBM Cloud Pak for Multicloud Management 2.3.x - Infrastructure Management to IBM Cloud Pak for AIOps 4.8.1 - Infrastructure Automation
- IBM Red Hat CloudForms 4.7 through 5.0 to IBM Cloud Pak for AIOps 4.8.1 - Infrastructure Automation

Ordering migrations

You must complete all migrations of the databases before you configure replication and high availability. The reason is because you deploy new appliances for migrating to Infrastructure Management.

Important: Do not run `evmserverd` if a database is in a partially migrated state.

Modifying the PostgreSQL configuration

This procedure modifies the strategy that is used for updating the PostgreSQL configuration. Previously, changes were made directly to the `postgresql.conf` file in the data directory. After you update the new configuration, changes will be made by editing a separate file in the `/etc/manageiq/postgresql.conf.d` directory. This modification allows future PostgreSQL configuration changes to be made more easily.

Note: The `01_miq_overrides.conf` file in the `/etc/manageiq/postgresql.conf.d` directory is reserved for application overrides and is re-created on application updates. **Important:** This file must not be edited. More files are read by the PostgreSQL server process according to the documentation here: PostgreSQL - Managing Configuration File Contents

1. Back up the `postgresql.conf` file so that you have any configuration changes:

```
cp $APPLIANCE_PG_DATA/postgresql.conf <file_location>
```

2. Copy the updated `postgresql.conf` file to the `pg_data` directory:

```
cp $APPLIANCE_SOURCE_DIRECTORY/TEMPLATE/var/opt/rh/rh-postgresql95/lib/pgsql/data/postgresql.conf
$APPLIANCE_PG_DATA/postgresql.conf
```

3. Optional: If you updated the **postgresql.conf** file before this procedure, create a new **02_user_overrides.conf** file, and add any modifications that you have from the original **postgresql.conf** file. Example layout of the **02_user_overrides.conf** file:

```
#--------------------------------------------
# WRITE AHEAD LOG
#--------------------------------------------

wal_level = 'logical'
wal_log_hints = on
wal_buffers = 16MB
min_wal_size = 1GB
max_wal_size = 2GB
checkpoint_completion_target = 0.9
```

After the **02_user_overrides.conf** file is modified and saved, restart the postgresql service:

```
systemctl restart $APPLIANCE_PG_SERVICE
```

All **postgresql.conf** customizations are now made to files in the **/etc/manageiq/postgresql.conf.d** directory due to the following **include_dir** line in the default configuration. The settings in these new files override ones in the **postgresql.conf** file.

```
include_dir = '/etc/manageiq/postgresql.conf.d`
```

Migrating to the Infrastructure Management appliance

Overview

This procedure describes the process of migrating to the Infrastructure Management appliance. This procedure does not necessarily include migration of all possible customer modifications, so it is recommended that you fully test any modifications before you migrate your environment.

Important:

- Read through all of the steps in this procedure before you begin the migration process.
- Infrastructure Management appliances require 12 GB memory. However, before you migrate your appliances, adjust resources in your environment to avoid performance issues.

You can classify the migration into three groups of appliances:

- VMDB appliance - An appliance with running workers that also hosts its own database that other appliances can also connect to.
- Non-VMDB appliance - An appliance with running workers that does not host a database. The appliance is connected to an external database.
- Dedicated database appliance - A Infrastructure Management appliance or non-Infrastructure Management virtual machine with no running workers on it: the appliance contains only a database for other appliances to connect to.

Migration workflow summary

In summary, the migration process follows this workflow:

1. Prepare appliances:

   Note: This step applies only for a multi-region Infrastructure Management environment. Remove all the subscriptions from the global region before you migrate the global database since the subscriptions must be dropped and re-created.

2. Back up the database of your appliance.
3. Migrate appliances:
    1. Provision new appliances to replace each appliance.
    2. On the database and nondatabase appliances, copy the GUID file from the version of the same appliance to the same location.
    3. On the database and nondatabase appliances, use the appliance console menu to stop the evmserverd service, manually remove the pre-configured database and key, and fetch the v2_key from an appliance.
    4. Copy the appliance database backup file and restore the database on each VMDB and dedicated database appliance.
    5. Load the new version of the pglogical library on the VMDB and dedicated database appliances.

       Note: This step applies only for a multi-region Infrastructure Management environment.

    6. Migrate the VMDB appliance databases and update the Automate Model.
    7. Restart **evmserverd** on the VMDB appliances.
4. Configure replication after the migration process is complete and appliances are running again.

Backing up current appliances

These steps do not affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

```
psql -d vmdb_production -c "SELECT pg_size_pretty(pg_database_size('vmdb_production'));"
```

2. If it's a multi-region environment, ensure you removed all subscriptions from the global database appliance.
3. Stop the **evmserverd** process.

   ```
   systemctl stop evmserverd
   ```

4. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

   ```
   pg_dumpall -c --if-exists | gzip > vmdb.pg.gz
   ```

   Notes:

   - This command creates a compressed text-based backup. **pg_dumpall** is a utility for writing out ("dumping") all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
   - When you run a **pg_dumpall**, the **pg_replication_slots** is not copied over causing the slots to become inactive. Also, the size of the global database appliance is reduced allowing for quicker backup and restore.
   - Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your file system available space by using the df command.
   - You must also provide backup for configuration files such as **/var/www/miq/vmdb/certs/v2_key**, **/var/www/miq/vmdb/config/database.yml**, and **/var/www/miq/vmdb/GUID**.
5. Back up the following files and take note of the appliance that each file comes from. These files are used to replace the old appliances in the database and for disaster recovery.

   - /var/www/miq/vmdb/GUID
6. During the upgrade, the iptables configuration file **/etc/sysconfig/iptables** is removed. If you changed the iptables configuration from the default, run the **iptables --list -n** command to see the current configuration. Run the following command to back up the iptables configuration:

   ```
   iptables-save > /etc/iptables.conf
   ```

   You can restore your iptables configuration file with the following command:

   ```
   iptables-restore < /etc/iptables.conf
   ```

   You can also add this command to **/etc/rc.local** to reload the rules at every restart.

Steps to follow on newly deployed Infrastructure Management 4.8.1 Appliances

Complete the following steps on your appliances as specified to migrate to Infrastructure Management.

Note: Some steps are run on certain appliances. Ensure you wait for each command to finish before you move to the next step.

1. Provision and start the new appliances.
2. On the VMDB and non-VMDB appliances, run the following commands as the root user:
   1. Copy the GUID file from the appliance.

      ```
      scp root@$IP_of_old_appliance:/var/www/miq/vmdb/GUID /var/www/miq/vmdb/
      ```

      Notes:

      - Complete these steps on VMDB appliances first. The non-VMDB appliances don't have a database to connect to if they're done first.
      - Complete this step only if you want to keep the same appliance/miq_servers records in the database and inherit that appliance's existing settings. If you plan on deploying new appliances with new IP addresses and hostname and not inherit the settings, you might prefer creating new records by not copying the GUID.
   2. Stop the evmserverd process.

      ```
      systemctl stop evmserverd
      ```

   3. Remove the previously generated encryption key.

      ```
      rm /var/www/miq/vmdb/certs/v2_key
      ```

   4. Copy the default database configuration file, ovewriting the previously generated one that used the deleted encryption key.

      ```
      scp root@<IP_of_old_appliance>:/var/www/miq/vmdb/config/database.yml /var/www/miq/vmdb/config/database.yml
      ```

   5. Drop the previously generated database.

      ```
      dropdb vmdb_production
      ```

   6. For VMDB appliances, configure the application, fetch the key from remote system, and Reset Configured Database.
   7. For VMDB appliances, select the same region number and database password as the old installation. You can find the region number that is used for an installation that is listed as **Database/Region:** in the **appliance_console** summary information.
   8. For non-VMDB appliances, configure the application by using the join region in external database option. Note as part of the database configuration step, fetch the **v2_key** from the appliances.
   9. Allow the appliance to start.
3. On the new VMDB and new dedicated database appliances:
   1. Stop the evmserverd process.

```
systemctl stop evmserverd
```

2. Drop the default database.

   ```
   dropdb vmdb_production
   ```

3. Copy the database dump from the appliance. Update the following location to reflect the source file location of the database dump and copy it to a location with enough available space.

   ```
   scp root@$IP_of_old_appliance:/var/tmp/vmdb.pg.gz /var/tmp
   ```

4. Decompress and restore the database dump backup file.

   ```
   gunzip < vmdb.pg.gz | psql postgres
   ```

5. Change to the vmdb directory.

   ```
   cd /var/www/miq/vmdb/
   ```

6. Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you run into an error, contact IBM Support.

   ```
   rake db:migrate
   ```

4. Update the Automate Model to the latest version. This resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance.

   ```
   rake evm:automate:reset
   ```

5. Start the evmserverd process. In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance:

   ```
   systemctl start evmserverd
   ```

The migration process is now complete. Once appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Infrastructure Management

Infrastructure Automation – Infrastructure Management delivers the insight, control, and automation enterprises need to address the challenges of managing virtual environments, which are far more complex than physical ones. This technology enables enterprises with existing virtual infrastructures to improve visibility and control, and those just starting virtualization deployments to build and operate a well-managed virtual infrastructure.

Infrastructure Management includes the following feature sets:

- Insight: discovery, monitoring, utilization, performance, reporting, analytics, chargeback, and trending.
- Control: security, compliance, alerting, policy-based resource and configuration management.
- Automate: IT process, task and event, provisioning, workload management and orchestration.
- Integrate: systems management, tools and processes, event consoles, CMDB, RBA, and web services.

Note: When the Infrastructure Management component of Infrastructure Automation is installed, you can access the Infrastructure Management UI from the IBM Cloud Pak for AIOps console UI by browsing to the menu: **Automate Infrastructure > Infrastructure management**. The documentation for Infrastructure Management, including the menu navigation for documented procedures, is written from the context of first browsing to the Infrastructure Management UI.

This documentation is delivered by various sources in the ManageIQ open source community. IBM accepts support tickets, and corrects documentation errors. Support and enhancement updates can also come from the open source community.

Use the following sections to install, configure and manage your virtual infrastructure. View and collect information from your clusters, hosts, virtual machines, and other resources.

- Installing
- Administration
- Managing infrastructure and Inventory
- Managing Providers
- Creating a Service in for Virtual Machine Provisioning
- Provisioning Virtual Machines and Instances
- Managing controls
- Managing integrations
- APIs
- Reference
- Troubleshooting

Infrastructure Management - Administration

Use the following resources to perform administration tasks for your deployment of IBM Infrastructure Management.

- Deployment planning guide
- High availability guide

- Appliance hardening guide
- Enable terraform-runner in an air-gapped environment

- Deployment Planning Guide
  - Introduction to Infrastructure Automation
    - Architecture
    - Requirements
      - Virtual Hardware Requirements
      - Database Requirements
      - Browser Requirements
      - Additional Requirements
    - Terminology
    - Regions
    - Roles
      - Appliance Types
    - Centralized Administration
    - Tenancy
    - Using a Load Balancer
      - Configuring the Load Balancer
      - Configuring Worker Appliances for Load Balancing
    - Database Configuration
      - User Overrides
      - Reading Configuration Settings
  - Capacity Planning
    - Capacity and Utilization Collection
    - Assigning the Capacity and Utilization Server Roles
    - Adding Database Credentials for Data Collection
    - Data Collection for Red Hat Virtualization
    - Data Collection for Red Hat Enterprise Linux OpenStack Platform
    - Capacity and Utilization Data Collected
      - Capacity and Utilization Charts for Hosts, Clusters, and Virtual Machines
      - Capacity and Utilization Charts for Datastores
    - Capacity and Utilization Chart Features
      - Zooming into a Chart
      - Drilling into Chart Data
    - Planning Where to Put a New Virtual Machine
- Appendix
  - Example PostgreSQL Configuration File

Deployment Planning Guide

Introduction to Infrastructure Automation

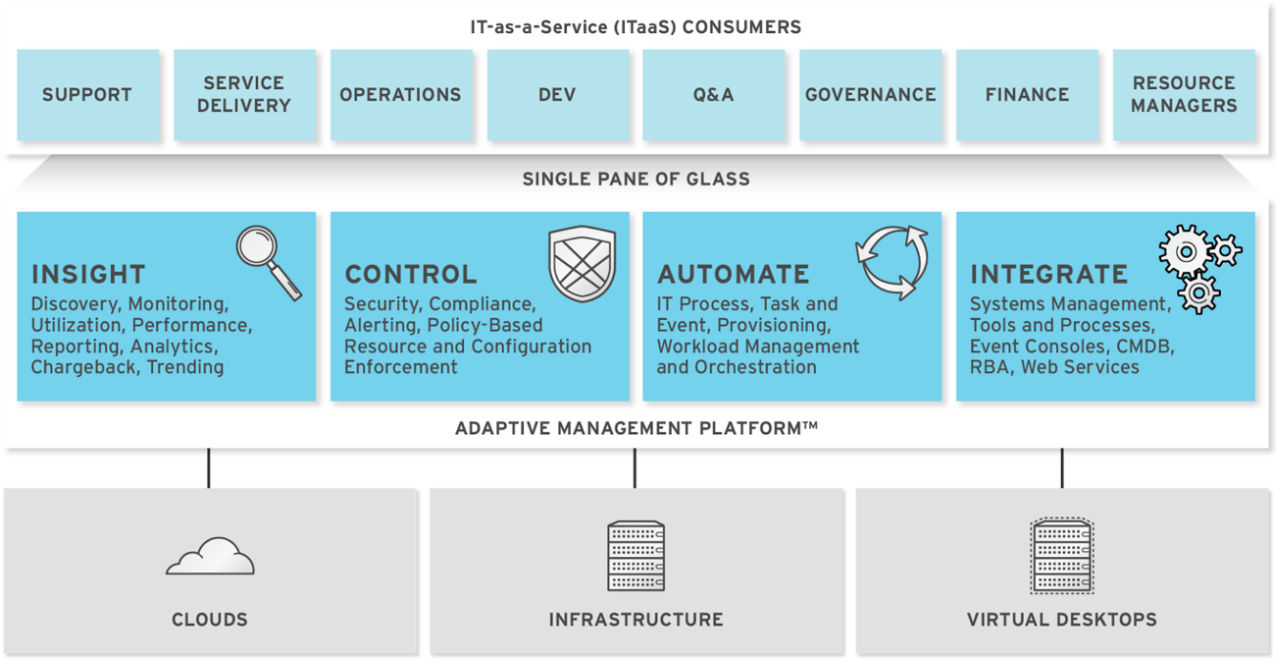
Infrastructure Automation delivers the insight, control, and automation that enterprises need to address the challenges of managing virtual environments. This technology enables enterprises with existing virtual infrastructures to improve visibility and control, and those starting virtualization deployments to build and operate a well-managed virtual infrastructure.

Infrastructure Automation provides the following feature sets:

- Insight: Discovery, Monitoring, Utilization, Performance, Reporting, Analytic, Chargeback, and Trending.
- Control: Security, Compliance, Alerting, and Policy-Based Resource, and Configuration Enforcement.
- Automate: IT Process, Task and Event, Provisioning, and Workload Management and Orchestration.
- Integrate: Systems Management, Tools and Processes, Event Consoles, Configuration Management Database (CMDB), Role-based Administration (RBA), and Web Services.

Architecture

The diagram below describes the capabilities of Infrastructure Automation. Its features are designed to work together to provide robust management and maintenance of your virtual infrastructure.



The architecture comprises the following components:

- The Infrastructure Automation appliance (appliance) which is supplied as a secure, high-performance, preconfigured virtual machine. It provides support for HTTPS communications.
- The Infrastructure Automation Server (Server) resides on the appliance. It is the software layer that communicates between the SmartProxy and the Virtual Management Database. It includes support for HTTPS communications.
- The Virtual Management Database (VMDB) resides either on the appliance or another computer accessible to the appliance. It is the definitive source of intelligence collected about your Virtual Infrastructure. It also holds status information regarding appliance tasks.
- The Infrastructure Automation Console (Console) is the Web interface used to view and control the Server and appliance. It is consumed through Web 2.0 mash-ups and web services (WS Management) interfaces.
- The SmartProxy can reside on the appliance or on an ESX Server. If not embedded in the Server, the SmartProxy can be deployed from the appliance. A SmartProxy agent must configured in each storage location, and must be visible to the appliance. The SmartProxy acts on behalf of the appliance communicating with it over HTTPS on standard port 443.

Requirements

To use Infrastructure Automation, certain virtual hardware, database, and browser requirements must be met in your environment.

Virtual Hardware Requirements

The Infrastructure Automation appliance requires the following virtual hardware at minimum:

- 4 VCPUs
- 12 GB RAM
- 44 GB HDD + optional database disk + optional messaging disk

Database Requirements

Red Hat recommends allocating the virtual machine disk fully at the time of creation. Three main factors affect the size of your database over time:

- Virtual Machine Count: the most important factor in the calculation of virtual machine database (VMDB) size over time.
- Host Count: the number of hosts associated with the provider.
- Storage Count: the number of individual storage elements as seen from the perspective of the provider or host. It is not the total number of virtual disks for all virtual machines.

Use the following table as a guideline to calculate minimum requirements for your database:

			Estimated VMDB Size in GB	
Virtual Machine Count	Host Count	Storage Count	1 year	2 year
100	5	50	3.5	5
500	10	100	17	25
5000	50	500	173	251

Note:

When enabling capacity and utilization for metrics gathering over a period of time, it is recommended that the VMDB size scale accordingly. Evaluate the number of instances in your provider inventory and storage duration requirements to plan for increased VMDB sizing requirements.

Use the following information to plan for your increased VMDB needs when working with metrics gathering:

- Realtime metrics data are stored for 4 hours.
- Rollup metrics data are stored for 6 months.

Example:

	Minute	**Hour**	**Day**
OpenStack Provider Instance	**3** Realtime Metrics	**181** (3 records * 60 minutes = 180 Realtime Metrics + 1 hourly Rollup Metric)	**4,345** (3 records * 60 minutes * 24 hours =4320 Realtime Metrics + 1 daily Rollup Metric)

- Metrics data storage times can be configured by editing the Advanced Settings.

Browser Requirements

To use Infrastructure Automation, the following browser requirements must be met:

- One of the following web browsers:
  - Google Chrome
  - Mozilla Firefox
  - Safari
  - Internet Explorer 10 or higher

Additional Requirements

Additionally, the following must be configured to use Infrastructure Automation:

- The Infrastructure Automation appliance must already be installed and activated in your enterprise environment.
- The SmartProxy must have visibility into the virtual machines and cloud instances that you want to control.
- For more information, see SmartProxies in the Infrastructure Automation *General Configuration* guide.

Terminology

The following terms are used throughout the documentation. Review them before proceeding.

- Account Role The level of access a user has to different parts and functions of the Infrastructure Automation console. There are a variety of Account Roles, which can be assigned to users to restrict or allow access to parts of the console and virtual infrastructure.
- Action An execution that is performed after a condition is evaluated.
- Alert Infrastructure Automation alerts notify administrators and monitoring systems of critical configuration changes and threshold limits in the virtual environment. The notification can take the form of either an email or an SNMP trap.
- Analysis Profile A customized scan of hosts, virtual machines, or instances. You can collect information from categories, files, event logs, and registry entries.
- Cloud A pool of on-demand and highly available computing resources. The usage of these resources are scaled depending on the user requirements and metered for cost.
- Infrastructure Automation Appliance A virtual machine where the virtual management database (VMDB) and Infrastructure Automation reside.
- Infrastructure Automation Console A web-based interface into the Infrastructure Automation appliance.
- Infrastructure Automation Role A designation assigned to a Infrastructure Automation server that defines what a Infrastructure Automation server can do.
- Infrastructure Automation Server The application that runs on the Infrastructure Automation appliance and communicates with the SmartProxy and the VMDB.
- Cluster Hosts that are grouped together to provide high availability and load balancing.

- Condition A control policy test triggered by an event, which determines a subsequent action.
- Discovery Process run by the Infrastructure Automation server which finds virtual machine and cloud providers.
- Drift The comparison of a virtual machine, instance, host, cluster to itself at different points in time.
- Event A trigger to check a condition.
- Event Monitor Software on the Infrastructure Automation appliance which monitors external providers for events and sends them to the Infrastructure Automation server.
- Host A computer running a hypervisor, capable of hosting and monitoring virtual machines. Supported hypervisors include RHV-H, VMware ESX hosts, Windows Hyper-V hosts.
- Instance/Cloud Instance A on-demand virtual machine based upon a predefined image and uses a scalable set of hardware resources such as CPU, memory, networking interfaces.
- Managed/Registered VM A virtual machine that is connected to a host and exists in the VMDB. Also, a template that is connected to a provider and exists in the VMDB. Note that templates cannot be connected to a host.
- Managed/Unregistered VM A virtual machine or template that resides on a repository or is no longer connected to a provider or host and exists in the VMDB. A virtual machine that was previously considered registered may become unregistered if the virtual machine was removed from provider inventory.
- Provider An external management system that Infrastructure Automation integrates in order to collect data and perform operations.
- Policy A combination of an event, a condition, and an action used to manage a virtual machine.
- Policy Profile A set of policies.
- Refresh A process run by the Infrastructure Automation server which checks for relationships of the provider or host to other resources, such as storage locations, repositories, virtual machines, or instances. It also checks the power states of those resources.
- Regions A region is the collection of zones that share the same database for reporting and charting. A master region may be added to synchronize multiple VMDBs into one VMDB for higher-level reporting, providing a “single pane of glass” view.
- Resource A host, provider, instance, virtual machine, repository, or datastore.
- Resource Pool A group of virtual machines across which CPU and memory resources are allocated.
- Repository A place on a datastore resource which contains virtual machines.
- SmartProxy The SmartProxy is a software agent that acts on behalf of the Infrastructure Automation appliance to perform actions on hosts, providers, storage and virtual machines.

- The SmartProxy can be configured to reside on the Infrastructure Automation appliance or on an ESX server version. The SmartProxy can be deployed from the Infrastructure Automation appliance, and provides visibility to the VMFS storage. Each storage location must have a SmartProxy with visibility to it. The SmartProxy acts on behalf of the Infrastructure Automation appliance. If the SmartProxy is not embedded in the Infrastructure Automation server, it communicates with the Infrastructure Automation appliance over HTTPS on standard port 443.

- SmartState Analysis Process run by the SmartProxy which collects the details of a virtual machine or instance. Such details include accounts, drivers, network information, hardware, and security patches. This process is also run by the Infrastructure Automation server on hosts and clusters. The data is stored in the VMDB.
- SmartTags Descriptors that allow you to create a customized, searchable index for the resources in your clouds and infrastructure.
- Storage Location A device, such as a VMware datastore, where digital information resides that is connected to a resource.
- Tags Descriptive terms defined by a Infrastructure Automation user or the system used to categorize a resource.
- Template A template is a copy of a preconfigured virtual machine, designed to capture installed software and software configurations, as well as the hardware configuration, of the original virtual machine.
- Unmanaged Virtual Machine Files discovered on a datastore that do not have a virtual machine associated with them in the VMDB. These files may be registered to a provider that the Infrastructure Automation server does not have configuration information on. Possible causes may be that the provider has not been discovered or that the provider has been discovered, but no security credentials have been provided.
- Virtual Machine A software implementation of a system that functions similar to a physical machine. Virtual machines utilize the hardware infrastructure of a physical host, or a set of physical hosts, to provide a scalable and on-demand method of system provisioning.
- Virtual Management Database (VMDB) Database used by the Infrastructure Automation appliance to store information about your resources, users, and anything else required to manage your virtual enterprise.
- Virtual Thumbnail An image in the web interface representing a resource, such as a provider or a virtual machine, showing the resource’s properties at a glance. Each virtual thumbnail is divided into quadrants, which provide information about the resource, such as its software and power state.
- Worker Appliance A Infrastructure Automation appliance dedicated to a role other than user interface or database.
- Zones Infrastructure Automation Infrastructure can be organized into zones to configure failover and to isolate traffic. Zones can be created based on your environment. Zones can be based on geographic location, network location, or function. When first started, new servers are put into the default zone. ## Planning

This guide provides some general guidelines to planning a deployment on Infrastructure Automation. This includes creating multiple regions containing Infrastructure Automation appliances, CPU sizing recommendations, database sizing recommendations, and database configuration.

Regions

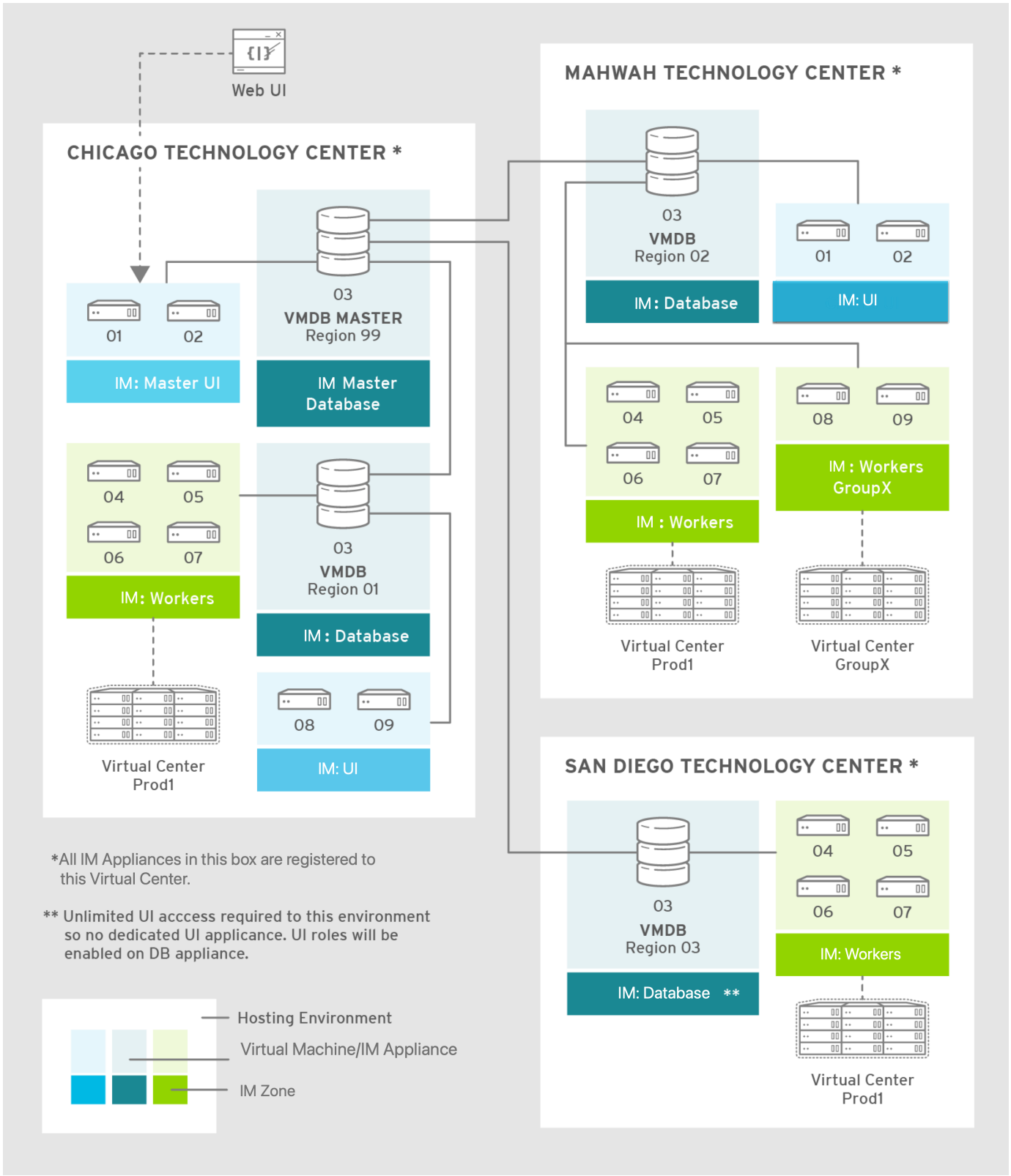
Regions are used for centralizing data which is collected from public and private virtualization environments. A region is ultimately represented as a single database for the VMDB. Regions are particularly useful when multiple geographical locations need to be managed, as they enable all the data collection to happen at each particular location and avoid data collection traffic across slow links between networks.

When multiple regions are being used, each with their own unique ID, a master region can be created to centralize the data of all the children regions into a single master database. To do this, configure each child region to replicate its data to the master region database (the recommended region is 99, though any number up to three digits will work). This parent and child region is a one-to-many relationship.

Regions can contain multiple zones, which in turn contain appliances. Zones are used for further segregating network traffic along with enabling failover configurations. Each appliance has the capability to be configured for a number of specialized server roles. These roles are limited to the zone containing the appliance they run on.

Only one failover type of each server role can run in a zone. If multiple appliances have the same failover role, the extras are used as backups that activate only if the primary appliance fails. Non-failover server roles can run on multiple appliances simultaneously in a zone, so resources can be adjusted according to the workload those roles are responsible for.

The following diagram demonstrates an example of the multiple regions working together in a Infrastructure Automation environment.

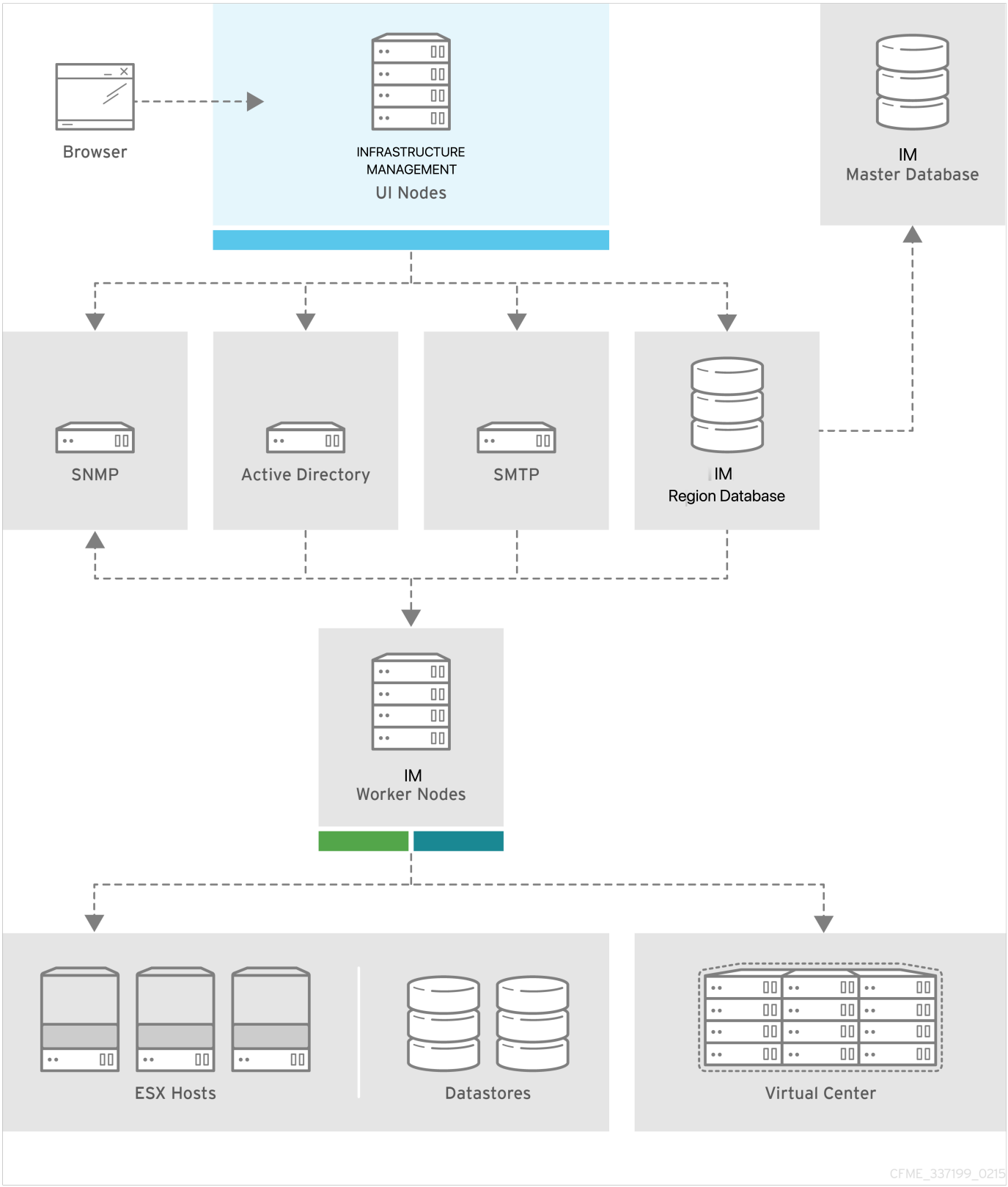


The master appliance is located in Chicago and contains a master region and a subregion that manages the worker appliances. The Mahwah technology center contains a single subregion that manages two zones. Likewise, the San Diego technology center contains a single subregion managing a single zone.

Note:

- Replicating a parent region to a higher-level parent is not supported.
- Parent regions can be configured after the child regions are online.

The following diagram provides a closer look at a region:



In this region, we have several Infrastructure Automation appliances acting as UI nodes and worker nodes. These worker nodes execute tasks on the providers in your environment. The region also uses a region database that reports to a master database on the main Infrastructure Automation appliance. All appliances can connect to the authentication services (Active Directory, LDAP, Identity Management), outgoing mail (SMTP), and network services (SNMP).

Note:

Infrastructure Automation can be configured in a highly available setup. In this case, all PostgreSQL instances must be running on a server that is deployed from the Infrastructure Automation appliance. High availability is achieved by database replication between two or more database servers.

Roles

Server roles define what a server can do. Assigning different server roles to appliances can allow them to focus on specific functions. When planning a deployment, consider which roles to assign to each appliance. Some server roles are enabled by default in Infrastructure Automation. Many server roles start worker processes.

Some roles are also dependent on other roles. For example, because the Infrastructure Automation user interface relies on the API for access, the Web Services role must be enabled with the User Interface role for users to log in to the appliance. See Server Roles in *General Configuration* for details on each server role and its function.

Appliance Types

Depending on the needs of your environment, you may choose to separate worker and database tasks between appliances. One example of this is to implement a highly available configuration so that certain appliances are running the PostgreSQL database and providing failover. For more details about configuring high availability, see the High Availability Guide.

The following provides a summary of types of appliances:

Appliance Type	Database	Workers	Description
VMDB appliance	Yes	Yes	Worker processes are running, and it also hosts its own database that other appliances can connect to.
Non-database appliance	No	Yes	Worker processes are running on the appliance, but it does not host a database. The appliance is connected to an external database.
Dedicated-database appliance	Yes	No	This appliance contains no worker processes, only a database for other appliances to connect to.
Non-Infrastructure Automation VM with database	Yes	No	This appliance contains no worker processes, only a database. As this is not a Infrastructure Automation appliance, you cannot run any Infrastructure Automation `rake` tasks on it. This appliance must be migrated using a non-database appliance that is pointed at it, using it as a database.

Appliance types

Centralized Administration

Infrastructure Automation includes centralized administration capabilities, where certain operations can be initiated from the global region and processed and executed on remote regions. From the global region, you can also access the user interfaces of virtual machines residing in remote regions.

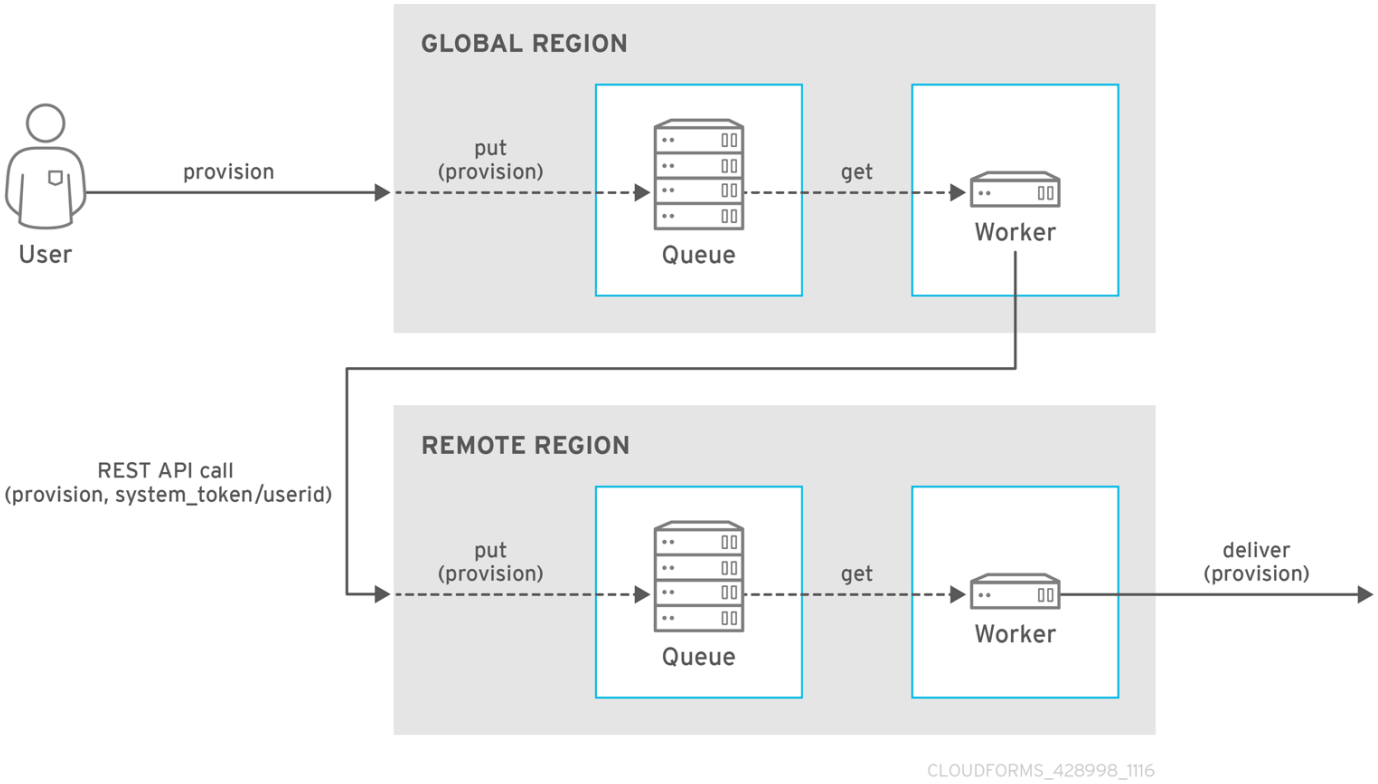
The following life cycle management operations can be initiated from the global region using centralized administration:

- Virtual machine provisioning
- Virtual machine power operations
- Virtual machine retirement
- Virtual machine reconfiguration
- Service provisioning
- Service retirement
- Opening a virtual machine in the remote region

Note:

Infrastructure Automation life cycle operations other than those listed above are not supported. Centralized administration capabilities are not supported from the Self Service user interface.

With centralized administration, the remote_queue_put leverages a new system-to-system REST API request to forward the original request to the remote region. This request is put in the local queue in the remote region, which is then delivered by a worker in the remote region as if it was queued there all along. As a result, a Infrastructure Automation operator in the global region can be seen as provisioning on behalf of a remote region.



Note:

The operations initiated from the global region are subject to the role-based access control (RBAC) rules on the remote region. The user in the remote region which matches the logged-in user's **user ID** will be used to enforce RBAC in the target region. The operation will fail on the remote system if the user does not have the required permissions.

Tenancy

Infrastructure Automation supports multitenancy. Tenants can be totally separate or they can be in a parent-child or peer relationship. Tenants in a relationship can share or inherit a certain configuration. You can subdivide and create child tenants and they, in turn, can have child tenants, and so on. The ability to have multi-level (nested) tenants in a hierarchy enables those at the bottom to inherit permissions from those above. This configuration allows for granular user permissions to be set on specific tenants.

A tenant can also contain a self-contained child tenant known as a 'project'. A project cannot have a child tenant, but is useful for allocating resources to a small group or team within a larger organization.

Note:

If you do not add any additional tenants, all resources and user accounts are contained in a single base tenant which is your Infrastructure Automation appliance itself. In Infrastructure Automation, is sometimes referred as 'tenant zero'.

Tenancy Account Roles.

In Infrastructure Automation, the following two account roles are associated with tenancy:

- Tenant administrator
- Tenant quota administrator

Tenant administrator and tenant quota administrator roles are like administrator and super administrator. These roles are not limited to the tenant upon which they are acting and act across all tenants, and therefore should be considered privileged users. These are not roles inside a tenant.
Tenancy Models.

The following two approaches exist for tenancy planning:

- **Tenantless** - You can create a single large tenant, sometimes referred as 'tenant zero', and perform all your operations in there without any subdivision of resources or user accounts.
- **Enterprise model** - A common scenario is to create a single tenant, and then subdivide it based on the structures or departments within your organization. Those departments are then able to further subdivide their resources into distinct projects. With this model, you have a single URL for user access, while still having the ability to divide resources into nested hierarchical tenants.

Tenancy Configuration.

You can create and configure tenancy using the Infrastructure Automation user interface in the same place you set up users, groups and roles by selecting **Configuration** from the settings menu, and then clicking on the **Access Control** accordion.

Tenancy in Automation.

One of the features of tenancy is that each tenant can have its own automate domain. Tenant-based domains can help in several use cases, such as if you have:

- groups that need their own naming routines
- varying types of approval needs
- departments that use different end ticketing systems
- a customer who is a holding company or centralized IT organization for managing different business units

Just like standard domains are nested, you can also add automate domains that are nested at the tenant level.

Tenancy Quota and Reporting.

You can allocate and enforce quotas for the following attributes:

- Virtual CPUs
- Memory in GB
- Storage in GB
- Number of virtual machines
- Number of templates

You can generate or schedule a report for **Tenant Quotas** similar to other reports.

Note:

Currently, in tenant quota reports you will see all of the tenants but there is no nesting information available by parent and child tenants.

Example:

In the following example of a tenant quota report, *DevOps Teams* is a parent tenant and *Team Alpha* and *Team Bravo* are child tenants.

Tenant Name	Quota Name	Total Quota	In Use	Allocated	Available
DevOps Teams	Allocated Virtual CPUs	125 Count	0 Count	75 Count	50 Count
DevOps Teams	Allocated Memory in GB	255.0 GB	0.0 GB	125.0 GB	130.0 GB
DevOps Teams	Allocated Storage in GB	1505.0 GB	0.0 GB	1100.0 GB	405.0 GB
DevOps Teams	Allocated Number of Templates	15 Count	0 Count	4 Count	11 Count
DevOps Teams	Allocated Number of Virtual Machines	160 Count	0 Count	70 Count	90 Count
DevOps Teams					
Team Alpha	Allocated Virtual CPUs	50 Count	0 Count	0 Count	50 Count
Team Alpha	Allocated Memory in GB	75.0 GB	0.0 GB	0.0 GB	75.0 GB
Team Alpha	Allocated Storage in GB	1000.0 GB	0.0 GB	0.0 GB	1000.0 GB
Team Alpha	Allocated Number of Virtual Machines	50 Count	0 Count	0 Count	50 Count
Team Alpha	Allocated Number of Templates	3 Count	0 Count	0 Count	3 Count
Team Alpha					
Team Bravo	Allocated Virtual CPUs	25 Count	0 Count	0 Count	25 Count
Team Bravo	Allocated Memory in GB	50.0 GB	0.0 GB	0.0 GB	50.0 GB
Team Bravo	Allocated Storage in GB	100.0 GB	0.0 GB	0.0 GB	100.0 GB
Team Bravo	Allocated Number of Virtual Machines	20 Count	0 Count	0 Count	20 Count
Team Bravo	Allocated Number of Templates	1 Count	0 Count	0 Count	1 Count
Team Bravo					

- Total Quota: Total quota enforced per attribute for a tenant
- In Use: Amount of quota currently in use by tenants
- Allocated: Amount of quota given to all child tenants
- Available: *Total Quota* minus (-) *In Use* minus (-) *Allocated*

Tenancy Chargeback.

You have the ability to do tenancy in chargeback where you are able to assign rates and have a different rate for each tenant. You can make use of the default rate or create your own set of rates depending on the tenant. As well, there is an ability to create chargeback reports by tenant.

Tenancy Service Catalogs.

Similar to automate domains, you can have service catalogs at each level of tenancy. Once you add a service catalog at a particular level of tenancy, it is visible to that tenant and its children (unless you use tagging to exclude).

Tenancy Providers.

Providers can be added at any level of tenancy. Once added, a provider is visible to any child or lower tenants, making it possible to easily separate resources that are owned or accessed by one group, and should not be available to other tenants.

Tenancy Relationships and Properties.

The tenant summary page found in **Configuration > Access Control > Tenants > Tenant** provides detailed information about the tenant and its relationships including:

- Catalog items and bundles
- Automate domains
- Provider relationships

Using a Load Balancer

Deploying multiple user interface worker appliances and placing them behind a third-party load balancer allows for redundancy and improved performance. This requires extra configuration in both the load balancer and in the Infrastructure Automation user interface worker appliances.

Configuring the Load Balancer

- Configure the load balancer to use sticky sessions. This ensures that when a session is started, all requests for that session are sent to the same worker appliance.
- Configure the load balancer to test for connectivity using the Infrastructure Automation ping response page: **`https://appliance_name/ping`**. The expected reply from the appliance is the text string *pong*. Using this URL is preferable to the appliance login URL as it does not establish a connection to the database.

Configuring Worker Appliances for Load Balancing

When using a load balancer, configure appliances that have the **User Interface** role enabled to store session data in the database. As a result, the user does not need to re-login if the load balancer redirects them to an alternative server in the case the original user interface worker is unresponsive.

On each appliance, configure the session data storage location using the **`session_store`** parameter within the advanced settings page in the Infrastructure Automation user interface:

1. Click  (**Configuration**).
2. Click the **Advanced** tab.
3. Change the **`session_store`** parameter to **`sql`** in the following line (the default parameter is **`cache`**):

   ```
   :server:
   ...
    :session_store: sql
   ```

4. Click **Save**.

Configure the `session_store` parameter to point to `sql` on each user interface appliance behind the load balancer.

Database Configuration

This section describes the Infrastructure Automation PostgreSQL database configuration. The below table provides information on each file: its location, primary function, and notes regarding behavior or recommendations.

File	Location	Description	Note
`postgresql.conf`	Data Directory	Default server configuration	Adds directive to include **`/etc/manageiq/postgresql.conf.d`** directory
`postgresql.auto.conf`	Data Directory	Contains configuration set by the **`ALTER SYSTEM`** command	**Do not edit manually**
`01_miq_overrides.conf`	**`/etc/manageiq/postgresql.conf.d`**	Contains Infrastructure Automation default overrides	Overwritten on upgrades
<other files>	**`/etc/manageiq/postgresql.conf.d`**	Contains user overrides	Takes precedence if alphabetically after **`01_miq_overrides.conf`**

Database files

User Overrides

Store custom configurations, or user overrides, in **`/etc/manageiq/postgresql.conf.d`**. Name the user override file so that it follows **`01_miq_overrides.conf`** alphabetically. This ensures custom configurations are not overwritten on Infrastructure Automation upgrades.

The following file name example follows **`01_miq_overrides.conf`** alphabetically in the **`/etc/manageiq/postgresql.conf.d`** directory:

Example:

`test_miq_overrides.conf`

Reading Configuration Settings

Query the Infrastructure Automation PostgreSQL database directly to read configuration settings. See the PostgreSQL Documentation for more information.

The following example queries the Infrastructure Automation for current value set for **`max_wal_senders`**:

Example:

```
`psql -d vmdb_production -c 'show max_wal_senders'
```

Capacity Planning

Capacity and Utilization Collection

Infrastructure Automation server can collect and analyze capacity and utilization data from your virtual infrastructure. Use this data to understand the limitations of your current environment and plan for growth.

For some capacity and utilization data, Infrastructure Automation calculates and shows trend lines in the charts. Trend lines are created by using linear regression, which is calculated by using the capacity and utilization data that is collected by Infrastructure Automation during the interval you specify for the chart. The more data that you have the better the predictive value of the trend line.

Three server roles are associated with the collection and metric creation of capacity and utilization.

- The Capacity & Utilization Coordinator role checks to see whether it is time to collect data, somewhat like a scheduler. If it is time, a job is queued for the Capacity & Utilization Data Collector. The Coordinator role is required to complete capacity and utilization data collection. If more than one server in a specific zone has this role, only one is active at a time.
- The Capacity & Utilization Data Collector performs the actual collection of capacity and utilization data. This role has a dedicated worker, and there can be more than one server with this role in a zone.
- The Capacity & Utilization Data Processor processes all of the data that is collected, allowing Infrastructure Automation to create charts. This role has a dedicated worker, and there can be more than one server with this role in a zone.

Assigning the Capacity and Utilization Server Roles

1. Click **Configuration**, then select the server to configure from menu: **Settings > Zone** in the accordion menu on the left.
2. Browse to the **Server Roles** list in the menu: **Server > Server Control** section. From there, set the required capacity and utilization roles to **ON**, namely:
    1. **Capacity & Utilization Coordinator**
    2. **Capacity & Utilization Data Collector**
    3. **Capacity & Utilization Data Processor**
3. Click **Save**.

Data collection is enabled immediately. However, the first collection begins 5 minutes after the server is started, and every 10 minutes after that. Therefore, the longest the collection takes after enabling the Capacity & Utilization Collector role is 10 minutes. The first collection from a particular provider may take a few minutes since Infrastructure Automation is gathering data points going one month back in time.

Note:

In addition to setting the server role, you must also select which clusters and data stores to collect data for. For more information, see the **General Configuration** guide. You must have super administrator privileges to edit these settings.

Adding Database Credentials for Data Collection

After you create the new user, add the user's credentials to the settings for the provider.

1. From menu: **Compute > Infrastructure > Providers**, select an infrastructure provider to update its settings.
2. Click **Configuration**, and then **Edit Selected Infrastructure Provider**.
3. In the **Credentials** area, click **C & U Database**.
4. Type in the credentials for the new database user you created.
5. Click **Save**.
6. Restart the Capacity and Utilization Data Collector.

Data Collection for Red Hat Virtualization

To collect capacity and utilization data for Red Hat Virtualization (RHV), you must add a user to the RHV-M history database for Infrastructure Automation to use.

Perform this procedure on the PostgreSQL server where the history database (ovirt_engine_history) is located. Usually, this is the RHV-M server.

1. Using SSH, access the RHV-M database server as the root user:

    ```
    $ ssh root@example.postgres.server
    ```

2. Switch to the postgres user:

```
# su - postgres
```

For RHV 4.2 and 4.3, the PostgreSQL database is delivered as a software collection in version 9.5 and must be enabled first. Therefore, to run the following psql commands, you need to enable the *rh-postgresql95* collection and load into the current shell prompt by using the source command: $ source /opt/rh/rh-postgresql95/enable

3. Create the user for Infrastructure Automation to be granted read-only access to the history database:

```
$ psql -U postgres -c "CREATE ROLE example_role_name WITH LOGIN ENCRYPTED PASSWORD '[password]';" -d
ovirt_engine_history
```

4. Grant the newly created user permission to connect to the history database:

```
$ psql -U postgres -c "GRANT CONNECT ON DATABASE ovirt_engine_history TO example_role_name;"
```

5. Grant the newly created user usage of the public schema:

```
$ psql -U postgres -c "GRANT USAGE ON SCHEMA public TO example_role_name;" ovirt_engine_history
```

6. Generate the rest of the permissions that will be granted to the newly created user and save them to a file:

```
$ psql -U postgres -c "SELECT 'GRANT SELECT ON ' || relname || ' TO example_role_name;' FROM pg_class JOIN
pg_namespace ON pg_namespace.oid = pg_class.relnamespace WHERE nspname = 'public' AND relkind IN ('r', 'v', 'S');" --
pset=tuples_only=on  ovirt_engine_history > grant.sql
```

7. Use the file that you created in the previous step to grant permissions to the newly created user:

```
$ psql -U postgres -f grant.sql ovirt_engine_history
```

8. Remove the file that you used to grant permissions to the newly created user:

```
$ rm grant.sql
```

9. Exit to the RHV-M database server prompt:

```
$ exit
```

10. Update the server's firewall to accept TCP communication on port 5432:

```
# firewall-cmd --add-port=5432/tcp --permanent
```

11. Enable external md5 authentication.

    1. For RHV 4.0, 4.1, and 4.4 update the following line in **/var/lib/pgsql/data/pg_hba.conf**:

       ```
       host    all      all    0.0.0.0/0     md5
       ```

    2. For RHV 4.2 and 4.3, update the following line in **/var/opt/rh/rh-postgresql95/lib/pgsql/data/pg_hba.conf**:

       ```
       host    all      all    0.0.0.0/0     md5
       ```

12. Enable PostgreSQL to listen for remote connections.

    1. For RHV 4.0, 4.1, and 4.4 ensure the **listen_addresses** line in **/var/lib/pgsql/data/postgresql.conf** matches the following example:

       ```
       listen_addresses  =  '*'
       ```

    2. For RHV 4.2 and 4.3, ensure the **listen_addresses** line in **/var/opt/rh/rh-postgresql95/lib/pgsql/data/postgresql.conf** matches the following example:

       ```
       listen_addresses  =  '*'
       ```

13. Reload the PostgreSQL configuration.

    1. For RHV 4.0, 4.1, and 4.4:

       ```
       # systemctl reload postgresql
       ```

    2. For RHV 4.2 and 4.3:

       ```
       # systemctl reload rh-postgresql95-postgresql
       ```

Data Collection for Red Hat Enterprise Linux OpenStack Platform

Before you can collect data from a Red Hat Enterprise Linux OpenStack Platform (RHEL-OSP) provider, you must install Ceilometer and configure it to accept queries from external systems.

These instructions require a Red Hat Enterprise Linux 6.4 @base installation of RHEL-OSP and registration to a satellite that has access to both the **RHEL-OSP** and **RHEL Server Optional** channels. Perform all steps on your RHEL-OSP system.

1. Add the required channels and update your system:

```
# rhn-channel --add -c rhel-x86_64-server-6-ost-3 -c rhel-x86_64-server-optional-6
# yum update -y
# reboot
```

2. Install **Ceilometer**:

```
# yum install *ceilometer*
```

3. Install and start the MongoDB store:

```
# yum install mongodb-server
# sed -i '/--smallfiles/!s/OPTIONS=\"/OPTIONS=\"--smallfiles /' /etc/sysconfig/mongod
# service mongod start
```

4. Create the following users and roles:

```
# SERVICE_TENANT=$(keystone tenant-list | grep services | awk '{print $2}')
# ADMIN_ROLE=$(keystone role-list | grep ' admin ' | awk '{print $2}')
# SERVICE_PASSWORD=servicepass
# CEILOMETER_USER=$(keystone user-create --name=ceilometer \
--pass="$SERVICE_PASSWORD" \
--tenant_id $SERVICE_TENANT \
--email=ceilometer@example.com | awk '/ id / {print $4}')
# RESELLER_ROLE=$(keystone role-create --name=ResellerAdmin | awk '/ id / {print $4}')
# ADMIN_ROLE=$(keystone role-list | awk '/ admin / {print $2}')
# for role in $RESELLER_ROLE $ADMIN_ROLE ; do
keystone user-role-add --tenant_id $SERVICE_TENANT \
--user_id $CEILOMETER_USER --role_id $role
done
```

5. Configure the authtoken in **ceilometer.conf**:

```
# openstack-config --set /etc/ceilometer/ceilometer.conf keystone_authtoken auth_host 127.0.0.1
# openstack-config --set /etc/ceilometer/ceilometer.conf keystone_authtoken auth_port 35357
# openstack-config --set /etc/ceilometer/ceilometer.conf keystone_authtoken auth_protocol http
# openstack-config --set /etc/ceilometer/ceilometer.conf keystone_authtoken admin_tenant_name services
# openstack-config --set /etc/ceilometer/ceilometer.conf keystone_authtoken admin_user ceilometer
# openstack-config --set /etc/ceilometer/ceilometer.conf keystone_authtoken admin_password $SERVICE_PASSWORD
```

6. Configure the user credentials in **ceilometer.conf**:

```
# openstack-config --set /etc/ceilometer/ceilometer.conf DEFAULT os_auth_url http://127.0.0.1:35357/v2.0
# openstack-config --set /etc/ceilometer/ceilometer.conf DEFAULT os_tenant_name services
# openstack-config --set /etc/ceilometer/ceilometer.conf DEFAULT os_password $SERVICE_PASSWORD
# openstack-config --set /etc/ceilometer/ceilometer.conf DEFAULT os_username ceilometer
```

7. Start the Ceilometer services:

```
# for svc in compute central collector api ; do
  service openstack-ceilometer-$svc start
  done
```

8. Register an endpoint with the service catalog. Replace **$EXTERNALIFACE** with the IP address of your external interface:

```
# keystone service-create --name=ceilometer \
--type=metering --description="Ceilometer Service"
# CEILOMETER_SERVICE=$(keystone service-list | awk '/ceilometer/ {print $2}')
# keystone endpoint-create \
--region RegionOne \
--service_id $CEILOMETER_SERVICE \
--publicurl "http://$EXTERNALIFACE:8777/" \
--adminurl "http://$EXTERNALIFACE:8777/" \
--internalurl "http://localhost:8777/"
```

9. Enable access to Ceilometer from external systems:

```
# iptables -I INPUT -p tcp -m multiport --dports 8777 -m comment --comment "001 ceilometer incoming" -j ACCEPT
# iptables save
```

10. Confirm the status of OpenStack and the Ceilometer services:

```
# openstack-status
# for svc in compute central collector api ; do
  service openstack-ceilometer-$svc status
  done
```

11. Verify that Ceilometer is working correctly by authenticating as a user with running instances, for example **admin**. Pipe the sample for the CPU meter to count lines, and confirm that the value changes according to the interval specified in **/etc/ceilometer/pipeline.yaml**. The default interval is 600 seconds.

```
# . ~/keystonerc_admin
# ceilometer sample-list -m cpu |wc -l
```

12. Add the configured OpenStack provider to Infrastructure Automation. See After adding the provider, capacity, and utilization data for your instances populate in a few minutes.

Capacity and Utilization Data Collected

Infrastructure Automation generates charts from the collected data that can be used to plan your hardware and virtual machine needs. Depending on the type of data, these charts can include lines for averages, maximums, minimums, and trends.

Note:

For reporting of daily capacity and utilization data, incomplete days (days with less than 24 hourly data points from midnight to midnight) that are at the beginning or end of the requested interval are excluded. Days with less than 24 hourly data points would be inaccurate and including them would skew trend lines. Therefore, at least one full day of hourly data from midnight to midnight is necessary for displaying the capacity and utilization charts under the menu: **Compute > Infrastructure** tab.

Capacity and Utilization Charts for Hosts, Clusters, and Virtual Machines

Resource Type	CPU Usage	CPU States	Disk I/O	Memory Usage	Network I/O	Running VMS	Running Hosts
Host	Y	Y	Y	Y	Y	Y	NA
Cluster	Y	Y	Y	Y	Y	Y	Y
Virtual Machine	Y	Y	Y	Y	Y	NA	NA

Capacity and Utilization Charts for Hosts, Clusters, and Virtual Machines

Capacity and Utilization Charts for Datastores

Charts that are created include:

Space by VM Type	Virtual Machines and Hosts
Used Space	Number of VMs by Type
Disk files Space	Hosts
Snapshot Files Space	Virtual Machines
Memory Files Space	
Non-VM Files	
Used Disk Space	

Capacity and Utilization Charts for Datastores

Capacity and Utilization Chart Features

Capacity and utilization charts for host, clusters, virtual machines, and datastore provides its own set of special features that include zooming in on a chart and shortcut menus.

Zooming into a Chart

1. Browse to the chart you want to zoom. If you hover anywhere on the chart, two dashed lines appear to target a coordinate of the chart.
2. Click (**Click to zoom in**) in the lower left corner of the chart to zoom into it.
3. To go back to the regular view click (**Click to zoom out**) on the enlarged chart.

Drilling into Chart Data

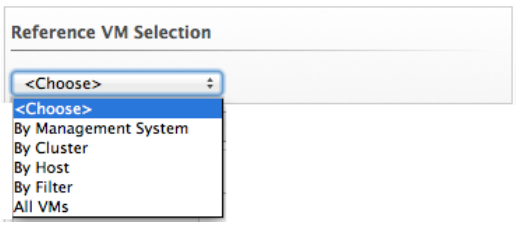
1. Browse to the chart you want to get more detail from.
2. Hover over a data point to see the coordinates.
3. Click a data point to open a shortcut menu for the chart. In this example, you can use the shortcut menu to go to the hourly chart or display the virtual machines that were running at the time the data was captured.
    - If you are viewing the **CPU**, **Disk**, **Memory**, or **Network** charts, selecting from the **Chart** option will change all of the charts on the page to the new interval selected.
    - If you are viewing the **CPU**, **Disk**, **Memory**, or **Network** charts, selecting from the **Display** option will allow you to drill into the virtual machines or **Hosts** that were running at the time.
    - If you are viewing the **VM** or **Hosts** chart, the **Display** menu allows you to view running or stopped virtual machines. The time of the data point will be displayed in addition to the virtual machines that apply. From here, click a virtual machine to view its details.

Planning Where to Put a New Virtual Machine

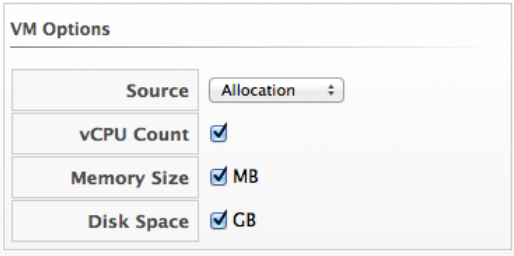
You can use the data that is collected in the VMDB to plan where you can put more virtual machines. With Infrastructure Automation, you can use a reference virtual machine as an example to plan on which hosts and clusters you can place a new virtual machine.

** Note:** For more information about utilization trends and identifying bottlenecks in your environment, see Data Optimization in Managing Infrastructure and Inventory.

1. Browse to **Optimize > Planning**.
2. From **Reference VM Selection**, use the drop-down menu to select the virtual machine that is most like the one that you want to add.

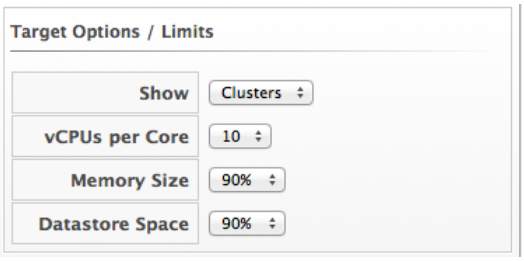


3. Select the required VM Options for what you want to base the calculations on.



From the **Source** list, select the type of data to use as the source for your projections. For example, select **Allocation** to calculate based on the current allocation values of each resource (CPU, memory, or disk space) for the reference virtual machine. Use **Reservation** to project based on the current guaranteed value of the specific resource (CPU Speed, CPU count, memory, or disk space) although that amount might not be allocated to the virtual machine at a specific moment in time. Select **Usage** if you want to calculate based on usage history of the reference virtual machine. Use **Manual Input** to enter your own set of parameters for each resource.

4. From **Target Options / Limits**, select if you want to use clusters or hosts as your targets.



Also, select the limit of how high the projection can go for CPU, memory, and datastore space. If you are targeting hosts, you are able to select a filter for which hosts can be targets.

5. From Trend Options, select how far back you want to use the trend data for, a Time Profile and Time Zone if applicable. Note that Time Profile will only show if the logged on user has a Time Profile available.
6. Click Submit.

The **Summary** tab shows the best clusters or hosts on which to place the virtual machines. The **Report** tab shows the best fit and statistics on the reference virtual machine in a tabular format. From the **Report** tab, you can also create a PDF of the report or download the data in `txt` or `csv` format.

Appendix

Example PostgreSQL Configuration File

```
# -----------------------------
# PostgreSQL configuration file - MIQ Dedicated Appliance Configuration
# -----------------------------
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes        Time units:  ms  = milliseconds
#                MB = megabytes                     s   = seconds
#                GB = gigabytes                     min = minutes
#                                                   h   = hours
#                                                   d   = days


#------------------------------------------------------------------------------
# FILE LOCATIONS
#------------------------------------------------------------------------------
```

```
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'       # use data in another directory
                    # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf' # host-based authentication file
                    # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
                    # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'       # write an extra PID file
                    # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '10.132.50.128'  # MIQ Value;
#listen_addresses = 'localhost'     # what IP address(es) to listen on;
                    # comma-separated list of addresses;
                    # defaults to 'localhost', '*' = all
                    # (change requires restart)
#port = 5432                # (change requires restart)
max_connections = 1600          # MIQ Value increased
#max_connections = 100          # (change requires restart) Note:  Increasing max_connections costs ~400 bytes of shared
memory per connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = ''     # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777     # begin with 0 to use octal notation
                    # (change requires restart)
#bonjour = off              # advertise server via Bonjour
                    # (change requires restart)
#bonjour_name = ''          # defaults to the computer name
                    # (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min      # 1s-600s
#ssl = off              # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH'  # allowed SSL ciphers
                    # (change requires restart)
#ssl_renegotiation_limit = 512MB    # amount of data between renegotiations
#password_encryption = on
#db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres'       # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -
# see "man 7 tcp" for details

tcp_keepalives_idle = 3         # MIQ Value;
#tcp_keepalives_idle = 0        # TCP_KEEPIDLE, in seconds;
                    # 0 selects the system default
tcp_keepalives_interval = 75        # MIQ Value;
#tcp_keepalives_interval = 0        # TCP_KEEPINTVL, in seconds;
                    # 0 selects the system default
tcp_keepalives_count = 9        # MIQ Value;
#tcp_keepalives_count = 0       # TCP_KEEPCNT;
                    # 0 selects the system default


#------------------------------------------------------------------------------
# RESOURCE USAGE (except WAL)
#------------------------------------------------------------------------------

# - Memory -

#shared_buffers = 128MB         # MIQ Value SHARED CONFIGURATION
shared_buffers = 4GB            # MIQ Value DEDICATED CONFIGURATION increased
#shared_buffers = 32MB          # min 128kB
                    # (change requires restart)
#temp_buffers = 8MB         # min 800kB
#max_prepared_transactions = 0      # zero disables the feature
                    # (change requires restart)
# Note:  Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 1MB             # min 64kB
#maintenance_work_mem = 16MB        # min 1MB
#max_stack_depth = 2MB          # min 100kB
```

```
# - Kernel Resource Usage -

#max_files_per_process = 1000       # min 25
                    # (change requires restart)
#shared_preload_libraries = ''      # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0ms        # 0-100 milliseconds
#vacuum_cost_page_hit = 1       # 0-10000 credits
#vacuum_cost_page_miss = 10     # 0-10000 credits
#vacuum_cost_page_dirty = 20        # 0-10000 credits
#vacuum_cost_limit = 200        # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms         # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100        # 0-1000 max buffers written/round
#bgwriter_lru_multiplier = 2.0      # 0-10.0 multipler on buffers scanned/round

# - Asynchronous Behavior -

#effective_io_concurrency = 1       # 1-1000. 0 disables prefetching


#------------------------------------------------------------------------------
# WRITE AHEAD LOG
#------------------------------------------------------------------------------

# - Settings -

#wal_level = minimal            # minimal, archive, or hot_standby
                    # (change requires restart)
#fsync = on             # turns forced synchronization on or off
#synchronous_commit = on        # synchronization level; on, off, or local
#wal_sync_method = fsync        # the default is the first option
                    # supported by the operating system:
                    #   open_datasync
                    #   fdatasync (default on Linux)
                    #   fsync
                    #   fsync_writethrough
                    #   open_sync
#full_page_writes = on          # recover from partial page writes
wal_buffers = 16MB          # MIQ Value;
#wal_buffers = -1           # min 32kB, -1 sets based on shared_buffers
                    # (change requires restart)
#wal_writer_delay = 200ms       # 1-10000 milliseconds

#commit_delay = 0           # range 0-100000, in microseconds
#commit_siblings = 5            # range 1-1000

# - Checkpoints -

checkpoint_segments = 15        # MIQ Value;
#checkpoint_segments = 3        # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 5min      # range 30s-1h
checkpoint_completion_target = 0.9  # MIQ Value;
#checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
#checkpoint_warning = 30s       # 0 disables

# - Archiving -

#archive_mode = off     # allows archiving to be done
                # (change requires restart)
#archive_command = ''       # command to use to archive a logfile segment
#archive_timeout = 0        # force a logfile segment switch after this
                # number of seconds; 0 disables


#------------------------------------------------------------------------------
# REPLICATION
#------------------------------------------------------------------------------

# - Master Server -

# These settings are ignored on a standby server

#max_wal_senders = 0        # max number of walsender processes
                # (change requires restart)
#wal_sender_delay = 1s      # walsender cycle time, 1-10000 milliseconds
#wal_keep_segments = 0      # in logfile segments, 16MB each; 0 disables
#vacuum_defer_cleanup_age = 0   # number of xacts by which cleanup is delayed
#replication_timeout = 60s  # in milliseconds; 0 disables
#synchronous_standby_names = '' # standby servers that provide sync rep
                # comma-separated list of application_name
                # from standby(s); '*' = all

# - Standby Servers -

# These settings are ignored on a master server
```

```
#hot_standby = off          # "on" allows queries during recovery
                    # (change requires restart)
#max_standby_archive_delay = 30s    # max delay before canceling queries
                    # when reading WAL from archive;
                    # -1 allows indefinite delay
#max_standby_streaming_delay = 30s  # max delay before canceling queries
                    # when reading streaming WAL;
                    # -1 allows indefinite delay
#wal_receiver_status_interval = 10s # send replies at least this often
                    # 0 disables
#hot_standby_feedback = off     # send info from standby to prevent
                    # query conflicts


#------------------------------------------------------------------------------
# QUERY TUNING
#------------------------------------------------------------------------------

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_material = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0            # measured on an arbitrary scale
#random_page_cost = 4.0         # same scale as above
#cpu_tuple_cost = 0.01          # same scale as above
#cpu_index_tuple_cost = 0.005       # same scale as above
#cpu_operator_cost = 0.0025     # same scale as above
#effective_cache_size = 128MB

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5            # range 1-10
#geqo_pool_size = 0         # selects default based on effort
#geqo_generations = 0           # selects default based on effort
#geqo_selection_bias = 2.0      # range 1.5-2.0
#geqo_seed = 0.0            # range 0.0-1.0

# - Other Planner Options -

#default_statistics_target = 100    # range 1-10000
#constraint_exclusion = partition   # on, off, or partition
#cursor_tuple_fraction = 0.1        # range 0.0-1.0
#from_collapse_limit = 8
#join_collapse_limit = 8        # 1 disables collapsing of explicit
                    # JOIN clauses


#------------------------------------------------------------------------------
# ERROR REPORTING AND LOGGING
#------------------------------------------------------------------------------

# - Where to Log -


log_destination = 'stderr'      # Valid values are combinations of
                    # stderr, csvlog, syslog, and eventlog,
                    # depending on platform.  csvlog
                    # requires logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on      # Enable capturing of stderr and csvlog
                    # into log files. Required to be on for
                    # csvlogs.
                    # (change requires restart)

# These are only used if logging_collector is on:
log_directory = '/www/postgres/log' # Customer specific setting
#log_directory = 'log'       # directory where log files are written,
                    # can be absolute or relative to PGDATA
log_filename = 'postgresql-%Y-%m-%d.log'        # log file name pattern,
                    # can include strftime() escapes
log_file_mode = 0644                    # creation mode for log files,
                    # begin with 0 to use octal notation
log_truncate_on_rotation = on           # If on, an existing log file with the
                    # same name as the new log file will be
                    # truncated rather than appended to.
                    # But such truncation only occurs on
                    # time-driven rotation, not on restarts
```

```
                    # or size-driven rotation.  Default is
                    # off, meaning append to existing files
                    # in all cases.
log_rotation_age = 1d                   # Automatic rotation of logfiles will
                    # happen after that time.  0 disables.
log_rotation_size = 0                   # Automatic rotation of logfiles will
                    # happen after that much log output.
                    # 0 disables.

# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'

#silent_mode = off          # Run server silently.
                    # DO NOT USE without syslog or
                    # logging_collector
                    # (change requires restart)


# - When to Log -

#client_min_messages = notice       # values in order of decreasing detail:
                    #   debug5
                    #   debug4
                    #   debug3
                    #   debug2
                    #   debug1
                    #   log
                    #   notice
                    #   warning
                    #   error

#log_min_messages = warning     # values in order of decreasing detail:
                    #   debug5
                    #   debug4
                    #   debug3
                    #   debug2
                    #   debug1
                    #   info
                    #   notice
                    #   warning
                    #   error
                    #   log
                    #   fatal
                    #   panic

#log_min_error_statement = error    # values in order of decreasing detail:
                    #   debug5
                    #   debug4
                    #   debug3
                    #   debug2
                    #   debug1
                    #   info
                    #   notice
                    #   warning
                    #   error
                    #   log
                    #   fatal
                    #   panic (effectively off)

log_min_duration_statement = 5000   # MIQ Value- ANY statement > 5 seconds
#log_min_duration_statement = -1    # -1 is disabled, 0 logs all statements
                    # and their durations, > 0 logs only
                    # statements running at least this number
                    # of milliseconds


# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_error_verbosity = default      # terse, default, or verbose messages
#log_hostname = off
log_line_prefix = '%t:%r:%c:%u@%d:[%p]:'    # MIQ Value;
#log_line_prefix = ''           # special values:
                    #   %a = application name
                    #   %u = user name
                    #   %d = database name
                    #   %r = remote host and port
                    #   %h = remote host
                    #   %p = process ID
                    #   %t = timestamp without milliseconds
                    #   %m = timestamp with milliseconds
                    #   %i = command tag
                    #   %e = SQL state
```

```
                    #   %c = session ID
                    #   %l = session line number
                    #   %s = session start timestamp
                    #   %v = virtual transaction ID
                    #   %x = transaction ID (0 if none)
                    #   %q = stop here in non-session
                    #        processes
                    #   %% = '%'
                    # e.g. '<%u%%%d> '
log_lock_waits = on          # MIQ Value - used to track possible deadlock issues
#log_lock_waits = off           # log lock waits >= deadlock_timeout
#log_statement = 'none'         # none, ddl, mod, all
#log_temp_files = -1            # log temporary files equal or larger
                    # than the specified size in kilobytes;
                    # -1 disables, 0 logs all temp files
#log_timezone = '(defaults to server environment setting)'


#------------------------------------------------------------------------------
# RUNTIME STATISTICS
#------------------------------------------------------------------------------

# - Query/Index Statistics Collector -

#track_activities = on
track_counts = on            # MIQ Value;
#track_counts = on
#track_functions = none         # none, pl, all
#track_activity_query_size = 1024   # (change requires restart)
#update_process_title = on
#stats_temp_directory = 'pg_stat_tmp'


# - Statistics Monitoring -

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off


#------------------------------------------------------------------------------
# AUTOVACUUM PARAMETERS
#------------------------------------------------------------------------------

autovacuum = on              # MIQ Value;
#autovacuum = on             # Enable autovacuum subprocess?  'on'
                    # requires track_counts to also be on.
log_autovacuum_min_duration = 0     # MIQ Value;
#log_autovacuum_min_duration = -1   # -1 disables, 0 logs all actions and
                    # their durations, > 0 logs only
                    # actions running at least this number
                    # of milliseconds.
autovacuum_max_workers = 1      # max number of autovacuum subprocesses
                    # (change requires restart)
autovacuum_naptime = 30min      # MIQ Value;
#autovacuum_naptime = 1min      # time between autovacuum runs
autovacuum_vacuum_threshold = 500   # MIQ Value;
#autovacuum_vacuum_threshold = 50   # min number of row updates before
                    # vacuum
autovacuum_analyze_threshold = 500  # MIQ Value;
#autovacuum_analyze_threshold = 50  # min number of row updates before
                    # analyze
autovacuum_vacuum_scale_factor = 0.05   # MIQ Value;
#autovacuum_vacuum_scale_factor = 0.2   # fraction of table size before vacuum
#autovacuum_analyze_scale_factor = 0.1  # fraction of table size before analyze
#autovacuum_freeze_max_age = 200000000  # maximum XID age before forced vacuum
                    # (change requires restart)
#autovacuum_vacuum_cost_delay = 20ms    # default vacuum cost delay for
                    # autovacuum, in milliseconds;
                    # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1  # default vacuum cost limit for
                    # autovacuum, -1 means use
                    # vacuum_cost_limit


#------------------------------------------------------------------------------
# CLIENT CONNECTION DEFAULTS
#------------------------------------------------------------------------------

# - Statement Behavior -

#search_path = '"$user",public'     # schema names
#default_tablespace = ''        # a tablespace name, '' uses the default
#temp_tablespaces = ''          # a list of tablespace names, '' uses
                    # only default tablespace
#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#default_transaction_deferrable = off
#session_replication_role = 'origin'
```

```
#statement_timeout = 0          # in milliseconds, 0 is disabled
#statement_timeout = 43200000           # MIQ statment timeout of 12 hours as a default
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#bytea_output = 'hex'           # hex, escape
#xmlbinary = 'base64'
#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, mdy'
#intervalstyle = 'postgres'
#timezone = '(defaults to server environment setting)'
#timezone_abbreviations = 'Default'     # Select the set of available time zone
                    # abbreviations.  Currently, there are
                    #   Default
                    #   Australia
                    #   India
                    # You can create your own file in
                    # share/timezonesets/.
#extra_float_digits = 0         # min -15, max 3
#client_encoding = sql_ascii        # actually, defaults to database
                    # encoding

# These settings are initialized by initdb, but they can be changed.
lc_messages = 'en_US.UTF-8'         # locale for system error message
                    # strings
lc_monetary = 'en_US.UTF-8'         # locale for monetary formatting
lc_numeric = 'en_US.UTF-8'          # locale for number formatting
lc_time = 'en_US.UTF-8'             # locale for time formatting

# default configuration for text search
default_text_search_config = 'pg_catalog.english'

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''


#------------------------------------------------------------------------------
# LOCK MANAGEMENT
#------------------------------------------------------------------------------

deadlock_timeout = 5s           # MIQ Value - one second is too low, 5 seconds is more "interesting"
#deadlock_timeout = 1s
#max_locks_per_transaction = 64     # min 10
                    # (change requires restart)
# Note:  Each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections + max_prepared_transactions)
# lock table slots.
#max_pred_locks_per_transaction = 64    # min 10
                    # (change requires restart)

#------------------------------------------------------------------------------
# VERSION/PLATFORM COMPATIBILITY
#------------------------------------------------------------------------------

# - Previous PostgreSQL Versions -

#array_nulls = on
#backslash_quote = safe_encoding    # on, off, or safe_encoding
#default_with_oids = off
escape_string_warning = off     # MIQ Value no sure why this is enabled
#escape_string_warning = on
#lo_compat_privileges = off
#quote_all_identifiers = off
#sql_inheritance = on
standard_conforming_strings = off   # MIQ Value not sure why this is enabled
#standard_conforming_strings = on
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equals = off


#------------------------------------------------------------------------------
# ERROR HANDLING
#------------------------------------------------------------------------------

#exit_on_error = off                # terminate session on any error?
#restart_after_crash = on           # reinitialize after backend crash?


#------------------------------------------------------------------------------
# CUSTOMIZED OPTIONS
#------------------------------------------------------------------------------

#custom_variable_classes = ''       # list of custom variable class names
```

- High Availability Guide
  - Environment Overview
    - Requirements
  - Installing the Appliances
    - Installing the Primary Database-Only Appliance
    - Installing the First Infrastructure Automation Appliance
    - Configuring the Primary Database-Only Appliance
    - Installing the Standby Database-Only Appliance
    - Configuring the Standby Database-Only Appliance
    - Installing Additional Infrastructure Automation Appliances
  - Configuring Database Failover
    - Failover Configuration Settings
    - Configuring the Failover Monitor
    - Testing Database Failover
    - Reintroducing the Failed Node
  - Configuring the HAProxy Load Balancer
    - Verifying the HAProxy Configuration
  - Scaling a Highly Available Infrastructure Automation Environment
  - Updating a Highly Available Infrastructure Automation Environment
  - Updating Hostnames on Database-Only Appliances
    - Preparing to Update Appliance Hostnames
    - Updating the Primary Database-Only Appliance Hostname
    - Updating the Standby Database-Only Appliance Hostname
    - Re-configuring High Availability on Database-Only Appliances
      - Configuring the Primary Database-Only Appliance
      - Configuring the Standby Database-Only Appliance
      - Restarting Services

High Availability Guide

Environment Overview

Learn how to configure and manage database high availability in a Infrastructure Automation environment. This configuration allows for disaster mitigation: a failure in the primary database does not result in downtime, as the standby database takes over the failed database's processes. This is made possible by database replication between two or more database servers. In Infrastructure Automation, these servers are *database-only Infrastructure Automation appliances*, which do not have **`evmserverd`** processes that are enabled. This is configured from the **`appliance_console`** menu at the time of deployment.

Two types of appliances that are used in high availability:

- *Database-only Infrastructure Automation appliances*, which do not have **`evmserverd`** processes that are enabled or a user interface.
- *Non-database Infrastructure Automation appliances*, which are standard appliances that contain a user interface and which have **`evmserverd`** processes that are enabled.

In this configuration, a failover monitor daemon is configured and running on each non-database Infrastructure Automation appliance. The failover monitor watches the **`repmgr`** metadata about the database-only appliances present in the cluster. When the primary database-only appliance goes down, the non-database Infrastructure Automation appliances start polling each of the configured standby database-only appliances to monitor which one comes up as the new primary. The promotion is orchestrated either by **`repmgrd`** on the database-only appliances or is done manually. When the non-database Infrastructure Automation appliances find that a standby has been promoted, Infrastructure Automation reconfigures the setup by writing the new IP address in the **`database.yml`** file to point to the new primary.

Note:

Manual steps are required to reintroduce the failed database node back as the standby server. See Reintroducing the Failed Node.

Note, this procedure also does not provide scalability or a multi-master database setup. While a Infrastructure Automation environment is comprised of an engine tier and a database tier, this configuration affects only the database tier and does not provide load balancing for the appliances.

Requirements

For a high availability Infrastructure Automation environment, you need a virtualization host containing at minimum four virtual machines with Infrastructure Automation installed, consisting of:

- One virtual machine for the primary external database containing a minimum of 4 GB dedicated disk space
- One virtual machine for the standby external database containing a minimum of 4 GB dedicated disk space
- Two virtual machines for the non-database Infrastructure Automation appliances

The database-only appliances must reside on a highly reliable local network in the same location.

Important:

It is essential to use the same Infrastructure Automation appliance template version to install each virtual machine in this environment.

Correct time synchronization is required before installing the cluster. After installing the appliances, configure time synchronization on all appliances by using **`chronyd`**.

Installing the Appliances

This chapter outlines the steps for installing and configuring the Infrastructure Automation components needed for high availability: a database cluster comprised of primary and standby database-only appliances, and two (at minimum) non-database Infrastructure Automation appliances.

Installing the Primary Database-Only Appliance

The primary database-only appliance functions as an external database to the non-database Infrastructure Automation appliances.

1. Deploy a Infrastructure Automation appliance with an extra (and unpartitioned) disk for the database at a size appropriate for your deployment. For recommendations on disk space, see Database Requirements in the *Deployment Planning Guide*.
2. Configure time synchronization on the appliance by editing **`/etc/chrony.conf`** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
    1. Log in to the appliance and run the **`appliance_console`** command to access the appliance console menu.
    2. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
4. Resynchronize time information across the appliances:

```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

5. In the appliance console, configure the following:
    1. Configure the hostname by selecting **Set Hostname**.
    2. Select **Configure Application**.
    3. Select **Create key** to create the encryption key. You can create a new key, or use an existing key on your system by selecting **Fetch key from remote machine** and following the prompts.
    4. Select **Create Internal Database**.
    5. Select the database disk. Infrastructure Automation then activates the configuration.
    6. For **Should this appliance run as a standalone database server?**, select **`y`**. Selecting this option configures this appliance as a database-only appliance, and therefore the CP4AIOPS:IA application and **`evmserverd`** processes will not run. This is required in highly available database deployments.

       Warning:

       This configuration is not reversible.
    7. Create the database password.

Note:

Do not create a region at this stage in the procedure.

You have now created the empty database.

You can check the configuration on the appliance console details screen. If configured successfully, **Local Database Server** shows as **`running (primary)`**.

Running the **`psql vmdb_production`** command also provides information about the database.

Installing the First Infrastructure Automation Appliance

Install and configure a Infrastructure Automation appliance to point to the primary database server. You can then create a database region and configure the primary database. This appliance does not serve as a database server.

After installing and configuring an empty database-only appliance in Installing the Primary Database-Only Appliance, the steps in this section create the database schema used by Infrastructure Automation on the primary database-only appliance, and populate the database with the initial data.

Region metadata is required to configure the primary database-only appliance as a primary node in the replication cluster. This must be configured from the Infrastructure Automation appliance before the primary and secondary database-only appliances can be configured.

1. Deploy a Infrastructure Automation appliance. There is no requirement for an extra disk on this appliance.
2. Configure time synchronization on the appliance by editing **`/etc/chrony.conf`** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
    1. Log in to the appliance and run the **`appliance_console`** command to access the appliance console menu.
    2. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
4. Re-synchronize time information across the appliances:

```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

5. In the appliance console, configure the following:
    1. Configure the hostname by selecting **Set Hostname**.
    2. Select **Configure Application**.
    3. Configure this appliance to use the encryption key from the primary database-only appliance:
        1. For **Encryption Key**, select **Fetch key from remote machine**.
        2. Enter the hostname for the primary database-only appliance you previously configured containing the encryption key.
        3. Enter the primary database-only appliance's username.
        4. Enter the primary database-only appliance's password.
        5. Enter the path of the remote encryption key. (For example, **`/var/www/miq/vmdb/certs/v2_key`**.)

            All appliances in the same region must use the same v2 key.
    4. Configure the database:
        1. Select **Create Region in External Database**, since the database is external to the appliances.

            Creating a database region will destroy any existing data and cannot be undone.
        2. Assign a unique database region number.
        3. Enter the port number.
        4. For **Are you sure you want to continue?** Select **`y`**.
    5. Enter the primary database-only appliance's name and access details:
        1. Enter the hostname for the primary database-only appliance.
        2. Enter a name to identify the database.
        3. Enter the primary database-only appliance's username.
        4. Enter a password for the database and confirm the password.

This initializes the database, which takes a few minutes.

You can check the configuration on the appliance console details screen. When configured successfully, **CP4AIOPS:IA Server** will show as **`running`**, and **CP4AIOPS:IA Database** will show the hostname of the primary database-only appliance.

Configuring the Primary Database-Only Appliance

On the primary database-only appliance you created in Installing the Primary Database-Only Appliance, initialize the nodes in the database cluster to configure the database replication. Run these steps from the appliance console:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Primary**.
3. Set a unique identifier number for the server and enter the database name and credentials:
    1. Select a number to uniquely identify the node in the replication cluster.
    2. Enter the name of the database you configured previously.
    3. Enter the cluster database username.
    4. Enter the cluster database password and confirm the password.
    5. Enter the primary database-only appliance hostname or IP address.

        Note:

        The hostname must be visible to all appliances that communicate with this database, including the non-database Infrastructure Automation appliances and any global region databases.
    6. Confirm that the replication server configuration details are correct, and select **`y`** to apply the configuration.

This configures database replication in the cluster.

Installing the Standby Database-Only Appliance

The standby database-only appliance is a copy of the primary database-only appliance and takes over the role of primary database in case of failure.

Follow these steps to create a new standby appliance, or to add another standby appliance to the cluster.

1. Deploy a Infrastructure Automation appliance with an extra (and unpartitioned) disk for the database that is the same size as the primary database-only appliance, as it will contain the same data. For recommendations on disk space, see Database Requirements in the *Deployment Planning Guide*.

2. Configure time synchronization on the appliance by editing **/etc/chrony.conf** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
    1. Log in to the appliance and run the **appliance_console** command to access the appliance console menu.
    2. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
4. Re-synchronize time information across the appliances:

```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

5. In the appliance console, configure the hostname by selecting **Set Hostname**.

You can now configure this appliance as a standby database-only appliance in the cluster.

Configuring the Standby Database-Only Appliance

The steps to configure the standby database-only appliance are similar to that of the primary database-only appliance, in that they prepare the appliance to be database-only, but as the standby.

On the standby database-only appliance, configure the following from the appliance console:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Standby**.
3. Select the database disk. Infrastructure Automation then activates the configuration.
4. Set a unique identifier number for the standby server and enter the database name and credentials:
    1. Select a number to uniquely identify the node in the replication cluster.
    2. Enter the cluster database name.
    3. Enter the cluster database username.
    4. Enter and confirm the cluster database password.
    5. Enter the primary database-only appliance hostname or IP address.
    6. Enter the standby database-only appliance hostname or IP address.

        Note:

        The hostname must be visible to all appliances that communicate with this database, including the engine appliances and any global region databases.

    7. Select **y** to configure the replication manager for automatic failover.
    8. Confirm that the replication standby server configuration details are correct, and select **y** to apply the configuration.

The standby server will then run an initial synchronization with the primary database, and start locally in standby mode. This takes a few minutes.

Verify the configuration on the appliance console details screen for the standby server. When configured successfully, **Local Database Server** shows as **running (standby)**.

Installing Additional Infrastructure Automation Appliances

Install a second virtual machine with a Infrastructure Automation appliance and any additional appliances in the region using the following steps:

1. Deploy a Infrastructure Automation appliance. There is no requirement for an extra disk on this appliance.
2. Configure time synchronization on the appliance by editing **/etc/chrony.conf** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
    1. Log in to the appliance and run the **appliance_console** command to access the appliance console menu.
    2. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
4. Re-synchronize time information across the appliances:

```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

5. In the appliance console, configure the following:
    1. Configure the hostname by selecting **Set Hostname**.
    2. Select **Configure Application**.
    3. Configure this appliance to use the encryption key from the primary database-only appliance:
        1. For **Encryption Key**, select **Fetch key from remote machine**.
        2. Enter the hostname for the primary database-only appliance you previously configured containing the encryption key.

3. Enter the port number.
4. Enter the primary database-only appliance's username.
5. Enter the primary database-only appliance's password.
6. Enter the path of the remote encryption key. (For example, **`/var/www/miq/vmdb/certs/v2_key`**.)
7. Select **Join Region in External Database** from the appliance console menu.

4. Enter the primary database-only appliance's name and access details:
    1. Enter the hostname for the primary database-only appliance.
    2. Enter a name to identify the database.
    3. Enter the primary database-only appliance's username.
    4. Enter a password for the database and confirm the password.

This configuration takes a few minutes to process.

You can check the configuration on the appliance console details screen. When configured successfully, **CP4AIOPS:IA Server** will show as **`running`**, and **CP4AIOPS:IA Database** will show the hostname of the primary database-only appliance.

Configuring Database Failover

The failover monitor daemon must run on all of the non-database Infrastructure Automation appliances to check for failures. In case of a database failure, it modifies the database configuration accordingly.

This configuration is crucial for high availability to work in your environment. If the database failover monitor is not configured, non-database Infrastructure Automation appliances will not react to the failover event and will not be reconfigured against the new primary database host.

Failover Configuration Settings

Configuration settings are located in **`/var/www/miq/vmdb/config/ha_admin.yml`**.

Default settings:

Parameter	Default Value	Description
`failover_attempts`	10	The number of times the **`failover_monitor`** will attempt to failover after discovering the connection to the current database is no longer active.
`db_check_frequency`	120s	The sleep interval in the monitor loop, it defines how frequently the connection to the database is checked.
`failover_check_frequency`	60s	The sleep interval between attempting to failover additional times.

Note:

`failover_attempts` and **`failover_check_frequency`** determine the time limit for the cluster to elect a new primary. To produce quicker failovers, reduce the value for **`failover_check_frequency`** and increase the value for **`failover_attempts`**.

Configuring the Failover Monitor

Configure the failover monitor only on the non-database Infrastructure Automation appliances with the following steps:

1. In the appliance console menu, select **Configure Application Database Failover Monitor**.
2. Select **Start Database Failover Monitor**.

Testing Database Failover

Test that failover is working correctly between your databases with the following steps:

1. Simulate a failure by stopping the database on the primary server:

   ```
   # systemctl stop postgresql.service
   ```

2. To check the status of the database, run:

   ```
   # systemctl status postgresql.service
   ```

   Note:

   You can check the status of the simulated failure by viewing the evm log on the engine appliances with **`journalctl -t evm`**.

3. Check the appliance console summary screen for the primary database. If configured correctly, the **CP4AIOPS:IA Database** value in the appliance console summary should have switched from the hostname of the old primary database to the hostname of the new primary on all Infrastructure Automation appliances.

Upon database server failover, the standby server becomes the primary. However, the failed node cannot switch to standby automatically and must be manually configured. Data replication from the new primary to the failed and recovered node does not happen by default, so the failed node must be reintroduced into the configuration.

Reintroducing the Failed Node

Manual steps are required to reintroduce the failed primary database node back into the cluster as a standby. This allows for greater control over the configuration, and to diagnose the failure.

To reintroduce the failed node, reinitialize the standby database. On the standby database-only appliance, configure the following:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Standby**.
3. Select **y** to remove all previous data from the server and configure it as a new standby database.
4. Set a unique identifier number for the standby server and enter the database name and credentials:
    1. Select a number to uniquely identify the node in the replication cluster. This number can be the same as the node's original identification number.
    2. Enter the cluster database name.
    3. Enter the cluster database username.
    4. Enter the cluster database password.
    5. Enter the new primary database-only appliance hostname or IP address.
    6. Enter the new standby database-only appliance hostname or IP address.

        Note:

        The hostname must be visible to all appliances that communicate with this database, including the engine appliances and any global region databases.

    7. Select **y** to configure the replication manager for automatic failover.

        Note:

        If re-using the node's identification number, select **y** to overwrite the existing node ID (this cannot be undone). Additionally, select **y** to overwrite and reconfigure the replication settings in **`/etc/repmgr/*/repmgr.conf`** when prompted.

    8. Confirm that the replication standby server configuration details are correct, and select **y** to apply the configuration.

The standby server will then run an initial synchronization with the primary database, and start locally in standby mode.

Verify the configuration on the appliance console details screen for the standby server. When configured successfully, **Local Database Server** shows as **`running (standby)`**.

Your Infrastructure Automation environment is now re-configured for high availability.

Configuring the HAProxy Load Balancer

After configuring the appliances as described in Installing the Appliances, configure a load balancer to direct traffic to the non-database Infrastructure Automation appliances.

The following steps highlight the configuration requirements for the load balancer, which in this case is HAProxy. The load balancer is assigned a virtual IP address for the Infrastructure Automation user interface and is pointed to one of the many non-database Infrastructure Automation appliances behind the load balancer in a round robin fashion.

Additionally, to avoid the HAProxy server being a single point of failure, two redundant HAProxy servers are configured in active-passive mode. The failover is orchestrated by using the **`keepalived`** daemon. **`keepalived`** monitors the health of the active load balancer and in case of a failure, the virtual IP is failed over to the passive load balancer, which then becomes active. The virtual IP is configured by `keepalived`.

The virtual IP is the single IP address that is used as the point of access to the Infrastructure Automation appliance user interfaces and is configured in the HAProxy configuration along with a load balancer profile. When an end user accesses the virtual IP, it directs traffic to the appropriate Infrastructure Automation appliance based on the configured HAProxy policy.

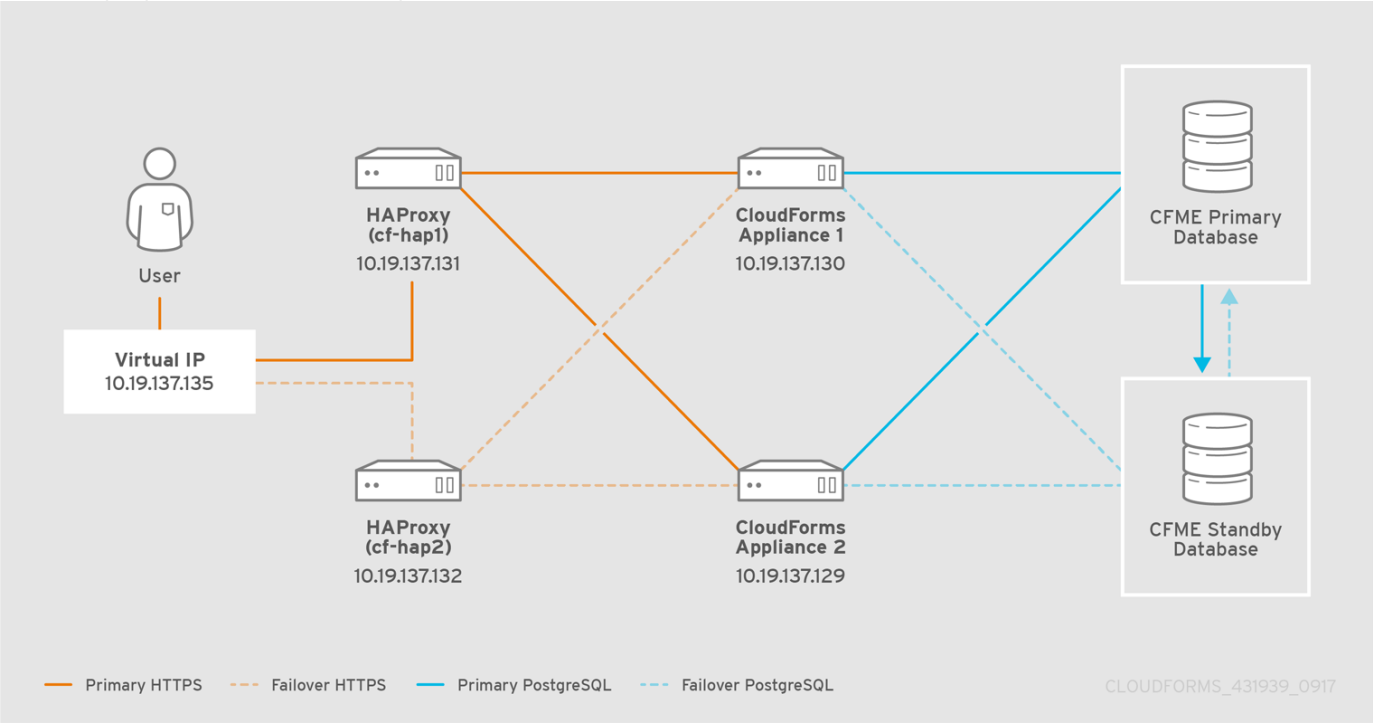
Note:

Additional configuration is required to run HAProxy on Red Hat OpenStack Platform. See the OpenStack Networking Guide for more information.

This configuration uses two HAProxy servers and a virtual IP (configured by **`keepalived`**). The following example procedure uses the following IP addresses and names; substitute values for your environment as needed:

- HAProxy1: 10.19.137.131 (cf-hap1.example.com)
- HAProxy2: 10.19.137.132 (cf-hap2.example.com)
- Virtual IP (to be configured by **`keepalived`**): 10.19.137.135 (cf-ha.example.com)
- CP4AIOPS:IA Appliance 1: 10.19.137.130 (cfme1.example.com)
- CP4AIOPS:IA Appliance 2: 10.19.137.129 (cfme2.example.com)

The following diagram shows the HAProxy configuration in this procedure:



To configure HAProxy load balancing:

1. Install two servers (virtual or physical) running Red Hat Enterprise Linux 7.2 or above, to be used as the HAProxy servers.
2. Configure subscriptions on both HAProxy servers (**cf-hap1** and **cf-hap2**) so that the **rhel-7-server-rpms** repository is enabled:

```
[root@cf-hap1 ~]# subscription-manager repos --list-enabled
+----------------------------------------------+
Available Repositories in /etc/yum.repos.d/redhat.repo
+----------------------------------------------+
Repo Name:   Red Hat Enterprise Linux Server 7 Server (RPMs)

Repo ID:     rhel-7-server-rpms

Repo URL: https://cdn.redhat.com/content/dist/rhel/server/7/$release/$basearch/os
Enabled: 1

[root@cf-hap2 ~]# subscription-manager repos --list-enabled
+----------------------------------------------+
Available Repositories in /etc/yum.repos.d/redhat.repo
+----------------------------------------------+
Repo Name:   Red Hat Enterprise Linux Server 7 Server (RPMs)

Repo ID:     rhel-7-server-rpms

Repo URL: https://cdn.redhat.com/content/dist/rhel/server/7/$release/$basearch/os
Enabled: 1
```

3. Configure the firewall on both HAProxy servers.
    1. On the **cf-hap1** server, run the following:

       Note:

       keepalived uses VRRP (Virtual Redundancy Router Protocol) to monitor the servers and determine which node is the master and which node is the backup. VRRP communication between routers uses multicast IPv4 address 224.0.0.18 and IP protocol number 112.

       ```
       [root@cf-hap1 ~]# firewall-cmd --permanent  --add-port=80/tcp --add-port=443/tcp --add-port=8443/tcp &&
       firewall-cmd --reload

       [root@cf-hap1 ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 \ --in-interface eth0 --
       destination 224.0.0.18 --protocol vrrp -j ACCEPT

       [root@cf-hap1 ~]#firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 \ --out-interface eth0 --
       destination 224.0.0.18 --protocol vrrp -j ACCEPT

       [root@cf-hap1 ~]# firewall-cmd --reload
       ```

    2. On the **cf-hap2** server, repeat the same commands by running the following:

       ```
       [root@cf-hap2 ~]# firewall-cmd --permanent  --add-port=80/tcp --add-port=443/tcp --add-port=8443/tcp &&
       firewall-cmd --reload

       [root@cf-hap2 ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 \ --in-interface eth0 --
       ```

```
destination 224.0.0.18 --protocol vrrp -j ACCEPT

[root@cf-hap2 ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 \ --out-interface eth0 --
destination 224.0.0.18 --protocol vrrp -j ACCEPT

[root@cf-hap2 ~]# firewall-cmd --reload
```

4. Install and configure **keepalived** on both servers.

    1. On the **cf-hap1** server, run the following:

    ```
    [root@cf-hap1 ~]# yum install keepalived -y

    [root@cf-hap1 ~]# cat /etc/keepalived/keepalived.conf
    vrrp_script chk_haproxy {
    script "killall -0 haproxy" # check the haproxy process
    interval 2 # every 2 seconds
    weight 2 # add 2 points if OK
    }
    vrrp_instance VI_1 {
    interface eth0              # interface to monitor
    state MASTER              # MASTER on haproxy1, BACKUP on haproxy2
    virtual_router_id 51
    priority 101              # 101 on haproxy1, 100 on haproxy2
    virtual_ipaddress {
    10.19.137.135/21 # virtual ip address
    }
    track_script {
    chk_haproxy
    }
    }
    ```

    2. On the **cf-hap2** server, run the following:

    ```
    [root@cf-hap2 ~]# yum install keepalived -y

    [root@cf-hap2 ~]# cat /etc/keepalived/keepalived.conf
    vrrp_script chk_haproxy {
    script "killall -0 haproxy" # check the haproxy process
    interval 2 # every 2 seconds
    weight 2 # add 2 points if OK
    }
    vrrp_instance VI_1 {
    interface eth0              # interface to monitor
    state BACKUP              # MASTER on haproxy1, BACKUP on haproxy2
    virtual_router_id 51
    priority 100              # 101 on haproxy1, 100 on haproxy2
    virtual_ipaddress {
    10.19.137.135/21 # virtual ip address
    }
    track_script {
    chk_haproxy
    }
    }
    ```

    3. On both servers, configure IP forwarding and non-local binding by appending the following to the **sysctl.conf** file. In order for the **keepalived** service to forward network packets properly to the real servers, each router node must have IP forwarding turned on in the kernel. On the **cf-hap1** server, run the following:

    ```
    [root@cf-hap1 ~]# cat /etc/sysctl.conf
    # System default settings live in /usr/lib/sysctl.d/00-system.conf.
    # To override those settings, enter new settings here, or in an /etc/sysctl.d/<name>.conf file
    #
    # For more information, see sysctl.conf(5) and sysctl.d(5).
    net.ipv4.ip_forward = 1
    net.ipv4.ip_nonlocal_bind = 1
    ```

    4. On the **cf-hap2** server, run the following:

    ```
    [root@cf-hap2 ~]# cat /etc/sysctl.conf
    # System default settings live in /usr/lib/sysctl.d/00-system.conf.
    # To override those settings, enter new settings here, or in an /etc/sysctl.d/<name>.conf file
    #
    # For more information, see sysctl.conf(5) and sysctl.d(5).
    net.ipv4.ip_forward = 1
    net.ipv4.ip_nonlocal_bind = 1
    ```

    5. Verify that the **sysctl.conf** settings were saved on each server:

    ```
    [root@cf-hap1 ~]# sysctl -p
    net.ipv4.ip_forward = 1
    net.ipv4.ip_nonlocal_bind = 1

    [root@cf-hap2 ~]# sysctl -p
    net.ipv4.ip_forward = 1
    net.ipv4.ip_nonlocal_bind = 1
    ```

5. Install HAProxy on both servers:

```
[root@cf-hap1 ~]# yum install haproxy -y
```

```
[root@cf-hap2 ~]# yum install haproxy -y
```

6. Configure the appropriate IPs for load balancing on the **cf-hap1** server as follows:

```
[root@cf-hap1 ~]# cat /etc/haproxy/haproxy.cfg
global
    log                 127.0.0.1 local0
    chroot              /var/lib/haproxy
    pidfile             /var/run/haproxy.pid
    maxconn         4000
    user                haproxy
    group               haproxy
    daemon
defaults
    mode                        http
    log                         global
    option                      httplog
    option                      dontlognull
    option             http-server-close
    option     forwardfor       except 127.0.0.0/8
    option                      redispatch
    retries                     3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client              1m
    timeout server          1m
    timeout http-keep-alive     10s
    timeout check           10s
# Infrastructure Automation Management UI URL
listen apache
  bind 10.19.137.135:80
  mode tcp
  balance source
  server cfme1 10.19.137.130:80 check inter 1s
  server cfme2 10.19.137.129:80  check inter 1s
#
listen apache-443
  bind 10.19.137.135:443
  mode tcp
  balance source
  server cfme1 10.19.137.130:443 check inter 1s
  server cfme2 10.19.137.129:443  check inter 1s
#
listen apache-8443
  bind 10.19.137.135:8443
  mode tcp
  balance source
  server cfme1 10.19.137.130:8443 check inter 1s
  server cfme2 10.19.137.129:8443  check inter 1s
```

Note:

- The virtual IP in this configuration is 10.19.137.135 (cf-haproxy.example.com).
- The IP of CP4AIOPS:IA Appliance 1 is 10.19.137.130 (cfme1.example.com).
- The IP of CP4AIOPS:IA Appliance 2 is 10.19.137.129 (cfme2.example.com).

7. Configure the appropriate IPs for load balancing on the **cf-hap2** server as well:

```
[root@cf-hap2 ~]# cat /etc/haproxy/haproxy.cfg
global
    log                 127.0.0.1 local0
    chroot              /var/lib/haproxy
    pidfile             /var/run/haproxy.pid
    maxconn         4000
    user                haproxy
    group               haproxy
    daemon
defaults
    mode                        http
    log                         global
    option                      httplog
    option                      dontlognull
    option             http-server-close
    option     forwardfor       except 127.0.0.0/8
    option                      redispatch
    retries                     3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client              1m
    timeout server          1m
    timeout http-keep-alive     10s
    timeout check           10s
# Infrastructure Automation Management UI URL
listen apache
  bind 10.19.137.135:80
  mode tcp
  balance source
```

```
  server cfme1 10.19.137.130:80 check inter 1s
  server cfme2 10.19.137.129:80  check inter 1s
#
listen apache-443
  bind 10.19.137.135:443
  mode tcp
  balance source
  server cfme1 10.19.137.130:443 check inter 1s
  server cfme2 10.19.137.129:443  check inter 1s
#
listen apache-8443
  bind 10.19.137.135:8443
  mode tcp
  balance source
  server cfme1 10.19.137.130:8443 check inter 1s
  server cfme2 10.19.137.129:8443  check inter 1s
```

8. On each server, start the **keepalived** and **haproxy** services:

```
[root@cf-hap1~]# systemctl enable keepalived
[root@cf-hap1~]# systemctl start keepalived
[root@cf-hap1~]# systemctl enable haproxy
[root@cf-hap1~]# systemctl start haproxy

[root@cf-hap2~]# systemctl enable keepalived
[root@cf-hap2~]# systemctl start keepalived
[root@cf-hap2~]# systemctl enable haproxy
[root@cf-hap2~]# systemctl start haproxy
```

Verifying the HAProxy Configuration

Verify the HAProxy configuration by inspecting the following:

On the master node (**cf-hap1**):

```
[root@cf-hap1 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:32:4e brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.131/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 10.19.137.135/21 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:324e/64 scope global mngtmpaddr dynamic
       valid_lft 2591800sec preferred_lft 604600sec
    inet6 fe80::201:a4ff:feac:324e/64 scope link
       valid_lft forever preferred_lft forever
```

On the backup node (**cf-hap2**):

```
[root@cf-hap2 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:33:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.132/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:33a6/64 scope global noprefixroute dynamic
       valid_lft 2591982sec preferred_lft 604782sec
    inet6 fe80::201:a4ff:feac:33a6/64 scope link
       valid_lft forever preferred_lft forever
```

Notice the virtual IP 10.19.137.135 has been started by **keepalived** (VRRP).

Simulate a failure on the master node:

```
[root@cf-hap1 ~]# systemctl stop keepalived
```

Notice the virtual IP failover on the master node (**cf-hap1**):

```
[root@cf-hap1 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:32:4e brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.131/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:324e/64 scope global mngtmpaddr dynamic
       valid_lft 2591800sec preferred_lft 604600sec
    inet6 fe80::201:a4ff:feac:324e/64 scope link
       valid_lft forever preferred_lft forever
```

The backup node (**cf-hap2**) shows the following:

```
[root@cf-hap2 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:33:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.132/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 10.19.137.135/21 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:33a6/64 scope global noprefixroute dynamic
       valid_lft 2591982sec preferred_lft 604782sec
    inet6 fe80::201:a4ff:feac:33a6/64 scope link
       valid_lft forever preferred_lft forever
```

Your environment is now configured for high availability.

The following additional configuration in the Infrastructure Automation user interface worker appliances and the load balancer are recommended for improved performance in worker appliances: - For each Infrastructure Automation appliance behind the load balancer, change the `session_store` setting to `sql` in the appliance's advanced settings. - Configure sticky sessions in the load balancer. - Configure the load balancer to test for appliance connectivity using the `https://appliance_name/ping` URL.

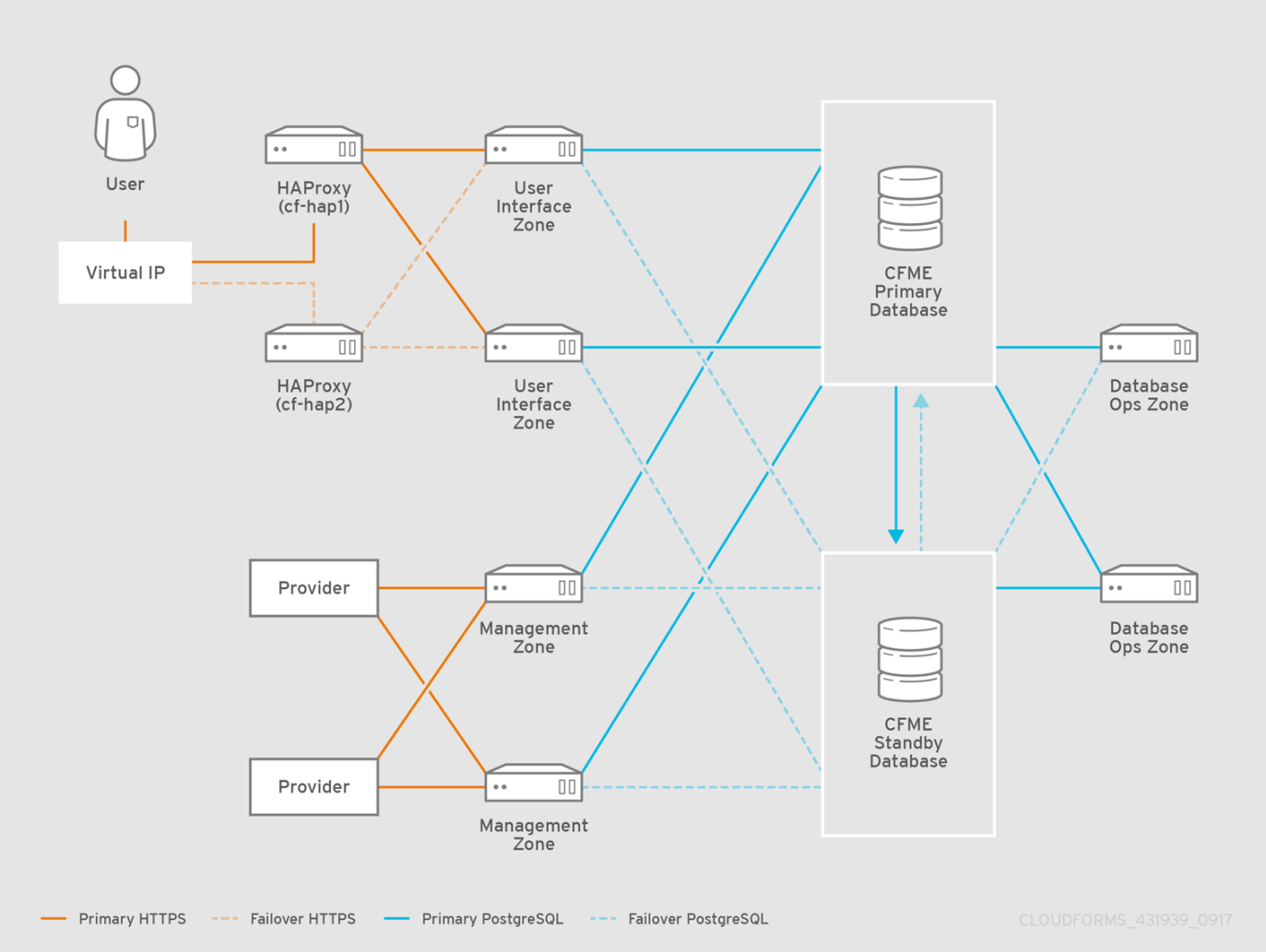
Scaling a Highly Available Infrastructure Automation Environment

After creating high availability for the database tier and the user interface tier, the rest of the infrastructure should be sized appropriately for the roles and the environments that they manage. These roles and tiers use built-in high availability mechanisms like primary, secondary, and tertiary failover.

You can configure additional worker appliances as needed using the steps in Installing Additional Appliances, and then assign zones and server roles. The non-database Infrastructure Automation appliances and roles can be configured in any order.

The following diagram shows an example of a highly available database configuration that contains worker appliances, providers, and the HAProxy load balancer configured in Configuring the HAProxy Load Balancer.

The worker appliances in the diagram are labeled by server role (**User Interface**, **Management**, and **Database Ops**) and corresponding zone to show how a highly available environment might be scaled with server roles and zones.



See Regions and Servers in *General Configuration* for more information on configuring servers and roles.

Updating a Highly Available Infrastructure Automation Environment

Applying software package minor updates (referred to as *errata*) to appliances in a high availability environment must be performed in a specific order to avoid migrating your databases to the next major Infrastructure Automation version.

Prerequisites.

Ensure each appliance is registered to Red Hat Subscription Manager and subscribed to the update channels required by Infrastructure Automation in order to access updates.

To verify if your appliance is registered and subscribed to the correct update channels, run:

```
# yum repolist
```

Appliances must be subscribed to the following channels:

- {product-repo_cfme}
- {product-repo_ansible}
- {product-repo_extras}
- {product-repo_optional}
- {product-repo_server}

Updating the Appliances.

Follow this procedure to update appliances in your environment without migrating the database to the next major version of Infrastructure Automation. Note the appliance to perform each step on: some steps are to be performed only on the database-only appliances, and other steps only on the non-database Infrastructure Automation appliances, while some steps apply to all appliances.

1. Power off the non-database Infrastructure Automation appliances.
2. Power off the database-only appliances.
3. Back up each appliance:
    1. Back up the database of your appliance. Take a snapshot if possible.
    2. Back up the following files for disaster recovery, noting which appliance each comes from:
        - **`/var/www/miq/vmdb/GUID`**
        - **`/var/www/miq/vmdb/REGION`**
    3. Note the hostnames and IP addresses of each appliance. This information is available on the summary screen of the appliance console.
4. Start each database-only appliance.
5. Start each non-database Infrastructure Automation appliance again, and stop **`evmserverd`** on each just after boot:

    ```
    # systemctl stop evmserverd
    ```
6. Apply updates by running the following on each appliance:

    ```
    # yum update
    ```
7. On one of the non-database Infrastructure Automation appliances, apply any database schema updates included in the errata, and reset the Red Hat and ManageIQ automation domains:

    ```
    # vmdb
    # rake db:migrate
    # rake evm:automate:reset
    ```
8. Power off the non-database Infrastructure Automation appliances.
9. Reboot the database-only appliances.
10. Wait five minutes, then start the non-database Infrastructure Automation appliances again.

The appliances in your high availability environment are now up to date.

Updating Hostnames on Database-Only Appliances

When updating the hostnames of database-only appliances in a cluster, you must also re-configure high availability on the primary and standby database-only appliances.

Preparing to Update Appliance Hostnames

Before updating the hostnames of your appliances, complete the following:

1. Note the hostnames of the active database (primary) appliance and the standby database appliances in the appliance console.

    You can verify this from the appliance summary screen of a database-only appliance, where the status for **Local Database Server** shows *primary* or *standby*, for example:

    ```
    ...
    Local Database Server:   running (primary)
    ...
    ```
2. On the non-database Infrastructure Automation appliances, stop the failover monitor:
    1. In the appliance console menu, select **Configure Application Database Failover Monitor**.
    2. Select **Stop Database Failover Monitor**.
3. On the *standby* database-only appliances, stop the **`postgresql`** database service:

```
# systemctl stop postgresql.service
```

4. Stop **evmserverd** on each non-database Infrastructure Automation appliance:

```
# systemctl stop evmserverd
```

Updating the Primary Database-Only Appliance Hostname

Run the following steps on your primary database-only appliance:

1. In the appliance console menu, configure the new hostname or IP address from the **Configure Network** option.
2. Restart the appliance.
3. Update the **host** key to the primary database appliance's new hostname in **/var/www/miq/vmdb/config/database.yml**.

Updating the Standby Database-Only Appliance Hostname

Run the following steps on your standby database-only appliances:

1. In the appliance console menu, configure the new hostname or IP address from the **Configure Network** option.
2. Restart the appliance.
3. Update the **host** key to the standby database appliance's new hostname in **/var/www/miq/vmdb/config/database.yml**.

Repeat these steps on any additional standby database-only appliances.

Your primary and standby appliances must be reachable to each other by their hostnames. If all appliances are on the same network and are resolvable, no additional updates are needed. If your appliances exist on different networks, edit the `/etc/hosts` file on each database appliance to include entries for the IP address and hostname of each other database appliance in the cluster.
Proceed to re-configure high availability on the primary and standby database-only appliances.

Re-configuring High Availability on Database-Only Appliances

Re-configure replication on the database-only appliances, and restart services on your cluster.

Configuring the Primary Database-Only Appliance

On the primary database-only appliance, initialize the nodes in the database cluster to re-configure the database replication:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Primary**.
3. Set a unique identifier number for the server and enter the database name and credentials:
    1. Select a number to uniquely identify the node in the replication cluster.
    2. Enter the cluster database name.
    3. Enter the cluster database username.
    4. Enter the cluster database password and confirm the password.
    5. Enter the new primary database-only appliance hostname or IP address.

       Note:

       The hostname or IP address must be visible to all appliances that communicate with this database, including the non-database Infrastructure Automation appliances and any global region databases.

    6. Confirm that the replication server configuration details are correct, and select **y** to apply the configuration.

Configuring the Standby Database-Only Appliance

The steps to re-configure the standby database-only appliances are similar to that of the primary database-only appliance, in that they prepare the appliance to be database-only, but as the standby.

On the standby database-only appliances, configure the following:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Standby**.
3. Select the database disk. Infrastructure Automation then activates the configuration.
4. Set a unique identifier number for the standby server and enter the database name and credentials:
    1. Select a number to uniquely identify the node in the replication cluster.
    2. Enter the cluster database name.
    3. Enter the cluster database username.
    4. Enter the cluster database password.

5. Enter the new primary database-only appliance hostname or IP address.
6. Enter the new standby database-only appliance hostname or IP address.

   Note:

   The hostname or IP address must be visible to all appliances that communicate with this database, including the engine appliances and any global region databases.
7. Select **y** to configure the replication manager for automatic failover.
8. Confirm that the replication standby server configuration details are correct, and select **y** to apply the configuration. The standby server will then run an initial synchronization with the primary database, and start locally in standby mode.

5. Verify the configuration on the appliance console details screen for the standby server. When configured successfully, **Local Database Server** shows as **running (standby)**.

Repeat these steps on any additional standby database-only appliances.

If you are using non-dedicated database appliances, also stop `evmserverd` on those appliances before changing their hostnames, and reconfigure `database.yml` before restarting.

Restarting Services

1. Start **evmserverd** on each non-database Infrastructure Automation appliance:

   ```
   # systemctl start evmserverd
   ```

   After **evmserverd** has started successfully, all appliances will be able connect to the database.
2. Restart the failover monitor on the non-database Infrastructure Automation appliances:
   1. In the appliance console menu, select **Configure Application Database Failover Monitor**.
   2. Select **Start Database Failover Monitor**.

Note:

You can view a summary of the updated appliances by running **repmgr cluster show** on one of the database appliances.

Your Infrastructure Automation environment is now re-configured for high availability.

- Appliance Hardening Guide
- Introduction to Hardening Infrastructure Automation
  - SSH Security
    - Setting the Root Password on the Appliance
    - Setting SSH Keys on the Appliance
  - Appliance Security
    - Setting the Password for the Administrative User
    - Configuring Host-Based Access Control Rules on your IPA Server
  - Server Security
    - Configuring Firewall Ports
    - Generating SSL Certificates for Your Appliance and Database
      - Creating a Certificate Signing Request
      - Creating a Self-signed Certificate
      - Enabling Your Certificate
    - Creating Custom Encryption Keys
    - Applying SCAP Standards
  - Database Security
    - Restricting Hosts Access to the Database
    - Configuring the Database to use SSL
      - Configure SSL on the database appliance
      - Configure SSL on all connected appliances
      - Hardening TLS Protocol Version

Appliance Hardening Guide

Introduction to Hardening Infrastructure Automation

The Infrastructure Automation appliance is distributed as a virtual machine image, which provides users with a simple installation process. The appliance also contains a set of factory default settings that can expose appliances to vulnerabilities if left unset. This guide provides a set of procedures to enhance security on your Infrastructure Automation appliance. This ensures your appliance has enhanced protection against any unwarranted intrusion.

It is recommended to perform these steps immediately after installing all appliances in your infrastructure.

SSH Security

Setting the Root Password on the Appliance

The Infrastructure Automation appliance is a virtual machine image that runs on a Red Hat Enterprise Linux-based operating system. This means users can access the base operating system through SSH. This is why it is advisable to change the default password. Continuing to use the default password leaves the appliance vulnerable to any user attempting to gain root access.

Changing the **root** password on the appliance 'uses' **the** same process as changing any user password on a Linux-based system.

1. Access your appliance through SSH as the **root** user:

   ```
   [user@localhost ~]$ ssh root@10.1.1.205
   ```

   Substitute **10.1.1.205** with the address of your appliance.

2. Enter the **passwd** command, which changes the password for the current user:

   ```
   [root@ ~]# passwd
   ```

3. Enter and Confirm and new password for the **root** user.

   ```
   Changing password for user root.
   New password: ************
   Confirm password: ************
   ```

4. Log out of the appliance.

The Infrastructure Automation appliance now has a non-default **root** password. This prevents unauthorized access to your appliance through SSH.

Setting SSH Keys on the Appliance

Another recommended practice is to use SSH keys to access the appliance from a single machine. An SSH key provides access from one machine to another through the SSH protocol. The following procedure shows how to create an SSH key on your local machine and add it to the appliance.

1. Check the **.ssh/** directory in your home directory for any existing key pairs:

   ```
   [user@localhost ~]$ ls ~/.ssh/
   ```

   A key pair usually consists of two files. One file is the private key, which stays on your local machine, and the other is the public key, which you copy to another machine. But files are named the same except the public key ends with a **.pub** extension.

   If a key pair already exists, you can use this key pair. Otherwise, use the next few steps to create your own.

2. On your local machine, start the key pair generation process using the **ssh-keygen** command:

   ```
   [user@localhost ~]$ ssh-keygen -t rsa
   ```

3. A prompt asks for the file and location to store these keys:

   ```
   Enter file in which to save the key (/home/user/.ssh/id_rsa):
   ```

   Accept the default path if you do not have a **id_rsa** key pair.

4. Another prompt asks for a passphrase:

   ```
   Enter passphrase (empty for no passphrase):
   ```

   This encrypts the key pair with a password. This protects the key pair if it ever falls into the wrong hands. Alternatively, you can leave the passphrase empty, which provides an automatic login between your local machine and the remote machine.

5. The **ssh-keygen** command generates two files:

   - The private key - the default is */home/user/.ssh/id_rsa*
   - The public key - the default is */home/user/.ssh/id_rsa.pub*

     Copy the public key to the appliance using the **ssh-copy-id** command:

     ```
     [user@localhost ~]$ ssh-copy-id ~/.ssh/id_rsa.pub root@10.1.1.205
     ```

     The command copies the public key to the appliance. You might receive a prompt for the password of the root user on the appliance.

6. Test the SSH key authentication:

   ```
   [user@localhost ~]$ ssh root@10.1.1.205
   ```

   This authenticates using the SSH key pair. If you entered a passphrase for the key, the command prompts you for the passphrase.

7. As an additional security measure, edit the **/etc/ssh/sshd_config** on the appliance and modify the following parameter:

   ```
   PermitRootLogin without-password
   ```

   This forces the **root** user account to use certificates instead of passwords for SSH login. This means only your local system can access the appliance.

The appliance now restricts access to only a single machine using the SSH key.

Appliance Security

Setting the Password for the Administrative User

Infrastructure Automation uses a unique **admin** user to control all functions in the web-based user interface. After installing the appliance, change the default password of the **admin** to restrict administrative access to the appliance's UI.

Changing the **admin** password uses the same process as changing any standard user in the appliance.

1. Access the appliance through your web browser and log in.
2. Click  (**Configuration**).
3. In the accordion tree on the left, click on **Access Control**, then select the **Administrator** under the **Users** section. This displays the details for the **admin** user.
4. On the details page, select **Configuration > Edit this user** from the toolbar.
5. Enter a new password in the **Change Password / Confirm Password** fields.
6. Click **Save** at the bottom of the page.
7. Log out of the user interface.
8. Test your new password by logging into the user interface. Additionally, test your new password in the appliance console.

The Infrastructure Automation appliance now has a non-default **admin** password. This restricts access to your appliance's administrative functions.

Configuring Host-Based Access Control Rules on your IPA Server

Infrastructure Automation provides support for external authentication using an IPA server. However, there are certain recommendations to enhance security to your appliance, such as creating a specific user group and host group that can access the appliance authentication service.

Run the following steps on your IPA server:

1. Create a user group and restrict access to only the Infrastructure Automation users:

```
[root@ipa ~]# ipa group-add infrastructure_automation_users --desc="infrastructure_automation Users"
[root@ipa ~]# ipa group-add-member infrastructure_automation_users --users=testuser1,testuser2
```

2. Create a host group and restrict access to your appliance hosts:

```
[root@ipa ~]# ipa hostgroup-add infrastructure_automation_hosts --desc "Infrastructure Automation hosts"
[root@ipa ~]# ipa hostgroup-add-member infrastructure_automation_hosts --
hosts=appliance1.example.com,appliance2.example.com
```

3. Add rules to allow the host group and user group access to the Infrastructure Automation HTTP service:

```
[root@ipa ~]# ipa hbacrule-add infrastructure_automation_access --srchostcat=all
[root@ipa ~]# ipa hbacrule-add-service infrastructure_automation_access --hbacsvcs httpd-auth
[root@ipa ~]# ipa hbacrule-add-user infrastructure_automation_access --groups infrastructure_automation_users
[root@ipa ~]# ipa hbacrule-add-host infrastructure_automation_access --hostgroups infrastructure_automation_hosts
```

4. Remove the default rule on your IPA server to allow access to all:

```
[root@ipa ~]# ipa hbacrule-disable allow_all
```

This ensures only users in the **infrastructure_automation_users** group can access the authentication service (**http-auth**) on the appliances in the **infrastructure_automation_hosts** host group.

Server Security

Configuring Firewall Ports

A new appliance starts with a few standard ports open:

- 22 for SSH communication
- 80 for HTTP access to the appliance
- 443 for HTTPS access to the appliance
- 5432 for the appliance database

You might need to restrict or open access to certain services on your appliance in the future. In such situations, use the following method:

- Use **firewalld** to enable a service or port, specifying the zone in use. For example, to open the LDAP port:

```
[root@ ~]# firewall-cmd --zone=manageiq --permanent --add-port=389/tcp
```

The following table lists the appliance's main services and their respective ports.

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
Administrator (Internet Browser)	CFME appliance (User Interface)	HTTPS	443		Access to CFME appliance User Interface

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
Administrator (Internet Browser)	CFME appliance (User Interface)	HTTP	80		Redirect Web Browser to HTTPS service (443)
Service Catalog or other integration through Web Service	CFME appliance (Web Service)	HTTPS	443		Access to CFME appliance Web Service
CFME appliance	NFS Server	NFS	2049	2049	Embedded NFS VM scanning
CFME appliance (User Interface)	Any Virtual Machine	TCP	903		VM Remote Console (if using MKS plug-in)
CFME appliance (User Interface)	Any Virtual Machine	TCP	5900-5999		VM Remote Console (if using VNC)
CFME appliance (any role)	CFME appliance running the VMDB	PostgreSQL Named Pipes	5432		CFME appliance connectivity to the CFME Database (PostgreSQL)
CFME Subordinate Region VMDB appliance (Database Operations)	CFME Master Region VMDB appliance	PostgreSQL Named Pipes	5432		Regional VMDB node replication up to Master VMDB node (PostgreSQL only)
CFME Subordinate Region VMDB appliance	CFME Master Region VMDB appliance (Web Services and/or User Interface)	PostgreSQL Named Pipes	5432		Subscription validation (PostgreSQL only)
CFME appliance(Authentication through LDAP)	LDAP Server (AD or other)	LDAP	389		LDAP integration
CFME appliance (Authentication through LDAPs)	LDAP Server (AD or other)	LDAPs	636		LDAPS integration
SNMP Agent	CFME appliance (Notifier)	SNMP (UDP)		161	SNMP Polling
CFME appliance (Notifier)	SNMP Server	SNMP (TCP)	162		SNMP Trap Send
CFME appliance (Notifier)	Mail server	SMTP	25		SNMP Trap Send
CFME appliance (any role)	NTP Server	NTP		123	Time Source
CFME appliance	CFME SmartProxy installed on VMware ESX Server	HTTPS	1139		Communication with SmartProxy
CFME appliance	DNS Server	UDP		53	DNS Lookups

Ports Used by Infrastructure Automation

The following tables detail the ports used by Infrastructure Automation to communicate with providers.

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance (SmartProxy)	RHEV-M Server	HTTPS	8443		API communications to RHEV-M environment (Inventory, Operations, SmartProxy)
CFME appliance (C\&U)	RHEV-M Server	PostgreSQL	5432		RHEV-M History Database (Database connectivity not enabled by default). See How to access the RHEV-M Postgres DB from a remote machine.
CFME appliance	RHEV-H Hosts or RHEL Hypervisors	SSH	22		SSH connections.
CFME appliance	RHEV-H Hosts or RHEL Hypervisors	DirectLUN			Direct LUN hook must be installed and enabled for embedded VM scanning on FC or iSCSI storage devices. Not a tcp/udp connection.

Red Hat Enterprise Virtualization Ports Used by Infrastructure Automation

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	RHOS (Keystone)	HTTP REST API	5000		Authentication and Service Entry Point
CFME appliance	RHOS (Nova)	HTTP REST API	8774		Compute Resources
CFME appliance (C\&U)	RHOS (Ceilometer)	HTTP REST API	8777		Metrics for Capacity and Utilization
CFME appliance	RHOS (Glance)	HTTP REST API	9292		Authentication and Service Entry Point
CFME appliance	RHOS (AMQP)	AMQP	5672		Events Integration
CFME appliance	RHOS (Neutron)	HTTP REST API	9696		Networking
CFME appliance	RHOS (Cinder)	HTTP REST API	8776		Block Storage

Red Hat OpenStack Platform Ports Used by Infrastructure Automation

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME Appliance	OpenShift Master Node(s) (or Load Balancer)	HTTPS	8443 or 443		Required for communication to the OpenShift API. Dependent on OpenShift configuration.
CFME Appliance	OpenShift Infrastructure Node(s) (or Load Balancer)	HTTPS	443		Metrics and logging

OpenShift Container Platform Ports Used by Infrastructure Automation

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance(Management System Inventory, Management System Operations, C & U Data Collection, SmartProxy)	vCenter	HTTPS	443		CFME appliance running any of these roles will initiate communication with vCenter on this port
CFME appliance (SmartProxy)	ESX, ESXi Host	HTTPS	443		CFME appliance
CFME appliance (SmartProxy)	ESX Hosts (if analyzing VMs through host)	SOAP over HTTPS	902		Communication from CFME appliance to hosts
CFME appliance (SmartProxy)	vCenter (if analyzing VMs through VC)	SOAP over HTTPS	902		Communication from CFME appliance to vCenters
CFME appliance(SmartProxy)	ESX Hosts (not needed for ESXi)	SSH	22		CFME appliance console access (ssh) to ESX hosts

VMware vSphere Ports Used by Infrastructure Automation

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	Hyper-V Host (VMM agent)	WinRM/RPC/NetBIOS/SMB (over TCP)	80/135/139/445		Communication from CFME appliance to Host
CFME appliance	Hyper-V Host (file transfer)	HTTPS (using BITS)	443		Communication from CFME appliance to Host
CFME appliance	VM Guest Agent (file transfer)	HTTPS (using BITS)	443		Communication from CFME appliance to VM Guest Agent
CFME appliance	VMware ESX 3.0/3.5 Host (file transfer)	SFTP	22		Communication from CFME appliance to ESX Host
CFME appliance	VMware ESXi Host (file transfer)	SSH/HTTPS (using BITS)	443		Communication from CFME appliance to ESX Host
CFME appliance	WSUS Server (data channel)	HTTP	80/443		Communication from CFME appliance to Server
CFME appliance	SQL Server database (remote)	TDS	1433		CFME appliance connectivity to the Database
CFME appliance	Load Balancer	Load balancer config provider	80/443		
CFME appliance	Hyper-V host in untrusted domain or perimeter network (File Transfer)	TCP	443		CFME appliance connectivity to the host
CFME appliance	Hyper-V Host (file transfer)	BITS	443		Communication from CFME appliance to Host
CFME appliance	VMware Web Services	WCF	443		

SCVMM Ports Used by Infrastructure Automation

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	SQL Management (*.database.windows.net)	TDS	1433		CFME appliance connectivity to the Database
CFME appliance	Upload into Storage (*.blob.core.windows.net)	HTTP/HTTPS	80/443		
CFME appliance	Service Bus Relay HTTP Mode (*.servicebus.windows.net)	SB over HTTP	80		
CFME appliance	Service Bus Pubsub over REST (*.servicebus.windows.net)	HTTPS	443		
CFME appliance	Access Control (*.accesscontrol.windows.net)	HTTPS	443		

Azure Ports Used by Infrastructure Automation

To provide your Infrastructure Automation infrastructure with an extra layer of security, use a network layer firewall to restrict port access.

Generating SSL Certificates for Your Appliance and Database

It is important to enhance the security of SSL communication of your appliances, which, depending on your setup, may include your database appliance. The appliance image ships with a default SSL certificate. It is recommended to replace this certificate with your own certificate, either signed by a trusted Certificate Authority (CA) or self-signed.

Creating a Certificate Signing Request

The first step is to determine the host name of your appliance or database appliance by running the following command:

```
$ hostname
```

The next step is to create a Certificate Signing Request (CSR) using the **openssl** command:

```
[root@ ~]# openssl req -new -newkey rsa:2048 -out appliance.csr -keyout appliance.key
```

This command generates a 2048-bit RSA private key and asks for a passphrase for the key.

```
Generating a 2048 bit RSA private key
..................+++
.........................+++
writing new private key to 'appliance.key'
Enter PEM pass phrase: **********
Verifying - Enter PEM pass phrase: **********
```

The command then provides a questionnaire requesting certain details for the key. Fill out this questionnaire. Use the output of the **hostname** command to specify the **Common Name**.

For example:

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:North Carolina
Locality Name (eg, city) [Default City]:Raleigh
Organization Name (eg, company) [Default Company Ltd]:Infrastructure Automation
Organizational Unit Name (eg, section) []:Customer Content Services
Common Name (eg, your name or your server's hostname) []:$(hostname)
Email Address []:example@example.com

Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Running the command produces two files:

- **appliance.key** - The private key
- **appliance.csr** - The Certificate Signing Request (CSR)

At this stage, you would send the CSR to a trusted Certificate Authority (CA) and in return they would send you a signed certificate.

Creating a Self-signed Certificate

As an alternative to obtaining a signed certificate, you can use the **appliance.key** and **appliance.csr** files to create a self-signed certificate by running the following **openssl** commands:

```
[root@ ~]# openssl rsa -in appliance.key -out server.cer.key
[root@ ~]# openssl x509 -in appliance.csr -out server.cer -req -signkey server.cer.key -days 3650
```

This produces two files:

- **server.cer.key** - The private key for your signed certificate
- **server.cer** - The self-signed certificate

Enabling Your Certificate

Despite whether you used a trusted CA or self-signed the certificate, you should now have your own certificate for your appliance.

Copy the certificate and key files to the certificate directory on the appliance:

```
[root@ ~]# cp ~/server.cer.key /var/www/miq/vmdb/certs/server.cer.key
[root@ ~]# cp ~/server.cer /var/www/miq/vmdb/certs/server.cer
```

After the certificate and key files have been copied, restart the appliance:

```
[root@ ~]# systemctl restart evmserverd
```

The appliance now uses your own certificate.

If your environment consists of multiple appliances connecting to a single database appliance, you can use your certificate and key files to set up SSL for the database connection. For more information, see Configuring the Database to use SSL.

Important:

Updates from the Red Hat Content Delivery Network might overwrite these certificate and key files. Make sure to copy your own certificate and key files to the certificate directory after performing an update to your appliance.

Note:

See also the following article for information on replacing SSL certificates in Infrastructure Automation : https://access.redhat.com/articles/449033.

Creating Custom Encryption Keys

To avoid storing passwords in plain text, Infrastructure Automation appliances use an encryption key to encode and decode passwords. Each appliance stores the key in the **/var/www/miq/vmdb/certs/v2_key**. Changing the encryption key is recommended during setting up new Infrastructure Automation appliances only.

Red Hat does not recommend changing the encryption key for an existing appliance as the ability to decrypt the password will be lost, affecting all stored passwords in Infrastructure Automation.
To generate a new encryption key:

1. Log in to the console of your master appliance as the **root** user.
2. Run the **appliance_console** command. The Infrastructure Automation appliance information screen appears.
3. Press any key to view the appliance menu.
4. Select **Generate Custom Encryption Key**.
5. A prompt asks if for confirmation to overwrite the existing key. Enter **Y**.
6. Enter **1** for **1) Create key**.
7. The appliance generates the new key. Press any key to complete this procedure.

This completes the procedure for generating the new key. If you have external Infrastructure Automation appliances, you must share this key to ensure your whole Infrastructure Automation infrastructure is using consistent encryption. Failure to use the same key results in encryption and decryption problems.

To copy an encryption key:

1. Log in to the console of an external appliance as the **root** user.
2. Run the **appliance_console** command. The Infrastructure Automation appliance information screen appears.
3. Press any key to view the appliance menu.
4. Select **Generate Custom Encryption Key**.
5. A prompt asks if for confirmation to overwrite the existing key. Enter **Y**.
6. Select **Fetch key from remote machine**.
7. Enter the hostname or IP address of the master appliance.
8. Enter the username for SSH access to the master appliance. Use the default **root** user.
9. Enter the password for SSH access to the master appliance.
10. Enter the location of the remote key. Accept the default as **/var/www/miq/vmdb/certs/v2_key**.
11. The appliance copies the new key from the remote server. Press any key to complete this procedure.

After distributing the new key, all appliances require an update to the database configuration. For all appliances, log in as the **root** user and run the following commands replacing **dbpassword** with your database password:

```
[root@{productname_short_l} ~]# fix_auth --databaseyml --hostname localhost --password dbpassword
[root@{productname_short_l} ~]# systemctl restart evmserverd
```

This completes the new encryption key generation for your Infrastructure Automation infrastructure.

Applying SCAP Standards

The Security Content Automation Protocol (SCAP) is a set of standards to assist with vulnerability management and policy compliance. Infrastructure Automation provides a set of SCAP standards to apply to your appliance. View these SCAP rules in the **/var/www/miq/vmdb/productization/appliance_console/config/scap_rules.yml** file.

To apply the SCAP standards to your appliance's server:

1. Log in to the appliance as the **root** user.
2. Enter the **appliance_console** command. The Infrastructure Automation Appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Harden Appliance Using SCAP Configuration**.
5. The appliance console displays the following:

   ```
   Harden Appliance Using SCAP Configuration

   Locking down the appliance for SCAP...
   ```

   The appliance applies the SCAP settings from the **scap_rules.yml** file.
6. When complete, press any key to return to the summary screen.

The appliance now meets the SCAP standards set in the **scap_rules.yml** file.

Database Security

Restricting Hosts Access to the Database

Strengthening the host-based authentication (HBA) settings on a database appliance helps with preventing unauthorized access from external hosts. The HBA settings restrict access to an IP address range so that only hosts within that range have access.

Restricting access to the database requires modifications to the **/var/lib/pgsql/data/pg_hba.conf** file. This file contains a text-based table with some initial settings:

```
# TYPE    DATABASE USER  ADDRESS       METHOD
local     all      all                  peer map=usermap
host      all      all   all            md5
#hostssl  all      all   all            md5
```

This format for this table uses the following header columns:

- TYPE This defines the access type, either local access from the database host (**local**), remote access from an external host regardless of encryption (**host**), external access with encryption (**hostssl**), or external access without encryption (**nohostssl**).
- DATABASE The name of the database the host can access. Use **all** for all databases.
- USER The name of the user the host can use to access the database. Use **all** for all users.
- ADDRESS The IP address of the host or address range of hosts with access to the database. This can either be:
  - A single address:

    ```
    host    all      all   192.168.1.10           md5
    ```

  - An address range using a CIDR mask:

    ```
    host    all      all   192.168.1.0/24          md5
    ```

  - An address range using a separate subnet mask value

    ```
    host    all      all   192.168.1.0  255.255.255.0            md5
    ```

    Note:

    ADDRESS is not required for **local** connections.
- METHOD The authentication method, which includes:
  - **trust** - Allow the connection unconditionally. This method allows anyone that can connect to the PostgreSQL database server to login as any PostgreSQL user they wish, without the need for a password or any other authentication.
  - **reject** - Reject the connection unconditionally. This is useful for "filtering out" certain hosts from a group, for example a **reject** line could block a specific host from connecting, while a later line allows the remaining hosts in a specific network to connect.
  - **md5** - Require the client to supply an MD5-encrypted password for authentication.
  - **password** - Require the client to supply an unencrypted password for authentication. Since the password is sent in clear text over the network, this should not be used on untrusted networks.
  - **ident** - Obtain the operating system user name of the client by contacting the ident server on the client and check if it matches the requested database user name. Ident authentication can only be used on TCP/IP connections. When specified for local connections, peer authentication will be used instead.
  - **peer** - Obtain the client's operating system user name from the operating system and check if it matches the requested database user name. This is only available for local connections.

Using a combination of these options, you create a series of rules that govern which hosts can access your database and which hosts are denied. For example, you might change the default HBA rules to only allow remote access to the Infrastructure Automation database (**vmdb_production**) from hosts in a certain subnet. The modified HBA table would looks like this:

```
# TYPE  DATABASE          USER  ADDRESS         METHOD
local   all               all                   peer map=usermap
host    vmdb_production   all   192.168.1.0/24  md5
#hostssl all              all   all             md5
```

These restrictions help when structuring your Infrastructure Automation appliances in relationships. For example, use these database restrictions to grant access only between a master database appliance in one region and appliances connecting from a separate region.

Configuring the Database to use SSL

Infrastructure Automation initially connects to the database through an unencrypted communication. If you are using multiple appliances that are connecting to a single database appliance, you can set up the database connection to use SSL. An SSL connection encrypts the communication between the Infrastructure Automation and the database.

The procedures in this section use the SSL certificate and the following key files. These files can be found on your main Infrastructure Automation database appliance.

Note:

The appliance image ships with a default SSL certificate and it is recommended to change this certificate. You can use a certificate that is signed by a trusted CA, or generate a self-signed certificate.

For more information, see Generating SSL Certificates for Your Appliance and Database.

- `/var/www/miq/vmdb/certs/server.cer` - Signed or self-signed certificate for the database appliance.
- `/var/www/miq/vmdb/certs/server.cer.key` - Private key for server certificate.

It is also recommended to stop all Infrastructure Automation services before configuring the database to use SSL.

Configure SSL on the database appliance

To configure SSL on the database appliance:

1. Log in as `root` to the appliance where the database resides.
2. Stop the `evmserverd` and `postgresql` services:

   ```
   $ systemctl stop evmserverd
   $ systemctl stop postgresql.service
   ```

3. Install the server key file in the correct location and set the ownership and permissions for it:

   ```
   $ install -m 600 -o postgres -g postgres \
   /var/www/miq/vmdb/certs/server.cer.key /var/lib/pgsql/data/postgres.key
   ```

4. Install the server certificate file in the correct location and set the ownership and permissions for it:

   ```
   $ install -m 644 -o postgres -g postgres \
   /var/www/miq/vmdb/certs/server.cer /var/lib/pgsql/data/postgres.crt
   ```

5. Open the `/var/lib/pgsql/data/postgresql.conf` file and uncomment and edit the `ssl` option:

   ```
   ssl=on
   ```

   In the same file, locate the options `ssl_cert_file` and `ssl_key_file` that specify the location of SSL certificates and edit them so that they are uncommented and point to the correct certificate files:

   ```
   ssl_cert_file = 'postgres.crt'   # (change requires restart)
   ssl_key_file  = 'postgres.key'   # (change requires restart)
   ```

6. Open the `/var/lib/pgsql/data/pg_hba.conf` file and locate the two lines that contain:

   ```
   host      all      all   all           md5
   #hostssl  all      all   all           md5
   ```

   Modify the two lines to comment the `host` entry and uncomment the `hostssl` entry:

   ```
   #host     all      all   all           md5
   hostssl   all      all   all           md5
   ```

   This changes the incoming communication protocol to use SSL and refuse any unencrypted PostgreSQL connections.

7. Start the `postgresql` and `evmserverd` services so that the changes take effect:

   ```
   $ systemctl start postgresql.service
   $ systemctl start evmserverd
   ```

Configure SSL on all connected appliances

The database appliance now accepts only connections from connecting appliances that use SSL. The following procedure sets up connecting appliances to communicate to the database by using SSL. Use this procedure for each connecting appliance:

1. Log in as `root` to the connecting appliance.
2. Copy the root certificate file from the database to the operating system's list of valid certificates:

   ```
   $ DATABASE=[database_appliance_fqdn]
   $ scp root@${DATABASE}:/var/www/miq/vmdb/certs/root.crt /etc/pki/ca-trust/source/anchors/${DATABASE}-postgres
   $ update-ca-trust extract
   $ export PGSSLROOTCERT=/etc/pki/tls/certs/ca-bundle.crt
   ```

   Where `[database_appliance_fqdn]` is the fully qualified domain name of the database appliance.

3. Test the certificate got added to the certificate database:

   ```
   $ openssl x509 -in /etc/pki/tls/certs/ca-bundle.crt -text -noout
   ```

4. Test the connection between the connecting appliance and the database appliance by using the `psql`command:

   ```
   $ PGSSLMODE=verify-full psql -h [database_appliance_fqdn] -d vmdb_production \
     -c "SELECT * from pg_stat_ssl WHERE pid=pg_backend_pid()"
   Password: ********
   psql (9.2.8)
   SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
   Type "help" for help.

   vmdb_production=#
   ```

   The `psql` displays information about the SSL connection, which indicates that the configuration succeeded. Enter `\q` to leave `psql`. The `PGSSLMODE` variable forces a certain level of SSL. To learn more, please see postgress sslmode documentation.

5. Ensure `/etc/default/manageiq.properties` has the proper ssl key file location defined:

```
$ grep PGSSLROOTCERT /etc/default/manageiq.properties || \
echo PGSSLROOTCERT=/etc/pki/tls/certs/ca-bundle.crt >> /etc/default/manageiq.properties
```

The standard **manageiq.properties** should already have the **PGSSLROOTCERT** defined. The above command adds an entry to **manageiq.properties** if it is not already present.

Complete this procedure for each external appliance. This enhances the security of all database transactions in your Infrastructure Automation infrastructure.

Hardening TLS Protocol Version

After configuring the database to use SSL, protocol TLS version 1.2 is used as default. The older versions of this protocol (TLS 1.0 and 1.1) are still available for clients to choose. You can disable older versions by inserting the following lines into **/var/lib/pgsql/data/postgresql.conf**:

```
ssl_ciphers = 'TLSv1.2:!aNULL'
ssl_prefer_server_ciphers=true
```

Terraform-runner in air-gapped environment (offline)

To support Terraform runner in an air-gapped (offline) environment you need to perform some additional configurations in IBM Infrastructure Management.

Use the following steps to support Terraform-runner in an air-gapped environment (offline).

1. Log in to the Infrastructure Management console.
2. Go to **Settings** > **Application Settings**.
3. Go to **Zones** > **"Default Zone" (current)**.
4. Click on **Server "EVM [1]" (current)**.
5. Click **Advanced**.
6. Search for **opentofu_worker** and set the value of **opentofu_offline** to **true** to support the air-gapped environment (offline):

```
:opentofu_worker:
  :container_image: <registry>/cp/cp4waiops/infra-mgmt-opentofu-runner:v4.8.0
  :heartbeat_timeout:
  :opentofu_offline: true
```

Where **<registry>** is the name of your registry where images are stored.

- General Configuration
  - Settings Overview
  - My Settings
    - Visual Settings
      - Grid and Tile Icons
      - Setting Default Items Per Page
      - Setting the Start Page
      - Setting Display Settings
    - Default Views
      - Setting Default Views for the User Interface
      - Setting Default Views for Services
      - Setting Default Views for Clouds
      - Setting Default Views for Infrastructure Components
      - Setting Default Views for Containers
    - Default Filters
      - Setting Default Filters for Cloud
      - Setting Default Filters for Containers
      - Setting Default Filters for Infrastructure
      - Setting Default Filters for Physical Infrastructure
      - Setting Default Filters for Services
    - Time Profiles
      - Creating a Time Profile
      - Editing a Time Profile
      - Copying a Time Profile
      - Deleting a Time Profile
  - Tasks
    - My VM and Container Analysis Tasks
      - Filtering the VM and Container Analysis Task List
      - Deleting a Task from the VM and Container Analysis Task List
    - My Other UI Tasks
      - Filtering the My Other UI Tasks List
      - Deleting a My Task
    - All Tasks
      - Filtering the VM and Container Analysis Task List
      - Deleting a VM and Container Analysis Task
    - All Tasks
      - Filtering the All Tasks List
      - Deleting a Task from the All Tasks List

- Configuration
  - Settings
    - Regions
      - Region Scope
      - Region Settings
      - Capacity and Utilization Collection
        - Capacity and Utilization Collection Settings
        - Enabling a Cluster, Host, or Datastore for Capacity and
      - Tags
        - Company Tag Categories and Tags
        - Creating a Tag Category
        - Deleting a Tag Category
        - Creating a Company Tag
        - Deleting a Company Tag
        - Importing Tags for Virtual Machines
        - Importing Tags for a Virtual Machine from a CSV File
        - Importing Custom Values for Virtual Machines and Hosts
        - Importing Asset Tags for a Virtual Machine from a CSV File
      - Tag Mapping
      - Customizing the Help Menu
    - Profiles
      - Creating an Analysis Profile
      - Creating a Host Analysis Profile
      - Creating a Virtual Machine Analysis Profile
      - Editing an Analysis Profile
      - Copying an Analysis Profile
      - Setting a Default Analysis Profile
    - Zones
      - Creating a Zone
      - Deleting a Zone
      - Editing a Zone
      - Adding SmartProxy Affinity to a Zone:
    - Servers
      - Changing Server Settings
        - Basic Information Settings
        - Server Control Settings
        - Server Roles
        - VMware Console Settings
        - NTP Servers Settings In the NTP Servers area, you can
        - Configuring SNMP
        - Outgoing SMTP Email Settings
        - Web Services Settings
        - Logging Settings
        - Custom Support URL Settings
      - Authentication
        - Changing Authentication Settings
      - Workers
        - Changing Settings for a Worker
      - Database
        - Changing a Database Setting
      - Customization and Logos
        - Custom Logos
        - Uploading a Custom Logo to the User Interface
        - Customizing the Login Background
        - Customizing the Login Panel Text
        - Displaying the Custom Configuration Settings
      - Advanced Settings
        - Editing Configuration Files Manually
        - Configuration Parameters
    - Schedules
      - Scheduling SmartState Analyses
        - Scheduling a SmartState Analysis or Compliance Check
        - Modifying a Schedule
      - Importing and Exporting Schedules
  - Access Control
    - Creating a Tenant
    - Creating a Project
    - Managing Tenant and Project Quotas
    - Tagging Tenants and Projects
    - Creating a User

- Deleting a User
- Groups
- Creating a Group
- Roles
  - Account Roles and Descriptions
- Creating a Role

- Diagnostics
  - Region Diagnostics
    - Server Role Priorities
  - Region Aware Server Roles
  - Setting the Priority of a Failover Role
  - Zone Diagnostics
    - Viewing the Status of Server Roles
    - Zone Aware Server Roles
      - Removing an Inactive Server
    - Zone Log Collections
      - Setting the Location of the Log Depot
      - Collecting and Downloading Logs from All Servers in a Zone
    - Capacity and Utilization Repair
      - Repairing Capacity and Utilization Data
  - Server Diagnostics
    - Workers
      - Reloading Worker Display
      - Restarting a Worker
    - Server and Audit Logs
      - Collecting Server Logs and Configuration Files
      - Setting the Location of the Log Depot for a Specific Server
      - Collecting the Current Log Set of a Server
      - Collecting All Log Sets from a Server
      - Viewing the Server, Audit, and Production Logs
      - Viewing the Server Log
      - Reloading the Server Log
      - Downloading the Server Log
      - Viewing the Audit Log
      - Reloading the Audit Log
      - Downloading the Audit Log
      - Viewing the Production Log
      - Reloading the Production Log
      - Downloading the Production Log
- Database Operations
  - Viewing Information on the VMDB
  - Database Regions and Replication
    - Creating a Region
  - Configuring Database Replication and Centralized Administration
    - Configuring a Remote Copy
    - Configuring the Global Copy
    - Resetting Database Replication
  - Backing Up and Restoring Infrastructure Automation in a Kubernetes Based Environment (podified)
  - Backing Up and Restoring a Database for an Appliance Based Environment
    - Running a Single Database Backup
    - Restoring a Database from a Backup
  - Performing a Binary Backup and Restoring the Database
    - Performing a Binary Backup
    - Restoring a Database from the Binary Backup
  - Running Database Garbage Collection
  - Changing the Database Password
    - Changing the Password on the Database Appliance
    - Changing the Password on the Worker Appliances
  - Adding a New Appliance to an Existing Region with a Non-default Password
  - Creating a Database Dump
  - Enabling Events Syndication
  - Enabling Inventory Syndication
  - Enabling Metrics Syndication
- SmartProxies
  - Installing the SmartProxy from the Console
  - Entering Credentials and Operating System for the Target Host
- About

- Default Roles
- EvmRole-super_administrator
- EvmRole-administrator

- EvmRole-approver
- EvmRole-auditor
- EvmRole-desktop
- EvmRole-operator
- EvmRole-security
- EvmRole-support
- EvmRole-user
- EvmRole-user_limited_self_service
- EvmRole-user_self_service
- EvmRole-vm_user
- EvmRole-tenant_administrator
- EvmRole-tenant_quota_administrator
- EvmRole-consumption_administrator
- EvmRole-container_administrator
- EvmRole-container_operator
- EvmRole-reader

General Configuration

Settings Overview

Use the **Settings** menus of the Infrastructure Automation user interface to view or modify the global configuration of your appliance.

Settings

To view or modify global settings for your appliance, click **Settings > My Settings** to modify. The availability of each menu item depends on the role assigned to your user account. For more information on roles, see Roles.

The following menu items are available from the settings menu:

- **My Settings**

  This menu is available to all Infrastructure Automation users. The settings in this menu control how elements in the user interface are displayed, time profiles, and tags for the currently logged-in individual user.

- **Tasks**

  This menu allows you to view virtual machine SmartState Analysis tasks that can be tracked through the console. The status of each task is displayed, including time started, time ended, what part of the task is currently running, and any errors encountered.

Application Settings

The **Application Settings** menu is now accessible from **Settings > Application Settings** from the user interface.

This menu allows you to specify enterprise, region, zone, and server settings for your Infrastructure Automation infrastructure. Diagnostics such as logs and the status of processes are also shown here.

My Settings

The options under the **My Settings** menu allow you to configure user interface display options specific to the user account with which you log in to Infrastructure Automation, such as the default view displayed on login, and personal tags. You can also configure the color scheme, and button options on the main dashboard.

These options are available by clicking **Settings > My Settings**.

Visual Settings

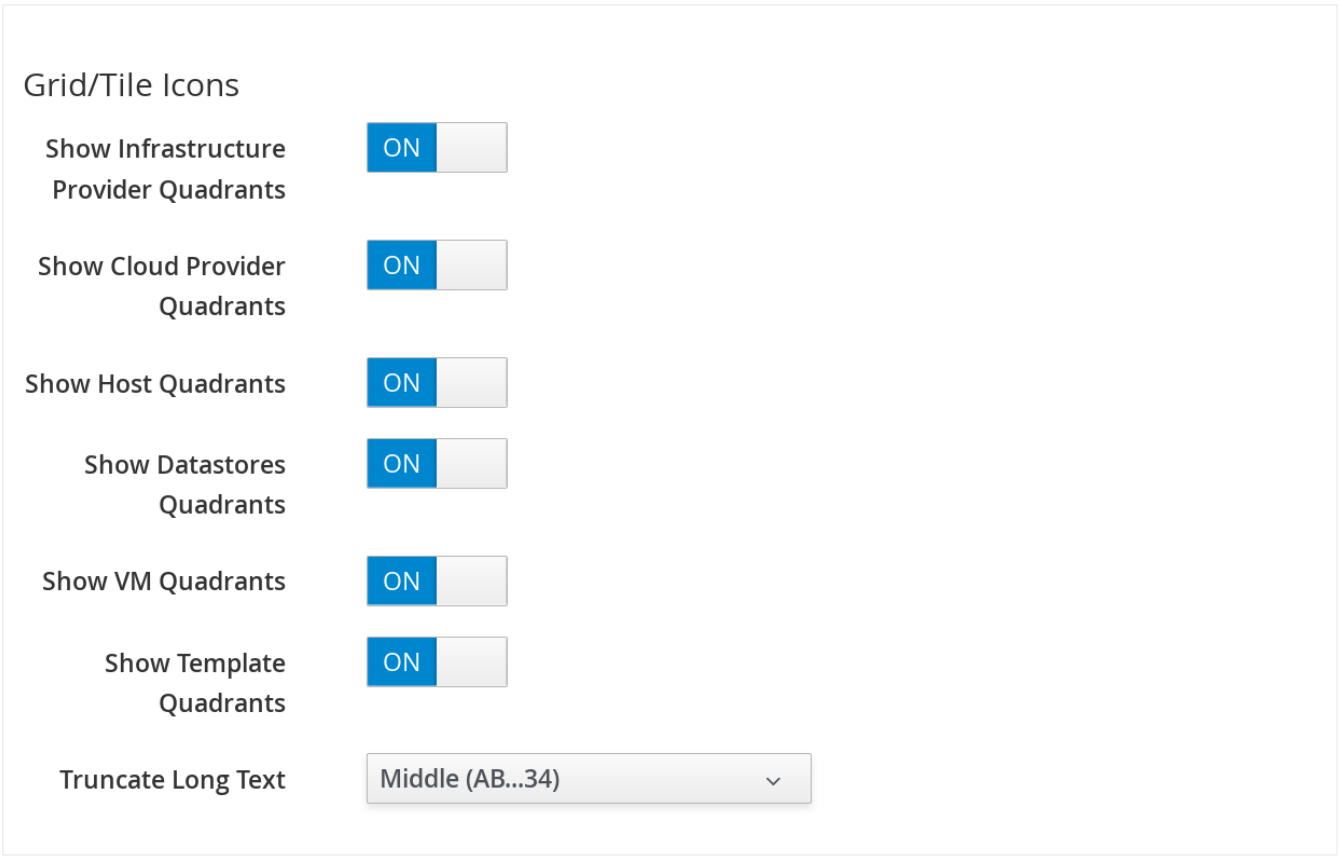
The options under the **Visual** menu allow you to configure how user interface elements are displayed in the web user interface. For all options, click **Save** to update the settings, or click **Reset** to undo any unsaved changes that have been made on the current screen.

Grid and Tile Icons

This group of settings is used to control the view of your virtual thumbnails. Each thumbnail can be viewed as a single icon or as an icon with four quadrants. Use the quadrant view to see a component's properties at a glance.

Use the following procedure to change grid and tile icons:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Visual** tab.
2. In **Grid/Tile Icons**, set items to **ON** to display all four quadrants for the item, or **OFF** to display only one icon.
3. Click **Save**.



- Set **Show Infrastructure Provider Quadrants** to **ON** to see the four icons in your provider under menu: **Compute > Infrastructure > Providers**.
- Set **Show Cloud Provider Quadrants** to **ON** to see the four icons in your hosts under menu: **Cloud > Providers**.
- Set **Show Host Quadrants** to **ON** to see the four icons in your hosts under menu: **Compute > Infrastructure > Hosts**.
- Set **Show Datastores Quadrants** to **ON** to see the four icons in your datastores under menu: **Compute > Infrastructure > Datastores**.
- Set **Show VM Quadrants** to **ON** to see the four icons in your virtual machines under menu: **Compute > Infrastructure > Virtual Machines**.
- Set **Show Template Quadrants** to **ON** to see the four icons in your templates under menu: **Compute > Infrastructure > Virtual Machines > Templates**.

Set any of the above options to **OFF** to see only one icon instead of four quadrants.

- Use the **Truncate Long Text** list to specify how the names of items are displayed if they are too long to show in full. Select the option based on the pattern shown.

Setting Default Items Per Page

Use the following procedure to set the default number of items to display on each resource page.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Visual** tab.

2. In the **Default Items Per Page** area, select the default number of items to display for each view from the corresponding drop down list.



3. Click **Save**.

Setting the Start Page

Use the following procedure to set the default start page after logging in. For example, instead of going to the **Infrastructure Automation** dashboard, you can set the default start page to see a list of your virtual machines.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Visual** tab.
2. In the **Start Page** area, select the page to display at login.
3. Click **Save**.

Setting Display Settings

Use the following procedure to set your own themes, colors, and time zone for the console. These settings are specific to the logged-on user.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Visual** tab.
2. Make selections from **Display Settings** for the following items:
    1. Use **Chart Theme** to select a group of colors and font sizes specifically for charts.
    2. Use **Time Zone** to select the time zone in which to display the console.

        Note:

        In time zones where clocks are set forward for daylight savings time, the time zone correctly displays as EDT (Eastern Daylight Time) in the console. When the clocks are set back, it correctly displays as EST (Eastern Standard Time).

    3. Use **Locale** to select the language in which to display the console.
3. Click **Save**.

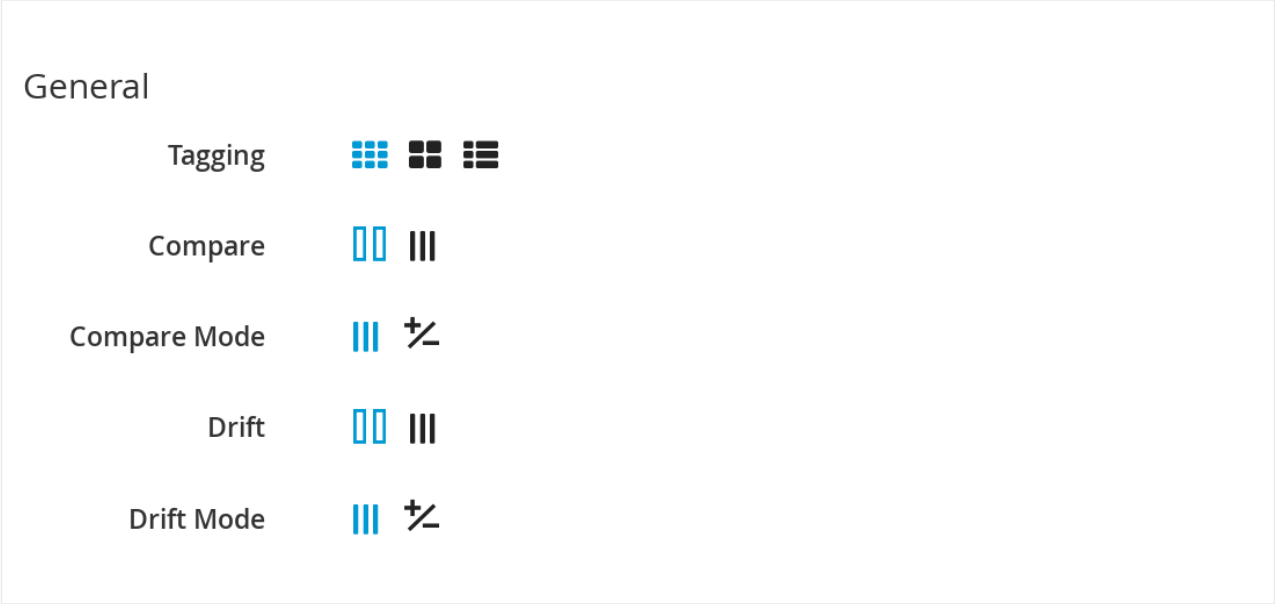
Default Views

The options under the **Default Views** menu allow you to configure the default layout used to display individual screens in the Infrastructure Automation user interface. The options you select under this menu specify the default options for each screen, but you can also change the layout for each screen using the layout buttons on each screen.

Setting Default Views for the User Interface

Use the following procedure to set general view options:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Views** tab.
2. In the **General** area, click the appropriate button for the way you want to view each type of screen listed. The selected view shows as a blue icon.



3. Click **Grid View** to view virtual thumbnails or icons.
4. Click **Tile View** for a view that combines the virtual thumbnail with some text properties that describe the items.
5. Click **List View** to view a detailed text listing.
6. Click **Expanded View** for an expanded view.
7. Click **Compressed View** for a compressed view.
8. Click **Exists Mode** to view only whether an attribute exists or not.
9. Click **Save**.

Setting Default Views for Services

Use the following procedure to set default views for services in the menu: **Compute > Services** tab.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Views** tab.

2. In the **Services** area, click the appropriate button for the way you want to view each item.



- Click **Grid View** to view virtual thumbnails or icons.
- Click **Tile View** for a view that combines the virtual thumbnail with some text properties that describe the items.
- Click **List View** to view a text listing.

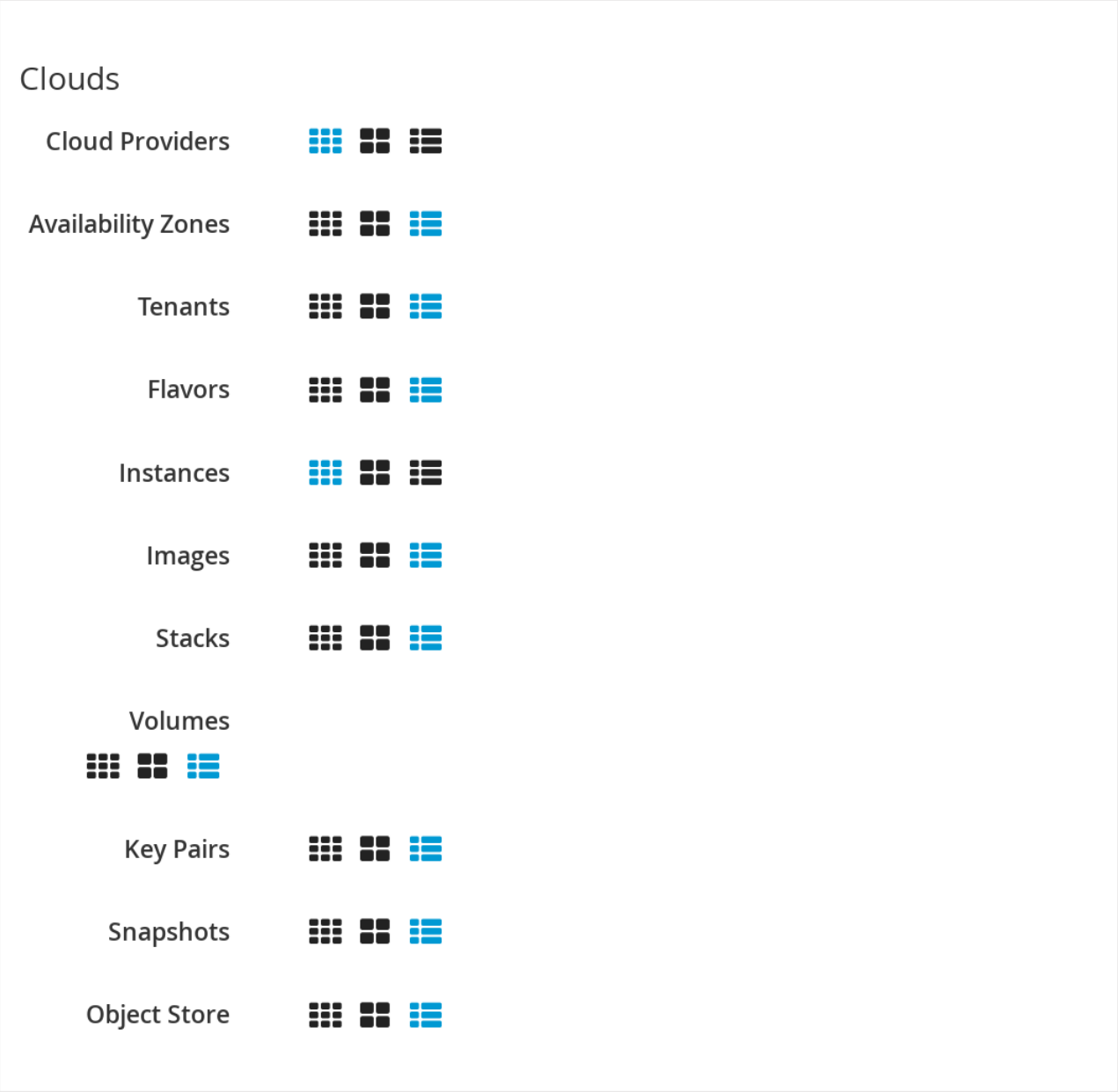
3. Click **Save**.

Setting Default Views for Clouds

Use the following procedure to set default views for clouds in the menu: **Compute > Clouds** tab.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Views** tab.

2. In the **Clouds** area, click the appropriate button for the way you want to view each item.



- Click **Grid View** to view virtual thumbnails or icons.
- Click **Tile View** for a view that combines the virtual thumbnail with some text properties that describe the items.
- Click **List View** to view a detailed text listing.

3. Click **Save**.

Setting Default Views for Infrastructure Components

Use the following procedure to set default views for infrastructure components in the menu: **Compute > Infrastructure** tab.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Views** tab.

2. In the **Infrastructure** area, click the appropriate button for the way you want to view each item.



- Click **Grid View** to view virtual thumbnails or icons.
- Click **Tile View** for a view that combines the virtual thumbnail with some text properties that describe the items.
- Click **List View** to view a detailed text listing.

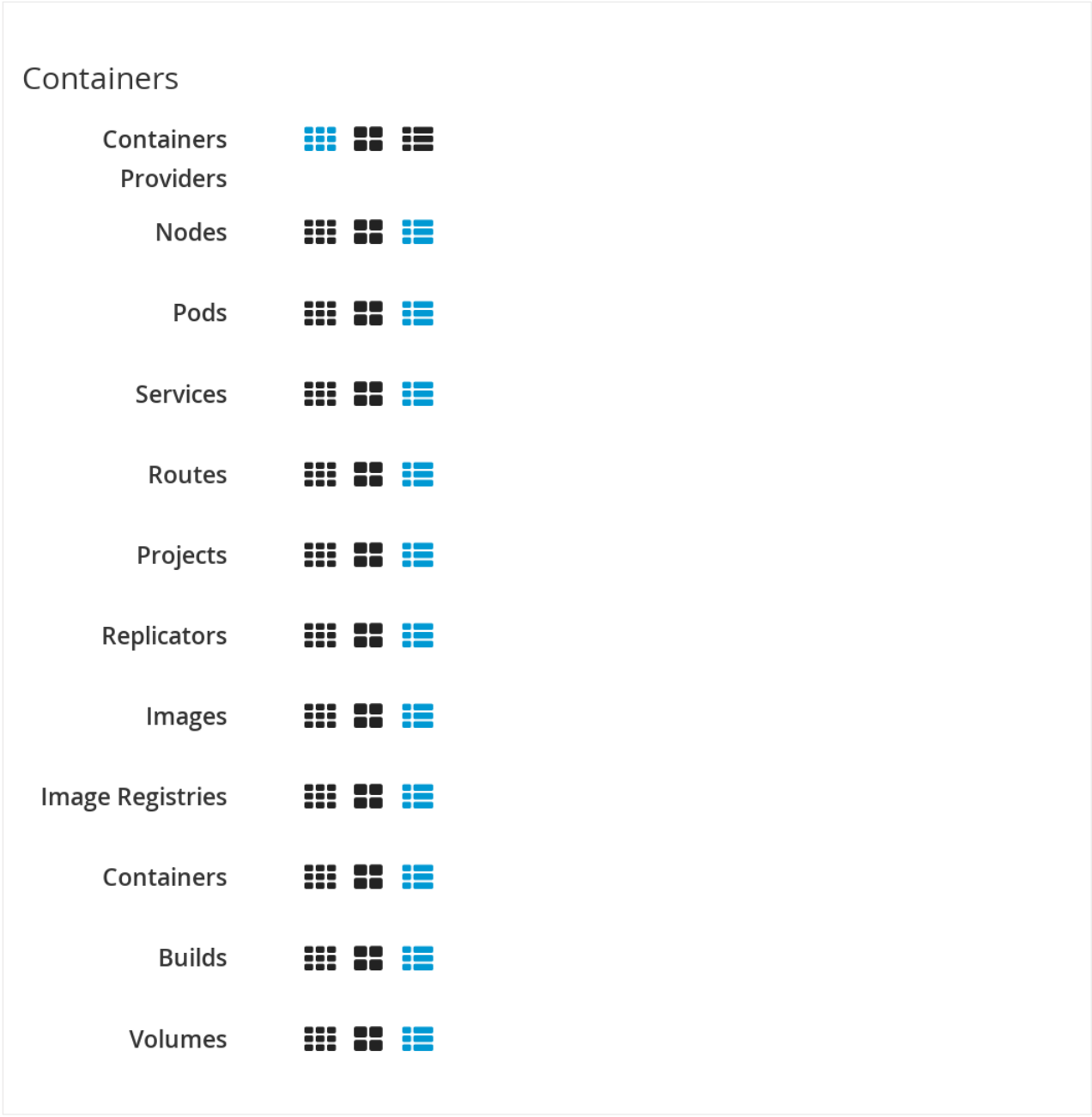
3. Click **Save**.

Setting Default Views for Containers

Use the following procedure to set default views for containers in the menu: **Compute > Containers** tab.

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Views** tab.

2. In the **Containers** area, click the appropriate button for the way you want to view each item.



- Click **Grid View** to view virtual thumbnails or icons.
- Click **Tile View** for a view that combines the virtual thumbnail with some text properties that describe the items.
- Click **List View** to view a text listing.

3. Click **Save**.

Default Filters

The options in the **Default Filters** menu allow you to configure the default filters displayed for your hosts, virtual machines, and templates. These settings are available to all users.

Setting Default Filters for Cloud

To set default filters for cloud components:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Filters** tab.
2. From the **Cloud** folder, check the boxes for the default filters that you want available. Items that have changed show in blue text.
3. Click **Save**.

Setting Default Filters for Containers

To set default filters for containers:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Filters** tab.
2. From the **Containers** folder, check the boxes for the default filters that you want available. Items that have changed show in blue text.
3. Click **Save**.

Setting Default Filters for Infrastructure

To set default filters for infrastructure components:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Filters** tab.
2. In the **Infrastructure** folder, select the default filters that you want available. Items that have changed show in blue text.
3. Click **Save**.

Setting Default Filters for Physical Infrastructure

To set default filters for physical infrastructure components:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Filters** tab.
2. In the **Physical Infrastructure** folder, select the default filters that you want available. Items that have changed show in blue text.
3. Click **Save**.

Setting Default Filters for Services

To set default filters for services:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Default Filters** tab.
2. In the **Services** folder, select the default filters that you want available. Items that have changed show in blue text.
3. Click **Save**.

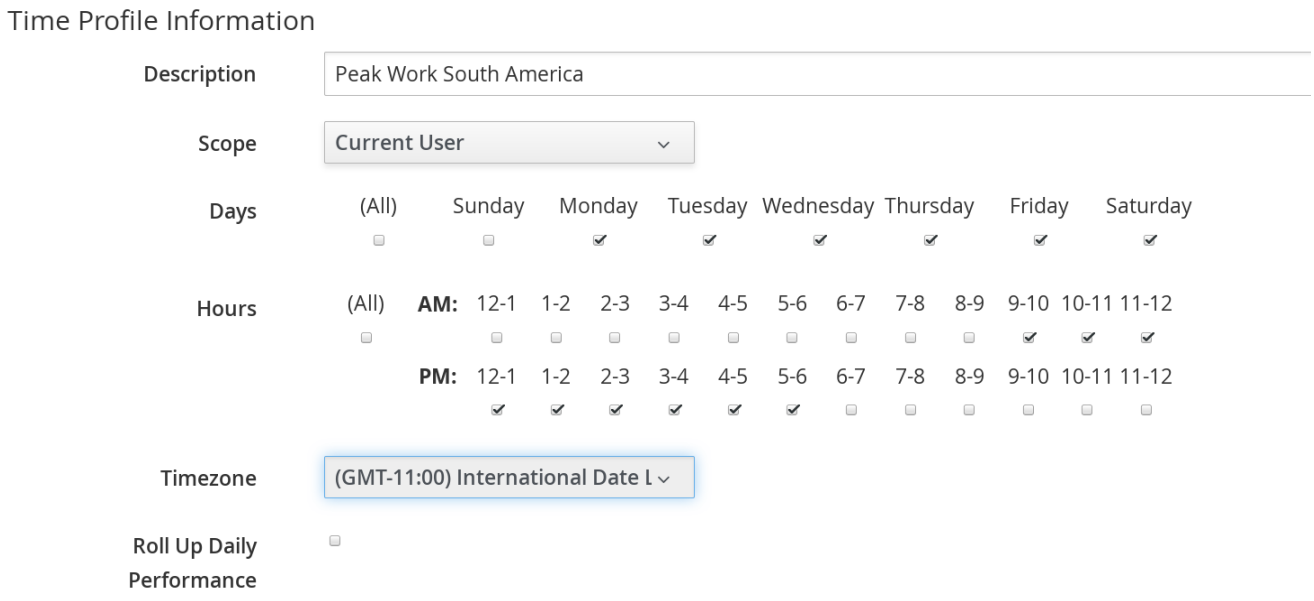
Time Profiles

The options under the **Time Profiles** menu allow you to specify the hours for which data is displayed when viewing capacity and utilization screens. Time profiles are also used to configure performance and trend reports.

Creating a Time Profile

To create a time profile:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Time Profiles** tab.
2. Click **Configuration**, and **Add a new Time Profile**.



3. Type a meaningful name in the **Description** field.

4. Select the users who can access the time profile from the **Scope** list:
    - Select **All Users** to create a time profile that is available to all users. Only the super administration and administration roles can create, edit, and delete a global profile.
    - Select **Current User** if this time profile should only be available to the user creating it.
5. Check the **Days** and **Hours** for the time profile.
6. For **Timezone**, you can select a specific time zone or, you can let the user select a time zone when displaying data.
7. If you select a specific time zone, you also have the option to **Roll Up Daily Performance** data. This option is only available to users with the administration or super administration role. Enabling the **Roll Up Daily Performance option** reduces the time required to process daily capacity and utilization reports and to display daily capacity and utilization charts.
8. Click **Add**.

Note:

The following relationships exist between time zones and performance reports:

- The configured time zone in a performance report is used to select rolled up performance data, regardless of the user's selected time zone.
- If the configured time zone is null, it defaults to UTC time for performance reports.
- If there is no time profile with the report's configured time zone that is also set to roll up capacity and utilization data, the report does not find any records.

For non-performance reports, the user's time zone is used when displaying dates and times in report rows.

Editing a Time Profile

To edit a time profile:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Time Profiles** tab.
2. Check the time profile you want to edit.
3. Click **Configuration**, and **Edit selected Time Profile**.
4. Make the required changes.
5. Click **Save**.

Copying a Time Profile

To copy a time profile:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Time Profiles** tab.
2. Check the time profile you want to copy.
3. Click **Configuration**, and **Copy selected Time Profile**.
4. Make the required changes.
5. Click **Save**.

Deleting a Time Profile

To delete a time profile:

1. From the **Settings** menu, browse to **My Settings**, then click on the **Time Profiles** tab.
2. Check the time profile you want to delete.
3. Click **Configuration**, and **Delete selected Time Profiles**.
4. Click **Save**.

Tasks

The options under the **Tasks** menu allow you to view and control currently running tasks in Infrastructure Automation. The status of each task is displayed, including time started, time ended, what part of the task is currently running, and any errors encountered.

My VM and Container Analysis Tasks

The **My VM and Container Analysis Tasks** menu allows you to view all tasks running on virtual machines and containers for the currently logged-in user.

From the **My VM and Container Analysis Tasks** menu, you can:

- See jobs that the logged on user created for the SmartProxy either through a schedule or by manually initiating a SmartState Analysis of a virtual machine or container.

- See if a job completed successfully, resulted in an error, or is running.
- See the reason for an error.
- Filter the tasks by status and state.
- View the owner or host of the virtual machine or container referenced.
- Delete a task either explicitly or older than another task.

Filtering the VM and Container Analysis Task List

This procedure describes how to filter the virtual machine and container analysis task list. You can filter the task list by zone, time period, task status, and task state.

To filter the virtual machine and container analysis task list:

1. From the **Settings** menu, select **Tasks**.
2. Click **My VM and Container Analysis Tasks**.
3. From the **Zone** list, select a specific zone, or select **All Zones**.
4. From the **24 Hour Time Period** list, select the period of time to view the tasks.
5. For **Task Status**, select the check boxes next to the status to view.
6. From the **Tasks State** list, select the state to view.
7. Click **Apply**.

Deleting a Task from the VM and Container Analysis Task List

To delete a task from the list of virtual machine and container analysis tasks:

1. From the **Settings** menu, select **Tasks**.
2. Click **My Tasks**.
3. Select the tasks to delete. From the **Delete** menu:
    1. Click **Delete** to delete the selected tasks.
    2. Click **Delete Older** to delete the tasks older than the selected tasks.
    3. Click **Delete All** to delete all tasks.
4. Click **OK**.

My Other UI Tasks

The **My Other UI Tasks** menu allows you to view all tasks running in the user interface for the currently logged-in user.

Filtering the My Other UI Tasks List

This procedure describes how to filter the user interface analysis task list. You can filter the task list by time period, task status, and task state.

To filter the other user interface task list:

1. From the settings menu, select **Tasks**.
2. Click **My Other UI Tasks**.
3. From the **24 Hour Time Period** list, select the period of time to view the tasks.
4. For **Task Status**, select the check boxes next to the status to view.
5. From the **Tasks State** list, select the state to view.
6. Click **Apply**.

Deleting a My Task

To delete a task from the list of other user interface tasks:

1. From the settings menu, select **Tasks**.
2. Click **My Tasks**.
3. Select the tasks to delete. From the **Delete** menu:
    1. Click **Delete** to delete the selected tasks.
    2. Click **Delete Older** to delete the tasks older than the selected tasks.
    3. Click **Delete All** to delete all tasks.

4. Click **OK**.

All Tasks

The **All Tasks** menu allows you to view all tasks running on virtual machines and containers for all users.

From the **All Tasks** menu, you can:

- See jobs that all users have created for the SmartProxy either through a schedule or by manually initiating a SmartState Analysis of a virtual machine or container.
- See if a job completed successfully, resulted in an error, or is running.
- See the reason for an error.
- Filter the tasks by status and state.
- View the owner or host of the virtual machine or container referenced.
- Delete a task either explicitly or older than another task.

Filtering the VM and Container Analysis Task List

This procedure describes how to filter virtual machine analysis task lists. You can filter the task list by zone, time period, task status, and task state.

To filter the virtual machine and container analysis task list:

1. From the **Settings** menu, select **Tasks**.
2. Click **All Tasks**.
3. From the **Zone** list, select a specific zone, or select **All Zones**.
4. From the **24 Hour Time Period** list, select the period of time to view the tasks.
5. For **Task Status**, select the check boxes next to the status to view.
6. From the **Tasks State** list, select the state to view.
7. Click **Apply**.

Deleting a VM and Container Analysis Task

To delete a task from the list of all virtual machine and container analysis tasks:

1. From the settings menu, select **Tasks**.
2. Click **All Tasks**.
3. Select the tasks to delete. From the **Delete** menu:
    1. Click **Delete** to delete the selected tasks.
    2. Click **Delete Older** to delete the tasks older than the selected tasks.
    3. Click **Delete All** to delete all tasks.
4. Click **OK**.

All Tasks

The **All Tasks** menu allows you to view all tasks for all users. This menu is only accessible when you are logged in as the **admin** user.

Filtering the All Tasks List

This procedure describes how to filter the list of all other tasks. You can filter the task list by time period, task status, and task state.

To filter the all other tasks list:

1. From the **Settings** menu, select **Tasks**.
2. Click **All Tasks**.
3. From the **24 Hour Time Period** list, select the period of time to view the tasks.
4. For **Task Status**, select the check boxes next to the status to view.
5. From the **Tasks State** list, select the state to view.
6. Click **Apply**.

Deleting a Task from the All Tasks List

To delete a task from the list of all other tasks:

1. From the settings menu, select **Tasks**.
2. Click **All Tasks**.
3. Select the tasks to delete. From the **Delete** menu:
    1. Click **Delete** to delete the selected tasks.
    2. Click **Delete Older** to delete the tasks older than the selected tasks.
    3. Click **Delete All** to delete all tasks.
4. Click **OK**.

Configuration

The options under the **Configuration** menu allow you to configure global options for your Infrastructure Automation environment, view diagnostic information, and view analytics on the servers in the environment. The menu displays the Infrastructure Automation environment at the enterprise, zone, and server levels.

There are four main areas:

- **Settings**

  This menu allows you to configure global settings for your Infrastructure Automation infrastructure. You can also create analysis profiles and schedules for these profiles.

- **Access Control**

  This menu contains options for configuring users, groups, roles, and tenants.

- **Diagnostics**

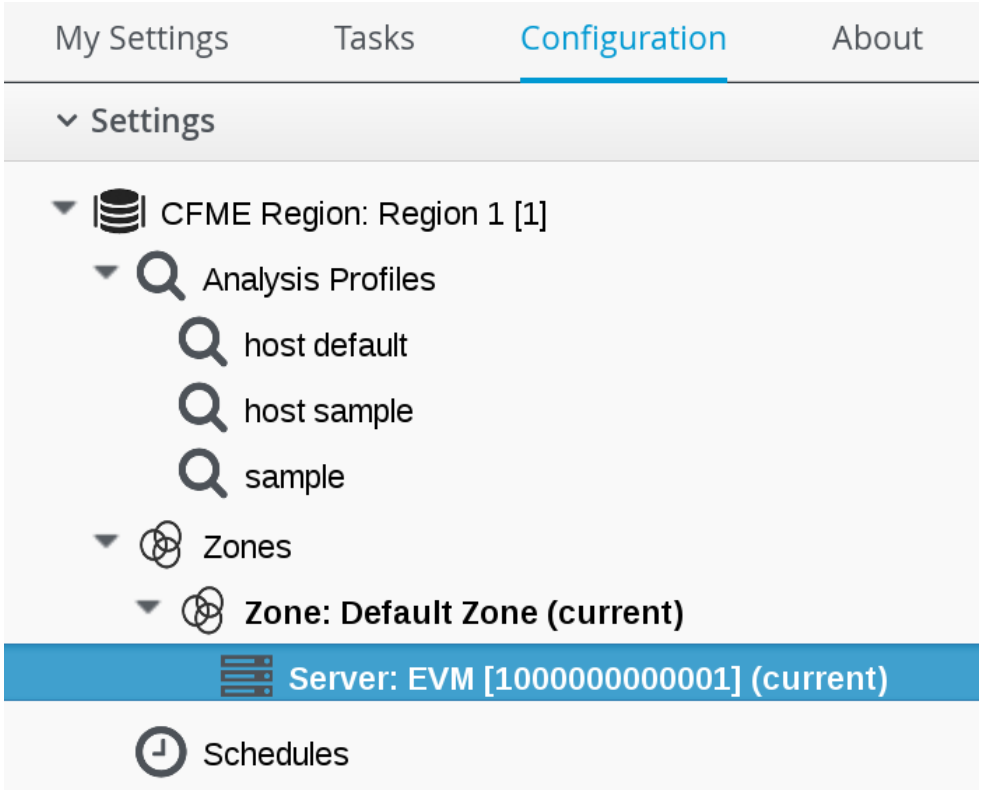
  This menu displays the status of your servers and their roles and provides access to logs.

- **Database**

  specify the location of your Virtual Machine Database (VMDB) and its login credentials.

Settings

The options under the **Settings** area provide a hierarchical view of options that allow you to configure global options for the infrastructure of your Infrastructure Automation environment. At the top level, you have **Settings** including users, LDAP Groups, account roles, capacity and utilization collection, tag categories, values, and imports, custom variable imports, and license uploads. When you click on **Settings** and expand it, you can configure **Analysis Profiles**, **Zones**, and **Schedules**.



When you go the **Settings** accordion, you are automatically taken to the server list under **Zones**.

Regions

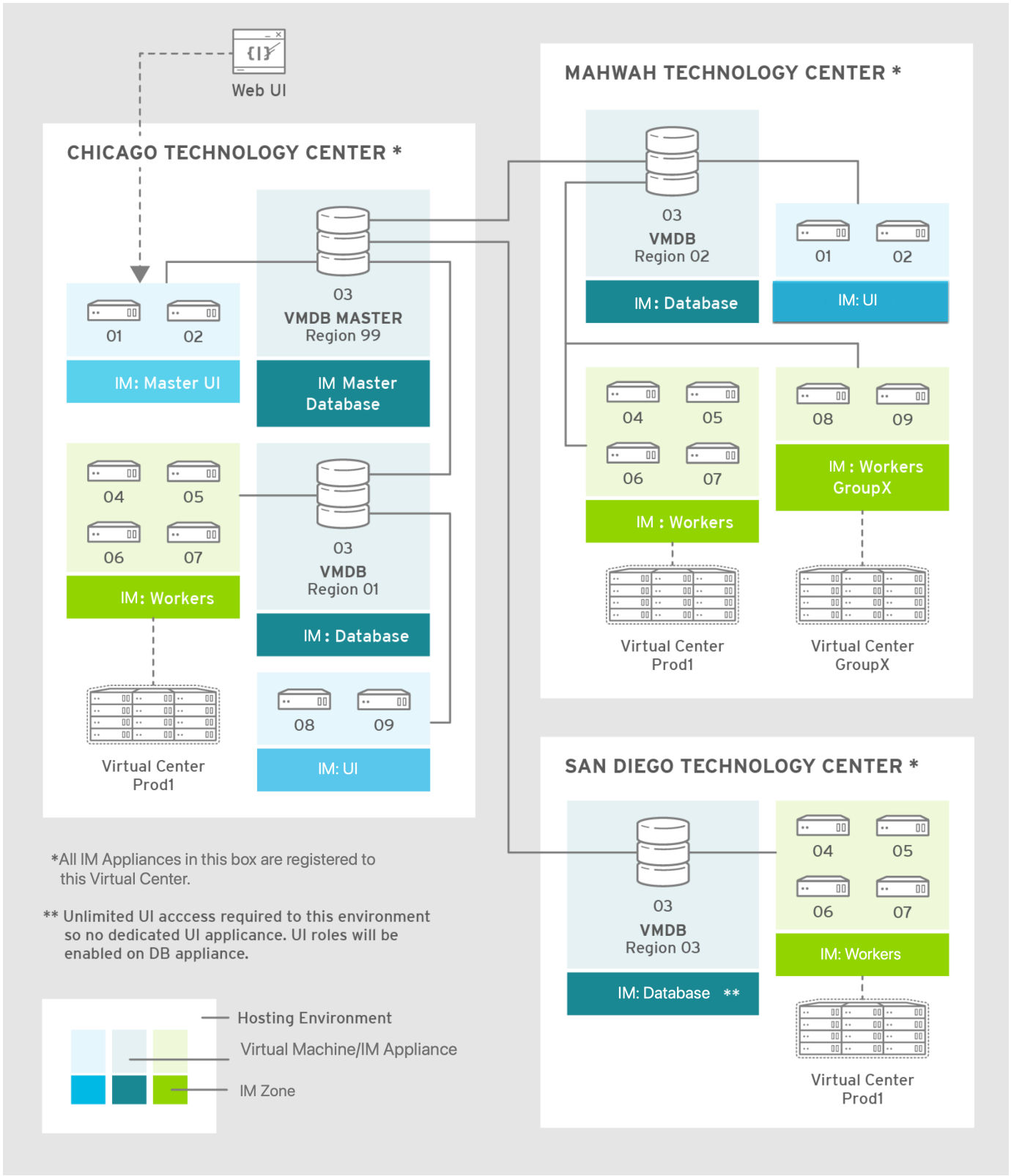
Use **Regions** for centralizing data which is collected from public and private virtualization environments. A region is ultimately represented as a single database for the VMDB. Regions are particularly useful when multiple geographical locations need to be managed as they enable all the data collection to happen at each particular location and avoid data collection traffic across slow links between networks.

When multiple regions are being used, each with their own unique ID, a master region can be created to centralize the data of all the children regions into a single master database. To do this, configure each child region to replicate its data to the master region database (Red Hat recommends use of region 99). This parent and child region is a one-to-many relationship.

Regions can contain multiple zones, which in turn contain appliances. Zones are used for further segregating network traffic along with enabling failover configurations. Each appliance has the capability to be configured for a number of specialized server roles. These roles are limited to the zone containing the appliance they run on.

Only one failover type of each server role can run in a zone. If multiple appliances have the same failover role, the extras are used as backups that activate only if the primary appliance fails. Non-failover server roles can run on multiple appliances simultaneously in a zone, so resources can be adjusted according to the workload those roles are responsible for.

The following diagram demonstrates an example of the multiple regions working together in a Infrastructure Automation environment.

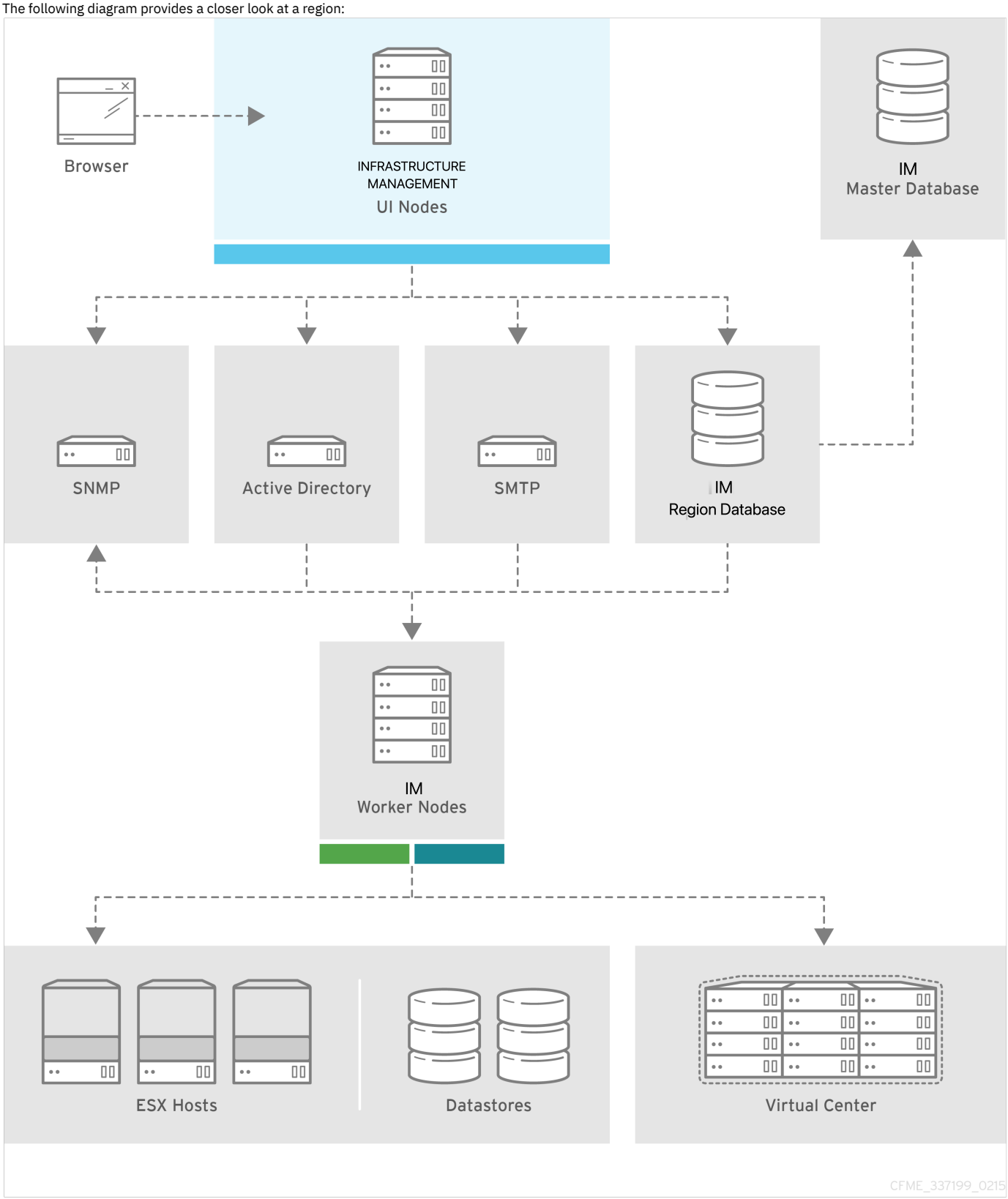


The Master appliance is located in Chicago and contains a master region and a subregion that manages the worker appliances. The Mahwah technology center contains a single subregion that manages two zones. Likewise the San Diego technology center contains a single subregion managing a single zone.

Note:

- Replicating a parent region to a higher-level parent is not supported.
- Parent regions can be configured after the child regions are online.

The following diagram provides a closer look at a region:



In this region, we have several Infrastructure Automation appliances acting as UI nodes and worker nodes. These worker nodes execute tasks on the providers in your environment. The Region also uses a region database that reports to a master database on the main Infrastructure Automation appliance. All appliances can connect to the authentication services (Active Directory, LDAP, Identity Management), outgoing mail (SMTP), and network services (SNMP).

Region Scope

Regions are used to consolidate data from multiple VMDBs to a central database. The database at the top level, the master VMDB, cannot be used for operational tasks such as SmartState Analysis or Capacity and Utilization data collection. It is intended for use as a reporting database that includes all information across multiple subordinate regions. The subordinate regions replicate their information to the master.

Note:

The subordinate regions are not aware of each other from a database perspective. You cannot see information from one subordinate region in another. The only VMDB with data visibility to all subordinate regions is the top level.

Master Regions Scope

- Reports all information from all subordinate VMDBs reporting up to it.
- Can perform power operations on virtual machines from subordinate regions.
- Controls its own access control list.

Subordinate Regions Scope

- Each subordinate controls its own access control independent of the other regions.
- Can only do work (such as SmartState Analysis and Capacity and Utilization collection) in its own region.
- Has no knowledge of the other regions.
- Replicates its data up to the master region.

Region Settings

In the **Region** area, set items that apply to your entire Infrastructure Automation infrastructure such as users, LDAP Groups, capacity and utilization collection, company tags and tag categories, and licensing. Regions are also used for database replication.

Capacity and Utilization Collection

Capacity and Utilization Collection Settings

Use **C & U Collection Settings** to select specifically which clusters and datastores you want to collect usage data for. By selecting a cluster, you are choosing to collect data for all hosts and virtual machines that are part of that cluster. You must also have a server with the Capacity & Utilization **Coordinator**, **Data Collector**, and **Data Processor** roles enabled as well. See Section **Server Control Settings**.

After a provider has been discovered and its relationships refreshed, view the clusters, hosts, and datastores from the settings menu. Navigate to **Settings**, then click on **Application Settings**. Click the menu: **Settings > Region**, then select the **C & U Collection** tab.

Enabling a Cluster, Host, or Datastore for Capacity and

Utilization Collection

To enable a cluster, host, or datastore for Capacity and Utilization Collection:

1. Click **Settings**, then click on **Application Settings**.
2. Click the **Settings** accordion, then select **Region**, then click on the **C & U Collection** tab.
3. In the **Clusters** area, check all clusters and hosts that you want to collect data for.
4. In the **Datastores** area, check all datastores that you want to collect data for.
5. Click **Save**.

Note:

1. As new clusters, hosts, and datastores are discovered, you will need to come back to this configuration to enable collection of capacity and utilization data unless you have used the **Collect for All** check boxes.
2. **Collect for All Clusters** must be checked to be able to collect capacity and utilization data from cloud providers such as Red Hat OpenStack Platform or Amazon EC2.

Tags

Company Tag Categories and Tags

Infrastructure Automation allows you to create your own set of tags and tag categories. Use tags to create a customized, searchable index for your resources. Depending on your database type, your tags may be case sensitive. After creating these values, you can apply them to your resources. There are two kinds of tags.

- **Company tags** which you will see under **My Company Tags** for a resource. Create company tags from the settings menu. Browse to **Settings > Application Settings**. Click the **Settings** accordion, then select the **Region** and click on the **Tags** tab, then **My Company Tags**. A selection of company tags is provided to you by default as samples. These can be deleted if you do not need them, but are not recreated by Infrastructure Automation.
- **System tags** are assigned automatically by Infrastructure Automation.

Creating a Tag Category

To create a tag category:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then **Region**.

3. Select the **Tags** tab, then click on the **My Company Categories** tab.
4. Click **Add** under the **Actions** column to create a new category.
5. In the **Category Information** area:



- Use **Name** to create a short name that refers to category in the VMDB.

  Note:

  The **Name** and **Single Value** fields cannot be changed after the category has been added.
- Use **Description** to type a brief explanation of how the category should be used. This shows when you try to add a value to the category.
- Use **Long Description** to type a detailed explanation of the category.
- Set **Show in Console** to ON when the category is ready for use in the console. For example, you want to populate values for the category before exposing it to users.
- Set **Single Value** to ON for categories that can only have a single value assigned to a resource. For example, a virtual machine can only be assigned to one location, but could belong to more than one department. This cannot be changed after the category is created.
- Set **Capture C & U Data by Tag** to ON for the ability to group capacity and utilization data by this tag category. To use this, be sure to assign this tag to all the resources that you want to group by.

6. Click **Add**.

Repeat these steps for each category you need. After you have created the category, you can add values to it.

Important

If no values are created for a category, you are unable to assign a value from that category nor be able to filter by that category.

Deleting a Tag Category

To delete a tag category:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then **Region**.
3. Select the **Tags** tab, then click on the **My Company Categories** tab.
4. Click **Delete** under the **Actions** column for the category you want to delete.
5. Click **OK** to confirm.

Note:

When you delete a tag category, the category values are removed, and any tags from the category are unassigned from all resources.

Creating a Company Tag

To create a company tag:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then **Region**.
3. Select the **Tags** tab, then click on the **My Company Tags** tab.
4. In the **Choose a Category** area, select a category from the **Category** list.

   Note:

   - Some categories only allow one value to be assigned to a resource.
   - For some databases such as **PostgreSQL**, tags are case sensitive. For example, filtering by 'Linux' in title case give you different results from filtering by 'linux' in lower case.
5. Click **Add** under the **Actions** column, and type a **Name** and **Description** for your new value.
6. Click **Add** once again to add the new entry to the table.

Deleting a Company Tag

To delete a company tag:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then **Region**.
3. Select the **Tags** tab, then click on the **My Company Tags** tab.
4. Click **Delete** under the **Actions** column next to the tag to delete it.

   Note:

   When you delete a tag, the tag is also deleted from any resource to which it was assigned.
5. Click **OK** to confirm.

Importing Tags for Virtual Machines

You can import a **CSV** file with tag assignments into the VMDB. For the import to be successful, be aware of the following:

- The file must be in the following format, with one line for each virtual machine. One virtual machine per tag must be on a separate line even if you are assigning multiple tags of the same category.
- You must use the display names of the category and the display name for the tag for the import to work.

```
name,category,entry
evm2,Provisioning Scope,All
evm2,Exclusions,Do not Analyze
evm2,EVM Operations,Analysis Successful
rhel6,Department,Presales
rhel6,Department,Support
```

Importing Tags for a Virtual Machine from a CSV File

To import tags for a virtual machine from a CSV file:

1. Make sure the **CSV file** is in the required format.
2. Browse to **Settings > Application Settings**.
3. Click on the **Settings** accordion, then **Region**.
4. Select the **Tags** tab, then click on the **Import Tags** tab.
5. Click **Choose file** to go to the location where the file is located.
6. Click **Upload**.

   Note:

   If there are any problems with the file, such as an incorrect column name, unknown virtual machine, unknown tag, or multiple values for a tag that should have only one, an error message will appear in the console for those records.
7. Click **Apply**.

Importing Custom Values for Virtual Machines and Hosts

You can import a **CSV** file with asset tag information into the VMDB for a virtual machine or import custom values for hosts. For the import to be successful, the file must be in the following format, with one line for each virtual machine or host.

- There are two columns.
- The first line of the file must have the column names as shown below.
- The column names are case sensitive.

- Each value must be separated by a comma.

Virtual Machine Import Example

```
name,custom_1
Ecommerce,665432
Customer,883452
SQLSrvr,1090430
Firewall,8230500
```

For virtual machines, the value for custom_1 will show in the **VM Summary** page as the **Custom Identifier** page as the **Custom Identifier** in the **Properties** area. All of the custom values will show in the **Custom Fields** area.

Host Import Example

```
hostname,custom_1,custom_2
esx303.galaxy.local,15557814,19948399
esxd1.galaxy.local,10885574,16416993
esxd2.galaxy.local,16199125,16569419
```

For hosts, the value for custom_1 will show in the **Host Summary** page as the **Custom Identifier** in the **Properties** area. All of the custom values will show in the **Custom Fields** area.

Importing Asset Tags for a Virtual Machine from a CSV File

To import asset tags for a virtual machine from a CSV file

1. Make sure the **CSV file** is in the required format.
2. Browse to **Settings > Application Settings**.
3. Click on the **Settings** accordion, then **Region**.
4. Select the **Tags** tab, then click on the **Import Variables** tab.
5. Select the type of custom variable you want to import, either **Host** or **VM**.
6. Click **Choose file** to go to the location where the custom variable file is located.
7. Click **Upload**.

   Note:

   If there are any problems with the file, such as an incorrect column name, unknown virtual machine or host, a message appears.
8. Click **Apply**.

Tag Mapping

You can map labels from external providers to tags within Infrastructure Automation.

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then select the **Region**.
3. Select the **Tags** tab, then click on the **Map Tags** tab.
4. Click **Add** to create a tag map.
5. In the **Choose a resource entity and label**, select the **Entity** and enter a **Resource label**, then enter a **Tag Category**.
6. Click **Add**.

Customizing the Help Menu

Infrastructure Automation allows administrators to customize the help menu. Use this feature to define menu labels, URLs and how each window opens for users.

Note:

Any change to the help menu will take effect upon a full page reload.

Customize the help menu using the following steps:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then **Region**.
3. Click on the **Help Menu** tab.
4. Provide custom **Menu item labels** and an associated **URL** for each. Define how each window should open by selecting from the options in the **Open in** menu.
5. Click **Submit**.

Profiles

Creating an Analysis Profile

You can create an analysis profile by referring to the sample profiles provided in the console. You can copy the sample profile or create a new one.

Creating a Host Analysis Profile

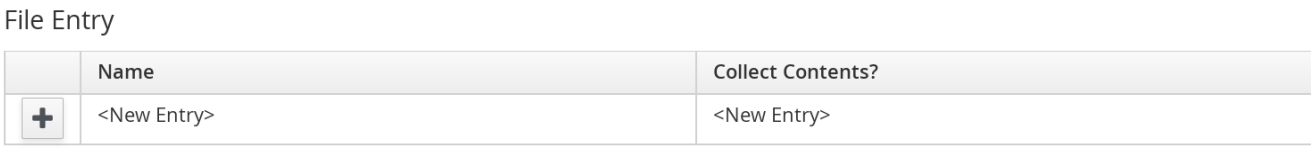
To create a host analysis profile:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Analysis Profiles**.

3. Click **Configuration**, and (**Add Host Analysis Profile**).
4. In the **Basic Information** area, type in a **Name** and **Description** for the analysis profile.



5. Click **File** to collect information about a file or group of files.
6. From the **File Entry** area, click (**Click to add a new entry**) to add a file or group of files.



   - Check **Collect Contents** to not only check for existence, but also gather the contents of the file. If you do this, then you can use the contents to create policies in Infrastructure Automation Control.
7. Click **Event Log** to specify event log entries to collect.
8. From the **Event Log Entry** area, click **Click to add a new entry** to add a type of event log entry. Type in a **Name**. You can type in a specific message to find in **Filter Message**. In **Level**, set the value for the level of the entry and above. Specify the **Source** for the entry. Finally, set the # number of days that you want to collect event log entries for. If you set this to 0, it will go as far back as there is data available.



9. Click **Add**.

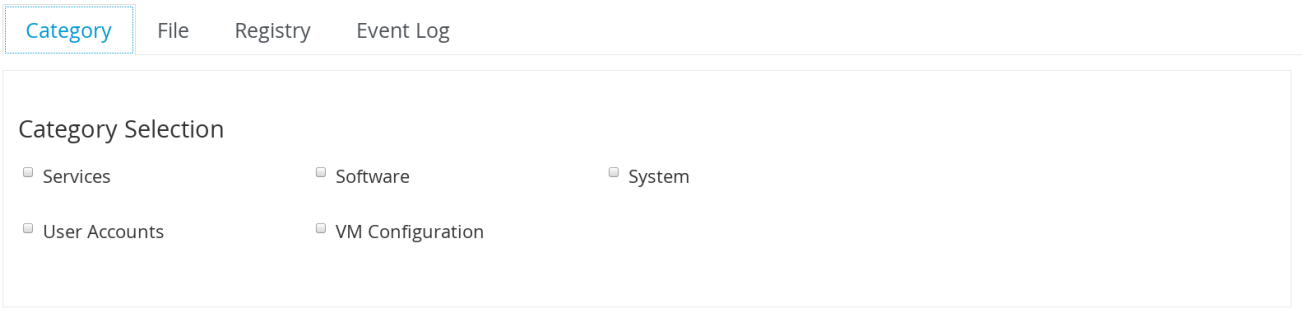
Creating a Virtual Machine Analysis Profile

To create a virtual machine analysis profile:

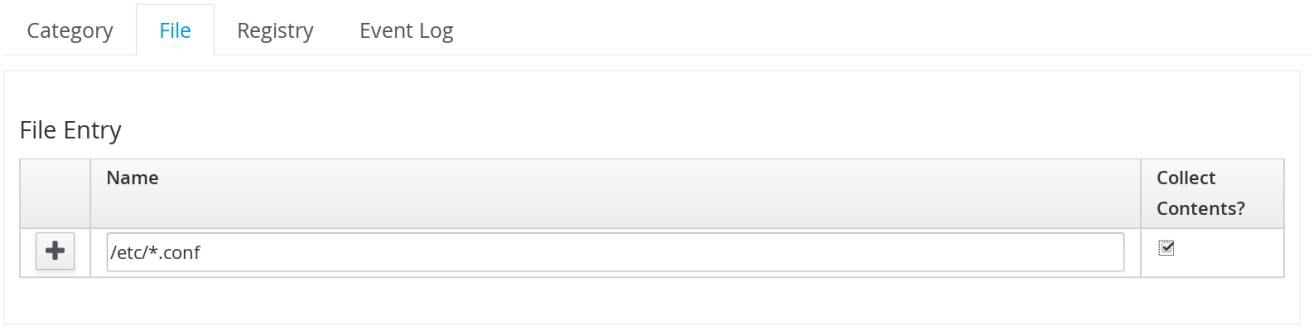
1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Analysis Profiles**.
3. Click **Configuration**, and **Add VM Analysis Profile**.
4. In the **Basic Information** area, type in a **Name** and **Description** for the analysis profile.



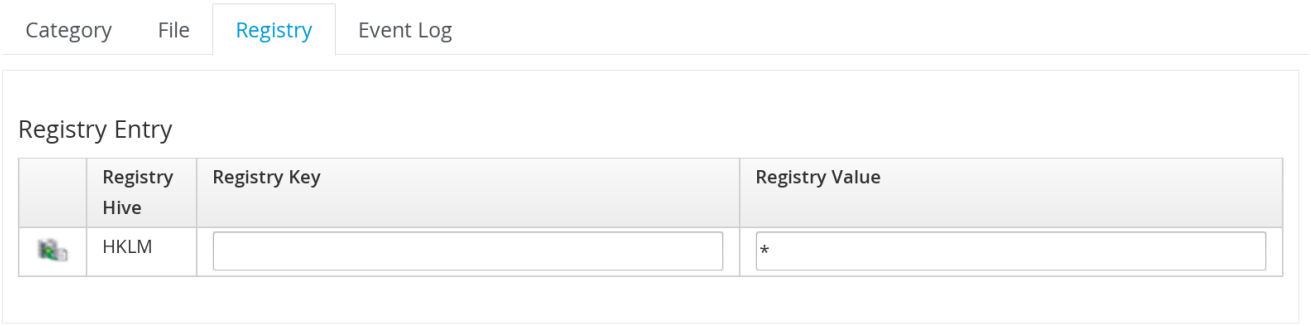
5. You begin in the **Category** tab. From the **Category Selection** area, check the categories you want to collect information for. This is available for virtual machine profiles only.



6. Click the **File** tab to collect information about a file or group of files.
7. From the **File Entry** area, click **Add this entry** to add a file or group of files, then type a name. For virtual machines, specify the file to check for. Check the box under **Collect Contents** if you want to collect the file contents as well. The files can be no larger than 1 MB.



8. Click the **Registry** tab to collect information on a registry key.
9. From the **Registry Entry** area, type your **Registry Key** and **Registry Value**. To evaluate whether a registry key exists or does not exist on a virtual machine, without providing a value, type * in the **Registry Value** field. Then, you do not need to know the registry value to collect the keys. This is available for virtual machine profiles only.



10. Click **Event Log** to specify event log entries to collect.
11. From the **Event Log Entry** area, complete the fields to add a type of event log entry. You can type in a specific message to find in **Filter Message**. In **Level**, set the value for the level of the entry and above. Specify the **Source for the entry**. Finally, set the # (number) of days that you want to collect event log entries for. If you set this to 0, it will go as far back as there is data available.



12. Click **Add**.

Editing an Analysis Profile

To edit an analysis profile:

1. Browse to **Settings > Application Settings**.

2. Click on the **Settings** accordion, then click **Analysis Profiles**.
3. Check the analysis profile you want to edit.
4. Click **Edit the selected Analysis Profiles**.
5. Make any changes.
6. Click **Save**.

The changes are added to the analysis profile. The virtual machines or hosts must be re-analyzed to collect the new or modified information.

Copying an Analysis Profile

To copy an analysis profile:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Analysis Profiles**.
3. Check the analysis profile you want to copy.
4. Click **Configuration**, then **Copy the selected Analysis Profiles**.
5. Type a new **Name** and **Description**.
6. Make required changes.
7. Click **Add**.

Setting a Default Analysis Profile

If you want to set an analysis profile to be used for all virtual machines, you can create a default profile.

To create a default analysis profile:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Analysis Profiles**.
3. Click on the analysis profile you want to set as the default.
4. Click **Configuration**, then **Edit the selected Analysis Profile**.
5. For a virtual machine profile, enter `default` in lower case in **Name**. For a host profile, enter `host default`.



6. Click **Save**.

Zones

You can organize your Infrastructure Automation Infrastructure into zones to configure failover and isolate traffic. A provider that is discovered by a server in a specific zone gets monitored and managed in that zone. All jobs, such as a SmartState Analysis or VM power operation, dispatched by a server in a specific zone can get processed by any Infrastructure Automation appliance assigned to that same zone.

Zones can be created based on your own environment. You can make zones based on geographic location, network location, or function. When first started, a new server is put into the default zone.

Suppose you have four Infrastructure Automation appliances with two in the East zone, appliances A and B, and two in the West zone, appliances C and D. VC East is discovered by one of the Infrastructure Automation appliances in the Infrastructure Automation Eastern zone. If Appliance A dispatches a job of analyzing twenty virtual machines, this job can be processed by either Appliance A or B, but not C or D.

Note:

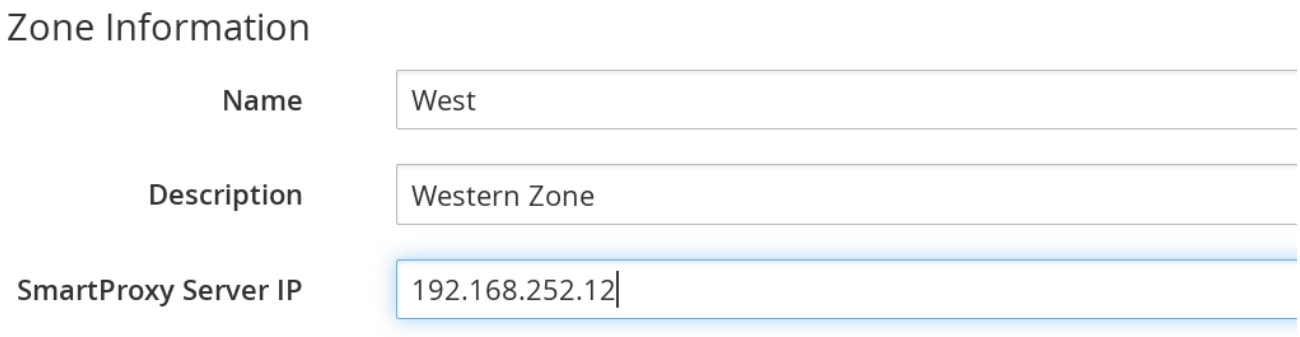
Only users assigned the super administrator role can create zones. There must always be at least one zone. The **Default Zone** is provided and cannot be deleted.

Creating a Zone

To create a zone:

1. Browse to **Settings > Application Settings**.

2. Click on the **Settings** accordion, then click **Zones**.
3. Click **Configuration**, and **Add a new Zone** to create a zone.
4. In the **Zone Information** area, type in a **Name** and **Description** for the new zone.



5. Use **SmartProxy Server IP** to specify the IP address of the server that you want SmartProxies installed in this zone to report to. If this is not set, then the IP address of the server that deployed the SmartProxy is used. This does not apply to embedded SmartProxies.
6. Optionally, you can configure **NTP servers** for the entire zone in the NTP Servers area. These settings will be used if the NTP servers have not been set for the appliance in the menu: **Operations > Server** page.
7. In the **Settings** area, set the number for **Max Active VM Scans**. The default is `Unlimited`.
8. Click **Save**.

Deleting a Zone

To delete a zone:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone you want to remove.

   Note:

   You cannot delete a zone if there are servers assigned to it.
4. Click **Configuration**, then click **Delete this Zone**.
5. Click **OK** to confirm.

Editing a Zone

To edit a zone:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone you want to edit.
4. Click **Configuration**, then click **Edit this Zone**.
5. Complete the required changes.
6. Click **Save**.

Adding SmartProxy Affinity to a Zone:

Enable SmartProxy Affinity for zones containing servers with the SmartProxy role to run a SmartState Analysis.

To add SmartProxy Affinity to a zone:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone in which you want to enable SmartProxy Affinity.
4. Click the **SmartProxy Affinity** tab and click the appropriate server.
5. Click **Save**.

Servers

Server settings enables you to control how each Infrastructure Automation server operates including authentication, logging, and email. If you have multiple servers in your environment that are reporting to one central VMDB, then you can edit some of these settings from the console by specifying which server you want to change.

Note:

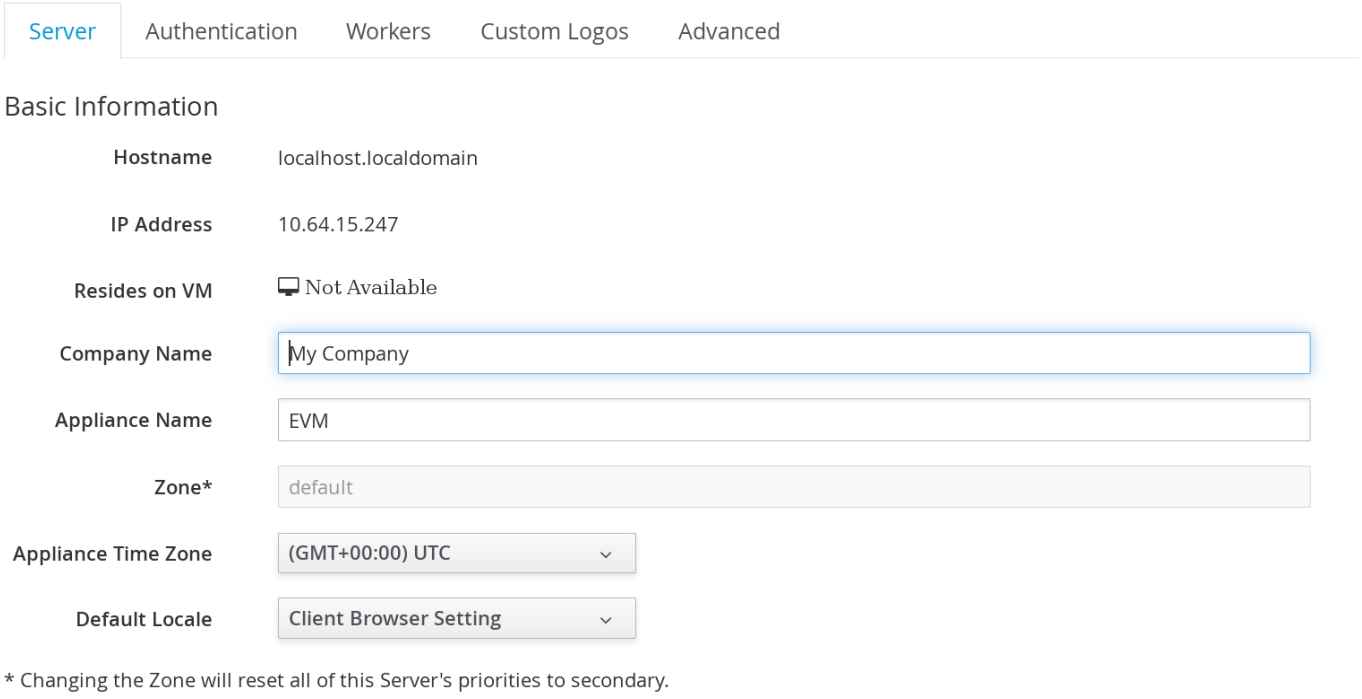
The server selection options are only available if you have multiple servers sharing one VMDB.

Changing Server Settings

To change server settings:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone where the Infrastructure Automation server is located.
4. In the **Servers** area, click on the Infrastructure Automation server.
5. Click **Server**.
6. Make any required changes.
7. Click **Save**.

Basic Information Settings



- Use **Company Name** (maximum 20 characters) to customize the interface with your company's name. You will see the company name when you are viewing or modifying the tags of an infrastructure object or virtual machine.
- Specify the **Appliance Name** (maximum 20 characters) you want displayed as the appliance that you are logged into. You will see this in the upper right corner of the interface with the name of the consoles logged on user.
- Use **Zone** to isolate traffic and provide load balancing capabilities. Specify the zone that you want this Infrastructure Automation appliance to be a member of. At startup, the zone is set to default.
- Use **Appliance Time Zone** to set the time zone for this server.

  Note:

  This is the time zone used when created scheduled analyses. This is not the same as the **Time Zone** parameter, which is found by navigating to the settings menu, then **My Settings**, then exploring the **Display Settings** area, and is the time zone displayed in the console.
- Use **Default Locale** to specify the default language for this server.

Server Control Settings

A server role defines what a server can do. Red Hat recommends that Database Operations, Event Monitor, Reporting, Scheduler, SmartState Analysis, User Interface, Provider Inventory, Provider Operations, and Web Services be enabled on at least one server in each zone. These roles are enabled by default on all servers.

- Use **Default Repository SmartProxy** to set the SmartProxy from which you refresh your virtual machine repositories. This host must have access to your repositories to analyze its virtual machines.

Note:

- Only super administrators can change server roles.
- If you are using more than one Infrastructure Automation appliance, be sure to set this on all of the appliances.

Server Roles

Note:

- Server roles that are in an active/active high availability configuration *(load balancing and failover protection)* are active in more than one location; whereas, roles that are in an active/passive *(primary/secondary in the case of Infrastructure Automation)* high availability configuration *(failover protection)*, if more than one Infrastructure Automation server in a specific zone or region has this role, only one will be active *(primary)* at a time and a failover has to occur to the passive *(secondary)* appliance with that role.
- For information on region and zone diagnostics and server role priorities, see Diagnostics.

Server Role	Description	Zone or Region Aware	Primary/Secondary or Active/Active
Automation Engine	Use this role if you want to use this Infrastructure Automation server to process automation tasks.	N/A	Active/Active
Capacity and Utilization Coordinator	The Capacity & Utilization Coordinator role checks to see if it is time to collect data, somewhat like a scheduler. If it is time, a job is queued for the Capacity and Utilization Data Collector. The coordinator role is required to complete Capacity and Utilization data collection. If more than one Infrastructure Automation server in a specific zone has this role, only one will be active at a time.	Zone	Primary/Secondary
Capacity & Utilization Data Collector	The Capacity & Utilization Data Collector performs the actual collection of capacity and utilization data. This role has a dedicated worker, and there can be more than one Infrastructure Automation server with this role in a zone.	Zone	Active/Active
Capacity & Utilization Data Processor	The Capacity & Utilization Data Processor processes all of the data collected, allowing Infrastructure Automation to create charts. This role has a dedicated worker, and there can be more than one Infrastructure Automation server with this role in a zone.	Zone	Active/Active
Database Operations	Use Database Operations to enable this Infrastructure Automation server to run database backups or garbage collection.	Zone	Active/Active
Embedded Ansible	This role is disabled by default. The Embedded Ansible role supports Ansible Automation Inside functionality. Enable this role to configure playbook repositories and run playbooks natively to back service catalog items. **NOTE**: Enable the Provider Inventory server role in the same zone as the Embedded Ansible server role to ensure proper functionality.	Region	Primary/Secondary
Event Monitor	This role is enabled by default and provides the information shown in timelines. The Event Monitor is responsible for the work between the Infrastructure Automation server and your providers. It starts 2 workers for each provider. One worker, the monitor, is responsible for maintaining a connection to a provider, catching events, and putting them on the Infrastructure Automation message queue for processing. The second worker, the handler, is a message queue worker responsible for delivering only those messages for a provider. You should have at least one of these in each zone.	Zone	Primary/Secondary
Git Repository	The Git Repositories Owner server role supports importing domains into automate from a git repository. This feature is available from the **Automate > Import/Export** screen in the Infrastructure Automation user interface.	Region	Primary/Secondary
Notifier	Use this role if you will be using Infrastructure Automation Control or Automate to forward SNMP traps to a monitoring system or send e-mails. See Creating an Alert for details on creating SNMP alerts. If more than one Infrastructure Automation server in a specific region has this role, only one will be active at a time.	Region	Primary/Secondary
Provider Inventory	This role is enabled by default. This role is responsible for refreshing provider information including EMS, hosts, virtual machines, and clusters, and is also responsible for capturing datastore file lists. If more than one Infrastructure Automation server in a specific zone has this role, only one will be active at a time. Required in the same zone as a Infrastructure Automation appliance with the Embedded Ansible role enabled.	Zone	Primary/Secondary
Provider Operations	This role is enabled by default. This role sends stop, start, suspend, shutdown guest, clone, reconfigure, and unregister to the provider, directly from the console or through a policy action if you have Infrastructure Automation Control. More than one Infrastructure Automation server can have this role in a zone.	Zone	Active/Active
RHN Mirror	An appliance with RHN Mirror enabled acts as a server containing a repository with the latest Infrastructure Automation packages. This also configures other appliances within the same region to point to the chosen RHN Mirror server for updates. This provides a low bandwidth method to update environments with multiple appliances.	N/A	Active/Active
Reporting	This role is enabled by default. The Reporting role specifies which Infrastructure Automation servers can generate reports. If you do not have a Infrastructure Automation server set to this role in a zone, then no reports can be generated in that zone. You should have at least one of these in each zone.	Zone	Active/Active
Scheduler	This role is enabled by default. The Scheduler sends messages to start all scheduled activities such as report generation and SmartState analysis. This role also controls all system schedules such as capacity and utilization data gathering. One server in each region must be assigned this role or scheduled Infrastructure Automation events will not occur. If more than one Infrastructure Automation server in a specific region has this role, only one will be active at a time.	Region	Primary/Secondary
SmartProxy	Enabling the SmartProxy role turns on the embedded SmartProxy on the Infrastructure Automation server. The embedded SmartProxy can analyze virtual machines that are registered to a host and templates that are associated with a provider. To provide visibility to repositories, install the SmartProxy on a host from the Infrastructure Automation console. This SmartProxy can also analyze virtual machines on the host on which it is installed. Enabling the SmartProxy role on an appliance requires selecting the SmartProxy Affinity for a zone to run a SmartState Analysis. By default, no selections are enabled under SmartProxy Affinity.	Zone	Active/Active

Server Role	Description	Zone or Region Aware	Primary/Secondary or Active/Active
SmartState Analysis	This role is enabled by default. The SmartState Analysis role controls which Infrastructure Automation servers can control SmartState Analyses and process the data from the analysis. You should have at least one of these in each zone.	Zone	Active/Active
User Interface	This role is enabled by default. The Web Services role must also be enabled with this role to log into the user interface, as the User Interface role queries the API to receive tokens for login. Uncheck User Interface if you do not want users to be able to access this Infrastructure Automation server using the Infrastructure Automation console. For example, you may want to turn this off if the Infrastructure Automation server is strictly being used for capacity and utilization or reporting generation. More than one Infrastructure Automation server can have this role in a zone.	Zone	Active/Active
Web Services	This role is enabled by default. The Web Services roles provides API access and must be enabled if the User Interface role is enabled to log into the user interface. You can also enable the Web Services role to provide API-only access to the server. Uncheck Web Services to stop this Infrastructure Automation server from acting as a web service provider. More than one Infrastructure Automation server can have this role in a zone.	N/A	Active/Active
Websocket	This role enables starting or stopping websocket workers required for proxying remote consoles.	N/A	Active/Active

VMware Console Settings

If you are using the Infrastructure Automation control feature set, then you have the ability to connect to a Web console for virtual machines that are registered to a host. To use this feature, you must have VNC installed, VMware's WebMKS SDK enabled in Infrastructure Automation, or the VMRC native desktop application installed for your environment.

Note:

See vendor documentation to ensure you are installing appropriate applications for your virtual infrastructure. Once you have installed the required software, you must specify its version in the Infrastructure Automation configuration settings.



- If you select **VNC**, type in the port number used. This port must be open on the target virtual machine and the VNC software must be installed there. On the computer that you are running the console from, you must install the appropriate version of Java Runtime if it is not already installed.
- If you select **VMware WebMKS**, select the appropriate version.
- If using **VMware VMRC** desktop application, be sure that you have fulfilled the requirements for your vCenter version. The correct version of the VMRC desktop application from VMware must be installed on the client computer. To do this, log into the vCenter Web Service and attempt to open a virtual machine console. The vSphere Web Client must be installed on vCenter version 5, and the provider must be registered to it. For vCenter version 4, the VMware vCenter Management Webservices must be running.

NTP Servers Settings In the NTP Servers area, you can

specify the NTP servers to use as source for clock synchronization here. The NTP settings specified here will override Zone NTP settings. Enter one NTP server hostname or IP address in each text box.

Configuring SNMP

You can use Simple Network Management Protocol (SNMP) traps to send alerts for various aspects of a Infrastructure Automation environment.

Requirements

- Configure your SNMP management station to accept traps from Infrastructure Automation appliances. Consult your management station's documentation.
- Each appliance that could process SNMP traps must have the **`snmpd`** and **`snmptrapd`** daemons running.
- The region where the appliances are located must have the **`Notifier`** role enabled and the failover role priority set.

To enable the **`snmpd`** and **`snmptrapd`** daemons

1. Access each SNMP processing appliance using SSH.

2. Set the SNMP daemons to run on start up:

```
# chkconfig --level 2345 snmpd on
# chkconfig --level 2345 snmptrapd on
```

3. The daemons run automatically when the appliance is restarted, but must be started manually now.

```
# service snmpd start
# service snmptrapd start
```

To enable the notifier role:

1. Access each SNMP processing appliance using their web interfaces.
2. Browse to **Settings > Application Settings**.
3. Click on the **Settings** accordion, then click **Zones**. Select the zone where the EVM server is located, and select the EVM server.
4. In the **Server Control** area, set the **Notifier** server role option to **ON**.
5. Click **Save**.

To set the failover priority role:

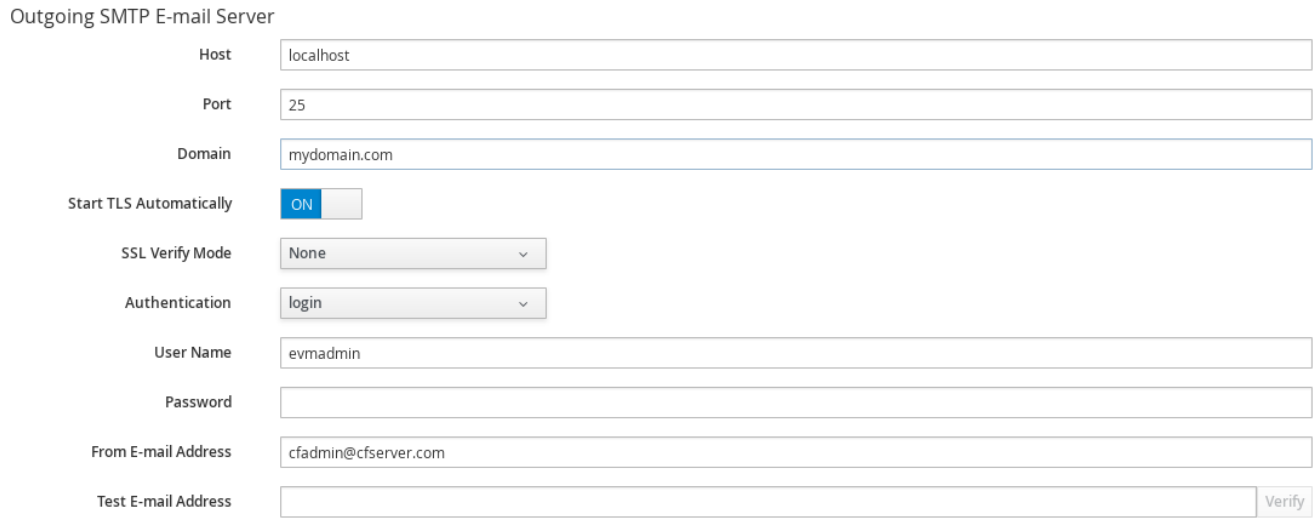
1. Browse to **Settings > Application Settings** then click the **Diagnostics** accordion.
2. Select the zone where the EVM server is located.
3. Click **Roles by Servers** or **Servers by Roles** to view your servers.
4. In the **Status of Roles for Servers** in **Zone Default Zone** area, click the role that you want to set the priority for.
5. Click **Configuration**, and **Promote Server** to make this the primary server for this role.

Outgoing SMTP Email Settings

To use the email action in Infrastructure Automation, set an email address to send emails from.

Note:

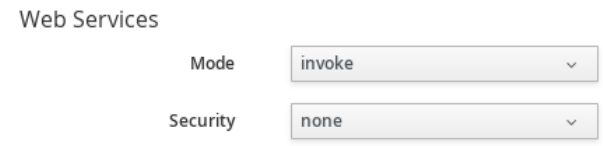
To be able to send any emails from the server, you must have the Notifier server role enabled. You can test the settings without the role enabled.



- Use **Host** to specify the host name of the mail server.
- Use **Port** to specify the port for the mail server.
- Use **Domain** to specify the domain name for the mail server.
- Set **Start TLS Automatically** on **ON** if the mail server requires TLS.
- Select the appropriate **SSL Verify Mode**.
- Use the **Authentication** drop down to specify if you want to use **`login`**, **`plain`**, or no authentication.
- Use **User Name** to specify the user name required for login authentication.
- Use **Password** to specify the password for login authentication.
- Use **From E-mail Address** to set the address you want to send the email from.
- Use **Test E-mail Address** if you want to test your email settings. Click **Verify** to send a test email.

Web Services Settings

Web services are used by the server to communicate with the SmartProxy.



- Set **Mode** to invoke to enable 2-way Web services communication between the Infrastructure Automation appliance and the SmartProxy. Set **Mode** to disabled to use Web services from the SmartProxy to the Infrastructure Automation appliance only. When the Infrastructure Automation appliance has work for the SmartProxy, the work will be placed in a queue in the VMDB. The work will be completed either when the Infrastructure Automation appliance is able to contact the SmartProxy or when the next SmartProxy heartbeat occurs, whichever comes first.
- If **Web Services** are enabled, you have the option to use **ws-security**.

Logging Settings



- Use **Log Level** to set the level of detail you want in the log. You can select from **fatal**, **error**, **warn**, **info**, and **debug**. The default setting is 'info'.

Custom Support URL Settings



- Use **URL** to specify a specific URL that you want to be accessible from the **About Product Assistance** area.
- Use **Description** to set a label for the **URL**.

Authentication

Use the **Authentication** tab to specify how you want users authenticated on the console. You can use the VMDB or integrate with LDAP, LDAPS, Amazon, or an external IPA server.

Changing Authentication Settings

To change authentication settings:

1. Browse to **Settings > Application Settings**.
2. Click the **Settings** accordion, then click **Zones**.
3. Click the zone where the server is located.
4. Click the server.
5. Click the **Authentication** tab.
6. Use **Session Timeout** to set the period of inactivity before a user is logged out of the console.
7. Set the authentication method in **Mode**.
8. Click **Save**.

Workers

Use the Workers page to specify the number of workers and amount of memory allowed to be used for each type.

Note:

Only make these changes when directed to by Red Hat Support.

Changing Settings for a Worker

To change the settings for a worker

1. Browse to **Settings > Application Settings**.

2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone where the server is located.
4. Click on the server.
5. Click **Workers**.
6. Go to the type of worker you have been directed to change.
7. If applicable, change Count or Memory Threshold using the dropdown boxes.
8. Click **Save**.

Database

Use the Database page to specify the location of your Virtual Machine Database (VMDB) and its login credentials. By default, the type is PostgreSQL on the Server.

Note:

The server may not start if the database settings are changed. Be sure to validate your new settings before restarting the server.

Changing a Database Setting

To change a database setting:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone where the server is located.
4. Click on the server.
5. Click the **Database** tab.
6. In the **Database** area, select the **Type** of database. You can select from **External Database on another CFME appliance**, **External Postgres Database**, and **Internal Database on this CFME Appliance**.
    - Use **Hostname** to specify the IP address or hostname of the external database server.
    - Use **Database Name** to specify the name of your VMDB.
    - Specify the **User Name** to connect to the VMDB.
    - Use **Password** and **Verify Password** to specify the password for the user name.
7. Click **Validate** to check the settings.
8. Click **Save**.
9. Click **OK** to the warning that the server will restart immediately after you save the changes.

During the restart, you are unable to access the server. When the restart is complete, the new database settings are in effect.

Customization and Logos

Custom Logos

Use **Custom Logos** to display your own logo in the corner of the Infrastructure Automation user interface and on the login screen. Use the procedures below to upload a custom logo to the user interface, and to customize the login background and login panel text on the user interface.

Note:

- If you have upgraded from an earlier Infrastructure Automation version and your custom logo was already in use before migration, although your logo image file is still in place in `vmdb/public/upload` you may have to uncheck and recheck the option to **Use Custom Logo Image** to re-enable displaying your custom logo. See Uploading a Custom Logo to the User Interface for the procedure on how to access the **Use Custom Logo Image** option, or if you want to upload another custom logo to the user interface and customize the login background image and login panel text.
- Additionally, ensure the option to use configuration settings for the tenant under **Access Control** is set to `Yes`; see Displaying the Custom Configuration Settings for the procedure on how to set the configuration settings.

Uploading a Custom Logo to the User Interface

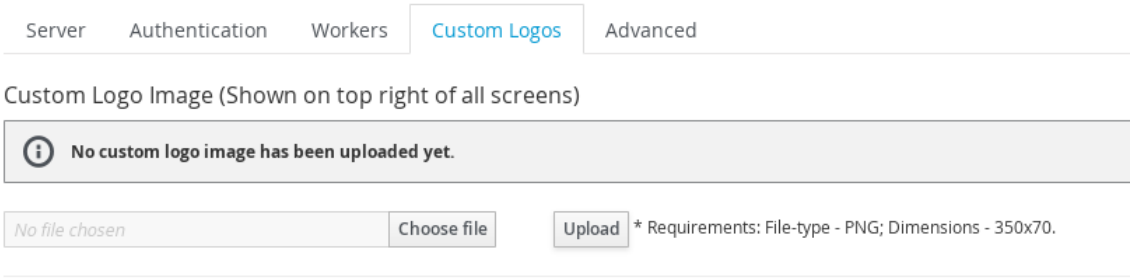
Note:

Make sure the desired logo is accessible from the computer where you are running the Infrastructure Automation user interface. The file must be in portable network graphics (png) format with dimensions of 350 px x 70 px.

To upload a custom logo to the user interface:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.

3. Click the zone where the Infrastructure Automation server is located.
4. Click on the server.
5. Click the **Custom Logos** tab.



6. In **Custom Logo Image (Shown on top right of all screens)**, click **Choose file** to go to the location where the logo file is located.
7. Click **Upload**. The icon is displayed above the file name box, and an option is shown to use the logo.
8. Check **Use Custom Logo Image** to add the logo to your user interface.
9. Click **Save**.

Note:

To enable displaying your custom logo, ensure the option to use configuration settings for the tenant under **Access Control** is set to `Yes`. See simpara_title for the procedure on how to set the configuration settings.

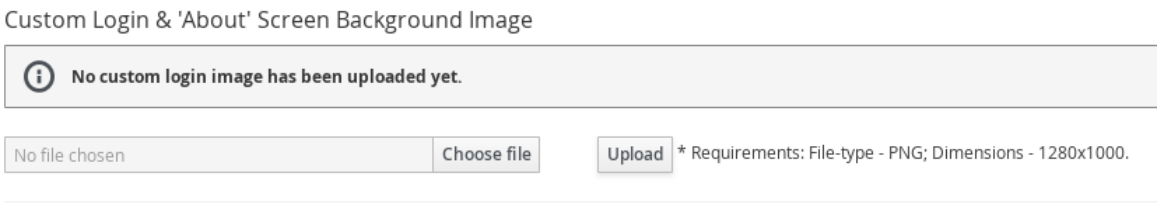
Customizing the Login Background

Note:

Make sure the background image that you want to use is accessible from the computer where you are running the user interface. The file must be in PNG format with dimensions of 1280 px x 1000 px.

To customize the login background:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone where the server is located.
4. Click on the server.
5. Click the **Custom Logos** tab.
6. In **Custom Login & 'About' Screen Background Image**, click **Choose file** to go to the location where the background image file is located.



7. Click **Upload**. The icon is displayed above the file name box, and an option is shown to use the logo.
8. Check **Use Custom Login Background Image** to add the background image to the login screen of the user interface.
9. Click **Save**.

Customizing the Login Panel Text

To customize the login panel text:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone where the server is located.
4. Click on the server.
5. Click the **Custom Logos** tab.
6. In **Custom Login Panel Text**, enter the text that you want to display on the login screen.

7. Click **Use Custom Login Text** to switch it to **Yes**.



8. Click **Save**.

Displaying the Custom Configuration Settings

To enable displaying your custom logo in the corner of the Infrastructure Automation user interface and on the login screen:

1. Browse to **Settings > Application Settings**.
2. Click the **Access Control** accordion.
3. Click **Tenants**, then click **My Company**.
4. Click (**Configuration**), then click (**Edit this item**).



5. Click **Use Configuration Settings** to switch it to **Yes**.
6. Click **Save**.

Advanced Settings

You may be instructed by Red Hat to edit some configuration settings manually. This feature is available for a limited number of options and can only be used by users assigned the super administrator role. Changing settings using this procedure may disable your Infrastructure Automation server.

- Only make manual changes to your configuration files if directed to do so by Red Hat. - Infrastructure Automation provides the ability to edit advanced configuration settings for all appliances through the **Advanced** tab from the user interface. You can also specify advanced settings for entire zones by clicking the **Advanced** tab at the zone level, with the advanced settings applied to individual appliances in that zone overriding zone-level settings when set.

Editing Configuration Files Manually

To edit configuration files manually:

1. Click **Configuration**.
2. Click on the **Settings** accordion, then click **Zones**.
3. Click the zone where the server is located.
4. You can click the **Advanced** tab at the zone level to set advanced configuration settings for the entire zone. Or, you can click on the server, and then click the **Advanced** tab to set advanced configuration settings at the server (appliance) level, which will override zone-level settings.
5. Make the required changes to the configuration file.
6. Click **Save**.

Configuration Parameters

Table: authentication

Parameters	Description
amazon_key	If using Amazon for the authentication mode, specify your Amazon Key. This is the same as Amazon Access Key in Configuration-Operations-Server-Amazon Settings in the appliance console. Default: blank
amazon_secret	If using Amazon for the authentication mode, specify your Amazon Secret. This is the same as Amazon Secret Key in Configuration-Operations-Server-Amazon Settings in the appliance console. Default: blank
basedn	If using ldap for the authentication mode, specify your Base DN. This is the same as Base DN in Configuration-Operations- Server-LDAP Settings in the appliance console. Default: blank

Parameters	Description
bind_dn	The user name to bind to the LDAP server. This user must have read access to all users and groups that will be used for Infrastructure Automation authentication and role assignment. This is the same as Bind DN in Configuration-Operations-Server-LDAP Settings in the appliance console. Default: blank
bind_pwd:	The password for the bind_dn user. This is the same as Bind Password in Configuration-Operations- Server-LDAP Settings in the appliance console. Default: blank
get_direct_groups	Use this to get the LDAP roles from the LDAP users' home forest. This is the same as Get Roles from Home Forest in the Authentication page for the Infrastructure Automation Server. Default: true
group_memberships_max_depth	When traversing group memberships in the LDAP directory it will stop at this value. Default: 2
ldaphost	Use ldaphost to specify the fully qualified domain name of your LDAP server. This is the same as LDAP Host Name in Configuration-Operations-Server-LDAP Settings in the appliance console. Default: blank
ldapport	Specify the port of your LDAP server. This is the same as LDAP Port in Configuration-Operations- Server-LDAP Settings in the appliance console. Default: 389
mode	Use database to use the VMDB for security. Use ldap or ldaps to use directory services. This is the same as Mode in Configuration-Operations-Server-Authentication in the appliance console. Default: database
user_type	Use userprincipalname to type the user name in the format of user@domainname. Use mail to login with the user's e-mail address. Use dn-cn for Distinguished Name (CN=<user>) or dn-uid Distinguished Name (UID=<user>) to use just the user name, but be sure to enter the proper user_suffix for either one. This is the same as User Type in Configuration-Operations- Server-LDAP Settings in the appliance console. Default: userprincipalname
user_suffix	Domain name to be used with user_type of dn-cn or dn-uid. This is the same as User Suffix in Configuration-Operations- Server-LDAP Settings in the appliance console. Default: blank

Table: coresident_miqproxy

Parameters	Description
use_vim_broker	Specify if you want the coresident SmartProxy to use a shared connection through the VIM broker to communicate with the VC or ESX host for SmartState Analysis. If it is disabled, then each SmartProxy SmartState Analysis would create its own connection. Default: true
concurrent_per_ems	Specify the number of co-resident SmartProxy SmartState Analyses that can be run against a specific management system at the same time. Default: 1
concurrent_per_host	Specify the number of co-resident SmartProxy SmartState Analyses that can be run against a specific host at the same time. Default: 1
scan_via_host	If you change scan_via_host to false, Infrastructure Automation will use the Management System to scan which is limited by the concurrent_per_ems setting instead of the concurrent_per_host setting. Note this will greatly increase traffic to the Management System. Default: true

Table: ems_refresh

Parameters	Description
capture_vm_created_on_date	Set to false to turn off historical event retrieval. Set to true to turn on. By setting the flag to true Infrastructure Automation will try to set the "ems_created_on" column in the vms table after an ems refresh for new VMs and any VMs with a nil "ems_created_on" value. Infrastructure Automation looks at event information in our database as well as looking up historical event data from the management system. This is optional since the historical lookup could timeout. Default: false
collect_advanced_settings	Set to false if you do not want to collect advanced Virtual Machine settings during a management system refresh. This will increase the speed of the refresh, but less data will be collected. If the parameter is not listed, then the value is true. Default: true
ec2	
get_private_images	For EC2 refreshes only; whether or not to retrieve private images. Default: true
get_public_images	For EC2 refreshes only; whether or not to retrieve public images. Default: false. Warning: setting get_public_images to true loads several thousand images in the VMDB by default and may cause performance issues.
get_shared_images	For EC2 refreshes only; whether or not to retrieve shared images. Default: true
public_images_filters	For EC2 refreshes only; a filter to reduce the number of public images. Default: all public images
ignore_terminated_instances	For EC2 refreshes only; whether or not to ignore terminated instances. Default: true
full_refresh_threshold	The number of targeted refreshes requested before they are rolled into a full refresh. For example, if the system and/or the user target a refresh against 7 VMs and 2 Hosts (9 targets), when the refresh actually occurs it will do a partial refresh against those 9 targets only. However, if a 10th had been added, the system would perform a full EMS refresh instead. Default: 100
raise_vm_snapshot_complete_if_created_within:	Raises vm_snapshot_complete event for a snapshot being added to VMDB only if the create time in Virtual Center is within the configured period of time. This prevents raising events for old snapshots when a new VC is added to Infrastructure Automation. Default: 15.minutes
refresh_interval	Scheduler does a periodic full EMS refresh every refresh_interval. Default: 24.hours

Table: host_scan

Parameters	Description
queue_timeout	Time period after which a host SmartState analysis will be considered timed out. Default: 20.minutes

Table: log

Parameters	Description
level	Specify the required level of logging for the Infrastructure Automation appliance. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. This is the same as Log Level in Configuration-Operations-Server-Logging in the appliance console and applies immediately to the evm log. Default: info

Parameters	Description
level_aws	Specify the level of logging for Amazon Web Services communications. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. This applies to the aws log. Default: info
level_aws_in_evm	Specify what level of Amazon Web Services communication log should be also shown in evm log. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. Default: error
level_fog	Specify the level of logging for Fog communications. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. This applies to the fog log. Default: info
level_fog_in_evm	Specify what level of Fog communication log should be also shown in evm log. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. Default: error
level_rails	Specify the level of logging for Rails. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. Once changed, this applies immediately to the production log. Default: info
level_rhevm	Specify the level of logging for Red Hat communications. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. This applies to the rhevm log. Default: warn
level_rhevm_in_evm	Specify what level of Red Hat communication log should be also shown in evm log. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. Default: error
level_vim	Specify the level of logging for VIM (communication with VMware ESX and Virtual Center). Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. This applies to the vim log. Default: warn
level_vim_in_evm	Specify what level of vim logging should be also shown in the evm log. Possible levels from most detailed to least detailed are: debug, info, warn, error, fatal. Default: error

Table: db_stats

Parameters	Description
enabled	Specify if you want to keep track of the number of queries, size of queries, number of responses, size of response, min/max for each, number of established connections at for each server process. This information will show in the EVM log. Default: false
log_frequency	How frequently in seconds the process will log the database statistic in seconds. Default: 60

Table: callsites

Parameters	Description
enabled	Specify if you want keep track of the code that is accessing the database. Enabling call sites will decrease performance because of the amount of information tracked. The db_stats: enabled parameter must be set to true to use this. Default: false
depth	Specify how many levels in the call stack to track for each database access. Default: 10
min_threshold	Do not keep track of code that does not access the database this many times per log_frequency. Default: 10
path	Set the path for the Infrastructure Automation appliance log. This is the same as Log Path in Configuration-Operations- Server-Logging in the appliance console. Default: If no value is present, the path is /var/www/miq/vmdb/log.
line_limit	Limit how many characters are retained in a single log line. 0 means no limit. Default: 0

Table: collection

Parameters	Description
ping_depot	Whether to use TCP port ping to the log depot before performing log collection. Default: true
ping_depot_timeout	Specify how long in seconds to wait for response from log depot before deciding that the TCP port ping failed. Default: 20
current	When collecting logs, specifies what is considered current logging as opposed to archived logging. Default: :pattern: log/*.log log/apache/*.log log/*.txt config/* /var/lib/pgsql/data/*.conf /var/lib/pgsql/data/log/* /var/log/syslog* /var/log/daemon.log* /etc/default/ntp* /var/log/messages* /var/log/cron* BUILD GUID VERSION
archive	Specifies what is considered archived logging. The default pattern is blank which means *.gz files in the log directory.

Table: log_depot

Parameters	Description
uri	Specify the URI for the log depot. This is the same as the URI in **Configuration > Diagnostics > Collect Logs** in the appliance console. Default: blank

Parameters	Description
username	Specify the user name for the log depot. This is the same as the user ID in **Configuration > Diagnostics > Collect Logs** in the appliance console. Default: blank
password	Specify the password for the user for the log depot. This is the same as the password in **Configuration > Diagnostics > Collect Logs** in the appliance console. Default: blank

Table: performance

Parameters	Description
capture_threshold	
vm	Amount of time in minutes to wait after capture before capturing again. Default: 50.minutes
host	Amount of time in minutes to wait after capture before capturing again. Default: 50.minutes
ems_cluster	Amount of time in minutes to wait after capture before capturing again. Default: 50.minutes
storage	Amount of time in minutes to wait after capture before capturing again. Default: 120.minutes
capture_threshold_with_alerts	
host	Amount of time in minutes to wait after capture before capturing again. This value is used instead of capture_threshold for Hosts that have alerts assigned based on real time Capacity & Utilization data. Default: 20.minutes
ems_cluster	Amount of time in minutes to wait after capture before capturing again. This value is used instead of capture_threshold for clusters that have alerts assigned based on real time Capacity & Utilization data. Default: 50.minutes
vm	Amount of time in minutes to wait after capture before capturing again. This value is used instead of capture_threshold for VMs that have alerts assigned based on real time Capacity & Utilization data. Default: 20.minutes
concurrent_requests	
vm	Amount of time in minutes to wait after capture before capturing again. This value is used instead of capture_threshold for VMs that have alerts assigned based on real time Capacity & Utilization data. Default: 20.minutes
hourly	Number of concurrent VC requests to make when capturing hourly raw metrics. Default: 1
realtime	Number of concurrent VC requests to make when capturing real time raw metrics. Default: 20
history	
initial_capture_days	How many days to collect data for on first collection. Default: 0
Keep_daily_performances	How long to keep daily performance data in the VMDB. Default: 6.months
keep_realtime_performances	How long to keep realtime performance data in the VMDB. Default: 4.hours
keep_hourly_performances	How long to keep hourly performance data in the VMDB. Default: 6.months
purge_window_size	When the purge needs to delete rows which are older than the keep_realtime_performances, keep_hourly_performances, and keep_daily_performances values, this value sets how many rows to delete in each batch. For example, a value of 1000 will cause us to issue ten 1,000 row deletes. Default: 1000

Table: repository_scanning

Parameters	Description
defaultsmartproxy	Specify the SmartProxy for repository scanning. This is the same as Default Repository Smartproxy in Configuration-Operations- Server-VM Server Control in the appliance console. Default: blank

Table: server

Parameters	Description
case_sensitive_name_search	Specify if you want the search by name on configuration item screens to be case senstive. Default: false
company	Specify the label you want to use for your company’s tagging. This is the same as Company Name in Configuration-Operations- Server-Basic Info. Default: “My Company”
custom_logo	Specify if you want to use a custom logo. This is the same as Use Custom Logo in Configuration-Custom Logo-Logo Selection. Default: false
events	
disk_usage_gt_percent	For Infrastructure Automation operational alerts, specify at what threshold the disk usage alerts will be triggered. Default: 80
heartbeat_timeout	How long to wait until the server heartbeat is considered timed out. if the timeout is exceeded, other appliances in the zone/region can vie for the roles active on the timed out Infrastructure Automation appliance. Default: 2.minutes
host	Infrastructure Automation Server’s IP address. Default: blank
hostname	Infrastructure Automation Server’s hostname. Default: localhost.localdomain
listening_port	Specify the port number on which the web server is listening. Note that this does not set the port that VMDB listens on. When deploying the SmartHost from the Infrastructure Automation appliance, it tells the SmartHost (miqhost) what port to talk to the VMDB on. Default: “443”
mks_version	Specify the version of the VMware MKS Plugin to use for the VM Console. This is the same as VMware MKS Plugin Version in Configuration-Operations- Server-VM Console. Default : 2.1.0.0
name	Set the name to display for the Infrastructure Automation appliance that you are logged on to in the appliance console. This is the same as appliance Name in Configuration-Operations- Server-Basic Information. Default : EVM

Parameters	Description
role	Specify the roles for this Infrastructure Automation Server, separated by commas without spaces. The possible values are automate, database_operations, ems_inventory, ems_metrics_collector, ems_metrics_coordinator, ems_metrics_processor, ems_operations, event, notifier, reporting, scheduler, smartproxy, smartstate, user_interface, web_services. This is the same as Server Roles in Configuration-Operations- Server- Server Control. Default: database_operations, event, reporting, scheduler, smartstate, ems_operations, ems_inventory, user_interface, web_services
session_store	Where to store the session information for all web requests. The possible values are sql, memory, or cache. SQL stores the session information in the database regardless of the type of database server. Memory stores all the session information in memory of the server process. Cache stores the information in a memcache server. Default: cache
startup_timeout	The amount of time in seconds that the server will wait and prevent logins during server startup before assuming the server has timed out starting and will redirect the user to the log page after login. Default: 300
vnc_port	If using VNC for remote console, the port used by VNC. Default: 5800
zone	Set the Zone for this appliance belongs. This is the same as Zone in Configuration-Operations- Server-Basic Information. Default : default
:worker_monitor	Starts and monitors the workers. Parameters specified here will override those set in the workers:default section.
poll	How often the worker monitor checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 15.seconds
miq_server_time_threshold	How much time to give the server to heartbeat before worker monitor starts to take action against non-responding server. Default: 2.minutes
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 1
sync_interval	Time interval to sync active roles and configuration for all workers. Default: 30.minutes
wait_for_started_timeout	How long to wait for a started worker to heartbeat before considering the worker timed out. Default: 10.minutes
kill_algorithm	
name	Criteria used to start killing workers. Default: used_swap_percent_gt_value
value	Value of the criteria used. Default: 80
start_algorithm	
name	After server startup, criteria that must be met to decide if the Infrastructure Automation Server can start a new worker. Default: used_swap_percent_lt_value
value	Value of criteria used. Default: 60

Table: session

Parameters	Description
interval	Set the time interval in seconds for checking inactive sessions in appliance console. Default: 60
timeout	Set the time period in seconds in which inactive console sessions are deleted. This is the same as Session Timeout in Configuration-Operations-Server-Authentication in the appliance console. Default: 3600
memcache_server	If you choose memory for session_store, you need to specify the memcache_server to retrieve the session information from. Default: 127.0.1.1:11211
memcache_server_opts	Options to send to memcache server. : blank
show_login_info	Specify whether or not you want to see login info on start page. Default: true

Table: smartproxy_deploy

Parameters	Description
queue_timeout	Timeout for host smartproxy deploy job. Default: 30.minutes

Table: smtp

Parameters	Description
host	Specify the hostname of the smtp mail server. This is the same as Host in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: localhost
port	Specify the port of the smtp mail server. This is the same as Port in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: “25”
domain	Specify the domain of the smtp mail server. This is the same as Domain in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: mydomain.com
authentication	Specify the type of authentication of the smtp mail server. This is the same as Authentication in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: login
user_name	Specify the username required for login to the smtp mail server. This is the same as User Name in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: evmadmin
password	Specify the encrypted password for the user_name account. This is the same as Password in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: blank
from	Set the address that you want to send e-mails from. This is the same as From E-mail Address in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: cfadmin@cfserver.com

Table: snapshots

Parameters	Description
create_free_percent	Ensures the % of free space available on the main datastore (datastore where vmx file is located) can support the % growth of the snapshot. The default is to require space for 100% of the provisioned size of all disks that are taking part in the snapshot. A value of 0 means do not check for space before creating the snapshot. Default: 100

Parameters	Description
remove_free_percent	Ensures the % of free space available on the main datastore (datastore where vmx file is located) has the % free space available to support the snapshot deletion process. Note that the deletion process consists of first composing a new snapshot then removing it once the original snapshot to be deleted has been collapsed in the VM. The default is to require 100% of the size of all disks to complete this process. A value of 0 means do not check for space before removing the snapshot. Default: 100

Table: webservices

Parameters	Description
contactwith	Set to ipaddress to contact miqhost using the IP address. Set to hostname to contact miqhost by its hostname. Set to resolved_ipaddress to take the hostname and resolve it to an IP address. Default: ipaddress
mode	Set to invoke to use webservices. Set to disable to turn off webservices. This is the same as Mode in Configuration-Operations- Server-Web Services in the appliance console. Default: invoke
nameresolution	If set to true, the hostname will be resolved to an IP address and saved with the host information in the VMDB. Default: false
security	If Web Services are enabled, you can set this to ws-security. This is the same as Security in Configuration-Operations- Server-Web Services in the appliance console. Note: This is not currently supported. Default: none
timeout	Specify the web service timeout in seconds. Default: 120
password	Specify the encrypted password for the user_name account. This is the same as Password in Configuration-Operations-Server-Outgoing SMTP E-mail Server. Default: blank
use_vim_broker	Controls if the vim_broker is used to communicate with VMware infrastructure. Default: true

Table: workers

Parameters	Description
worker_base	
defaults	If the following parameters are NOT explicitly defined for a specific worker, then these values will be used.
count	Number of this type of worker. Default: 1
gc_interval	How often to do garbage collection for this worker. Default: 15.minutes
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 3.seconds
poll_method	If set to normal, the worker checks for work the number of seconds set in the poll parameter. If set to escalate, the worker will increase the time between checks when there is no work to be done. Default: normal
poll_escalate_max	The maximum number of time to wait between checks for work. Poll_method must be set to escalate for this option to be used. Default: 30.seconds
heartbeat_freq	How often to “heartbeat” the worker. Default: 60.seconds
heartbeat_method	Set which way to dispatch work. Possible values are sql or drb. Default: drb
heartbeat_timeout	How long to wait until the worker heartbeat is considered timed out. Default: 2.minutes
parent_time_threshold	How long to allow the parent to go without heartbeating before considering the “parent’ not responding. For workers, the worker monitor is the parent. For Worker monitor, the Server is the parent. Default: 3.minutes
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 150.megabytes
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 10
restart_interval	How long to let a worker remain up before asking it to restart. All queue based workers are set to 2.hours and every other worker does not get restarted by a 0.hours value. Default: 0.hours
starting_timeout	How long to wait before checking a worker’s heartbeat when it is starting up to mark it as not responding, similar to a grace period before you begin monitoring it. Default: 10.minutes
event_catcher	Associated with Event Monitor Server Role. Captures ems events and queues them up for the event_handler to process. Parameters specified here will override those set in the worker_base:default section.
ems_event_page_size	Internal system setting which sets the maximum page size for the event collector history. This should not be modified. Default: 100
ems_event_thread_shutdown_timeout	Internal system setting which determines how long the event catcher at shutdown will wait for the event monitor thread to stop. This should not be modified. Default: 10.seconds
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 2.gigabytes
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 1
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 1.seconds
event_catcher_redhat	Contains settings that supersede the event_catcher for event_catcher_redhat.
event_catcher_vmware	Contains settings that supersede the event_catcher for event_catcher_vmware.
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 1.seconds
event_catcher_openstack	Contains settings that supersede the event_catcher for event_catcher_openstack.
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 15.seconds
topics	List of AMQP topics that should be monitored by Infrastructure Automation when gathering events from OpenStack.

Parameters	Description
duration	Qpid Specific. Length of time (in seconds) the receiver should wait for a message from the Qpid broker before timing out. Default: 10.seconds
capacity	Qpid Specific. The total number of messages that can be held locally by the Qpid client before it needs to fetch more messages from the broker. Default: 50.seconds
amqp_port	Port used for AMQP. Default: 5672
schedule_worker	Settings for Scheduler Server Role and any other work that runs on a schedule. Parameters specified here will override those set in the worker_base:default section.
db_diagnostics_interval	How frequently to collect database diagnostics statistics. Default: 30.minutes
job_proxy_dispatcher_interval	How often to check for available SmartProxies for SmartState Analysis jobs. Default: 15.seconds
job_proxy_dispatcher_stale_message_check_interval	How often to check for the dispatch message in the queue Default: 60.seconds
job_proxy_dispatcher_stale_message_timeout	Kill a message if this value is reached. Default: 2.minutes
job_timeout_interval	How often to check to see if a job has timed out. Default: 60.seconds
license_check_interval	How often to check for valid license. Default: 1.days
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 150.megabytes
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 3
performance_collection_interval	Controls how often the schedule worker will put performance collection request on the queue to be picked up by the collection worker. Default: 3.minutes
performance_collection_start_delay	How long after Infrastructure Automation Server has started before starting capacity and utilization collection, if collection needs to be done. Default: 5.minutes
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 15.seconds
server_logs_stats_interval	How often to log the Infrastructure Automation Server statistics. Default: 5.minutes
server_stats_interval	How often to collect the Infrastructure Automation Server statistics. Default: 60.seconds
session_timeout_interval	How often to check to see if a UI (appliance console) session has timed out. Default: 30.seconds
storage_file_collection_interval	How often to perform file inventory of storage locations. Default: 1.days
storage_file_collection_time_utc	What time to perform file inventory of storage locations. Default: “06:00”
vdi_refresh_interval	How often to refresh vdi inventory. Default: 20.minutes
vm_retired_interval	How often to check for virtual machines that should be retired. Default: 10.minutes
vm_scan_interval	How often to check virtual machines to see if scan needs to be done. Default: 10.minutes
smis_refresh_worker	Settings for Storage Inventory Server Role and any other work that runs on a schedule. Parameters specified here will override those set in the worker_base:default section
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 15.seconds
connection_pool_size	Maximum number of database connections allowed per process. Default: 5
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 1.gigabytes
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 3
smis_update_period	How frequently to update smis information. Default: 1.hours
status_update_period	How frequently to update smis status. Default: 5.minutes
stats_update_period	How frequently to update smis statistics. Default: 10.minutes
vim_broker_worker	Launched for any of these roles: Capacity & Utilization Collector, SmartProxy, SmartState Analysis, Management System Operations, Management System Inventory. Also launched if the use_vim_broker setting is on. Provides connection pooling, caching of data to and from the VMware infrastructure. Parameters specified here will override those set in the workers:default section.
heartbeat_freq	How often to heartbeat the worker. Default: 15.seconds
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 1.gigabytes
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 3
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 1.seconds

Parameters	Description
reconnect_retry_interval	Period after which connection is retried. Default: 5.minutes
vim_broker_status_interval	Internal system setting which configures how much time to wait after receiving event updates before checking for more updates. Default: 0.seconds
wait_for_started_timeout	Time between the worker's preload and startup time before considering the worker timed out. Default: 10.minutes
ui_worker:	Settings for User Interface Server Role. Parameters specified here will override those set in the worker_base:default section.
connection_pool_size	Maximum number of database connections allowed per process. Default: 5
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 1.gigabytes
nice_delta: 1	Tells the worker monitor what Unix "nice" value to assign the workers when starting. A lower number is less nice to other processes. Default: 1
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 60.seconds
web_service_worker	Settings for Web Services Server Role. Parameters specified here will override those set in the worker_base:default section.
connection_pool_size	Maximum number of database connections allowed per process. Default: 5
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 1.gigabytes
nice_delta	Tells the worker monitor what Unix "nice" value to assign the workers when starting. A lower number is less nice to other processes. Default: 1
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 60.seconds
queue_worker_base	Base class of all queue workers that work off of the queue..
defaults	If the following parameters are NOT explicitly defined for a queue worker, then these values will be used.
cpu_usage_threshold	How much cpu to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 100.percent
queue_timeout	How long a queue message can be worked on before it is considered timed out. Default: 10.minutes
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 400.megabytes
restart_interval	Queue workers restart interval. Default: 2.hours
poll_method	If set to normal, the worker checks for work the number of seconds set in the poll parameter. If set to escalate, the worker will increase the time between checks when there is no work to be done. Default: normal
generic_worker	Performs work that is not classified as any specific type of work. Processes all normal priority or non-specific queue items. Parameters specified here will override those set in the queue_worker_base:default section
count	Number of this type of worker. Default: 4
ems_refresh_worker	Performs all ems (management system) refreshes to keep the vmdb in sync with the state of the components of the virtual infrastructure in the various management systems. Parameters specified here will override those set in the queue_worker_base:default section
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 10.seconds
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 2.gigabytes
nice_delta	Tells the worker monitor what Unix "nice" value to assign the workers when starting. A lower number is less nice to other processes. Default: 7
restart_interval	Queue workers restart interval. Default: 2.hours
queue_timeout	How long a message can be worked on before it is considered timed out. Default: 120.minutes
event_handler	Associated with Event Monitor Server Role. Handles all events caught by the event catcher worker. Parameters specified here will override those set in the workers:default section. Parameters specified here will override those set in the queue_worker_base:default section
cpu_usage_threshold	How much cpu to allow the worker to grow to before gracefully requesting it to exit and restart. The value of 0 means that this worker will never be killed due to CPU usage. Default: 0.percent
nice_delta	Tells the worker monitor what Unix "nice" value to assign the workers when starting. A lower number is less nice to other processes. Default: 7
perf_collector_worker	Connects to VC/ESX to collect the raw performance data. Same as the Capacity & Utilization Data Collector Server Role. Parameters specified here will override those set in the queue_worker_base:default section count. Number of this type of worker. Default: 2
count	Number of this type of worker. Default: 2
poll_method	If set to normal, the worker checks for work the number of seconds set in the poll parameter. If set to escalate, the worker will increase the time between checks when there is no work to be done. Default: escalate
nice_delta	Tells the worker monitor what Unix "nice" value to assign the workers when starting. A lower number is less nice to other processes. Default: 3
perf_processor_worker	Processes the raw performance data into a reportable format. Same as the Capacity & Utilization Data Processor Server Role. Parameters specified here will override those set in the queue_worker_base:default section
count	Number of this type of worker. Default: 2
poll_method	If set to normal, the worker checks for work the number of seconds set in the poll parameter. If set to escalate, the worker will increase the time between checks when there is no work to be done. Default: escalate
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 400.megabytes
nice_delta	Tells the worker monitor what Unix "nice" value to assign the workers when starting. A lower number is less nice to other processes. Default: 7
priority_worker	Performs all high priority queue items including many tasks on behalf of the UI. UI requests are normally executed by a priority worker so they will not to block the UI. Parameters specified here will override those set in the queue_worker_base:default section
count	Number of this type of worker. Default: 2
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 200.megabytes

Parameters	Description
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 1
poll	How often the workers checks for work. This value only is only used when the worker has no more work to do from the queue. It will wait for an amount of time determined by the poll value and poll method. Therefore, if there is constant work on the queue, the worker will not wait in between messages. Default: 1.seconds
reporting_worker	Compiles reports. Settings for Reporting Server Role. Parameters specified here will override those set in the queue_worker_base:default section
count	Number of this type of worker. Default: 2
nice_delta	Tells the worker monitor what Unix “nice” value to assign the workers when starting. A lower number is less nice to other processes. Default: 7
smart_proxy_worker	Performs the embedded scanning of virtual machines. Settings for SmartProxy Server Role. Parameters specified here will override those set in the queue_worker_base:default section
count	Number of this type of worker. Default: 3
memory_threshold	How much memory to allow the worker to grow to before gracefully requesting it to exit and restart. Default: 600.megabytes
queue_timeout	How long a queue message can be worked on before it is considered timed out. Default: 20.minutes
restart_interval	Queue workers restart interval. Default: 2.hours

Schedules

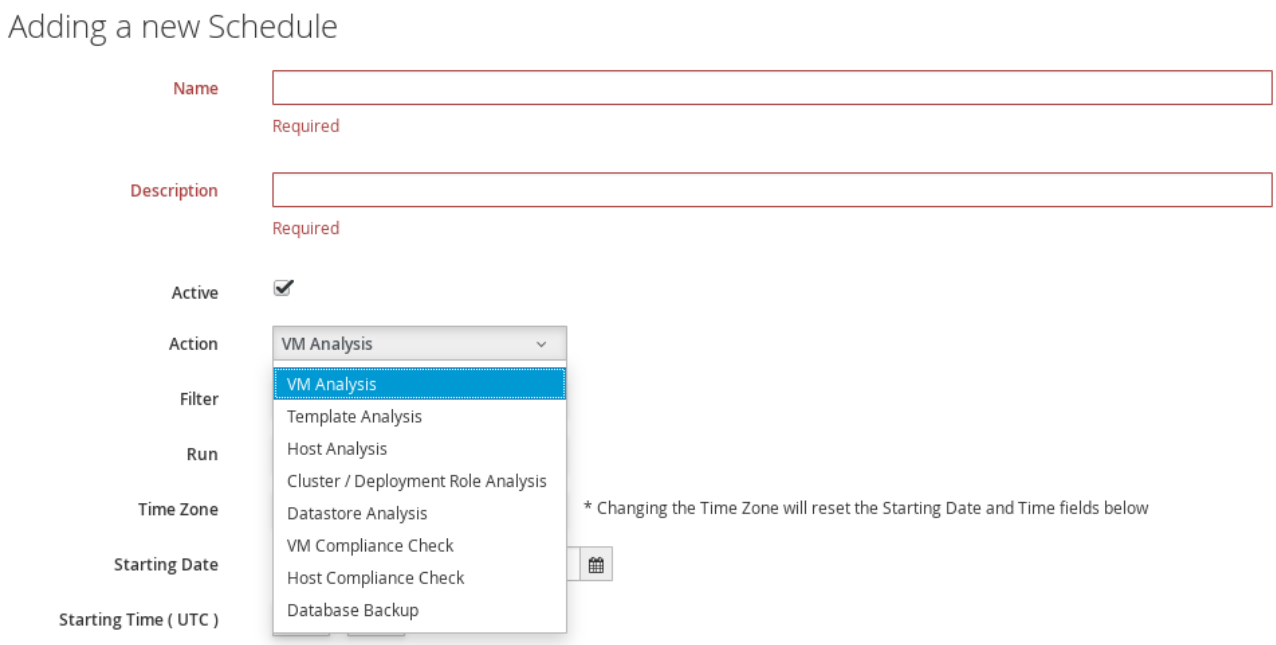
Scheduling SmartState Analyses

From the **Schedules** area in **Settings**, you can schedule the analyses of virtual machines, hosts, clusters, and datastores to keep the information current. Depending on which resource you want to analyze, you can filter which ones to analyze. You may also specify only one virtual machine or perform an analysis on all virtual machines. In addition, you can schedule compliance checks, and database backups.

Scheduling a SmartState Analysis or Compliance Check

To schedule a SmartState Analysis or Compliance Check:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Schedules**.
3. Click (**Configuration**), and (**Add a new Schedule**).
4. In the **Basic Information** area, type in a **Name** and **Description** for the schedule.
5. Select **Active** to enable this scan.
6. From the **Action** list, select the type of analysis to schedule. Based on the type of analysis you choose, you are presented with one of the following group boxes:



- **VM Analysis**: Displays **VM Selection** where you can choose to analyze **All VMs**, **All VMs for Provider**, **All VMs for Cluster**, **All VMs for Host**, **A single VM**, or **Global Filters**.
- **Template Analysis**: Displays **Template Selection** where you can choose to analyze **All Templates**, **All Templates for Provider**, **All Templates for Cluster**, **All Templates for Host**, **A single Template**, or **Global Filters**.
- **Host Analysis**: Displays **Host Selection** where you can choose to analyze **All Hosts**, **All Hosts for Provider**, **A single Host**, or **Global Filters**.

Note:

You can only schedule host analyses for connected virtual machines, not repository virtual machines that were discovered through that host. Since repository virtual machines do not retain a relationship with the host that discovered them, there is no current way to scan them through the scheduling feature. The host is shown because it may have connected virtual machines in the future when the schedule is set to run.

- **Cluster / Deployment Role Analysis**: Displays **Cluster Selection** where you can choose to analyze **All Clusters**, **All Clusters for Provider**, or **A single Cluster**.
- **Datastore Analysis**: Displays **Datastore Selection** where you can choose to analyze **All Datastores**, **All Datastores for Host**, **All Datastores for Provider**, **A single Datastore**, or **Global Filters**.
- **VM Compliance Check**: Displays **VM Selection** where you can choose to analyze **All VMs**, **All VMs for Provider**, **All VMs for Cluster**, **All VMs for Host**, **A single VM**, or **Global Filters**.
- **Host Compliance Check**: Displays **Host Selection** where you can choose to analyze **All Hosts**, **All Hosts for Provider**, **All Hosts for Cluster**, **A single Host**, or **Global Filters**.
- **Database Backup**: Under **Type**, displays **Network File System**, **Samba**, **Amazon AWS S3** and **OpenStack Swift**..

7. By applying **Global Filters** within any of the above items, you can designate which virtual machines or hosts to analyze.
8. In **Run**, set the frequency of the analysis to run. There are further options based on which **Run** option you choose.
   - Click **Once** to have the analysis run only one time.
   - Click **Daily** to run the analysis on a daily basis. You will be prompted to select the number of days between each analysis.
   - Click **Hourly** to run the analysis hourly. You will be prompted to select the number of hours between each analysis.
9. Select a **Time Zone**.

   Note:

   If you change the **Time Zone**, you will need to reset the stating date and time.
10. Type or select a date to begin the schedule in **Starting Date**.
11. Select a **Starting Time** based on a 24 hour clock in the selected Time Zone.
12. Click **Add**.

Modifying a Schedule

To modify a schedule:

1. Browse to **Settings > Application Settings**.
2. Click on the **Settings** accordion, then click **Schedules**.
3. Click the schedule that you want to edit.
4. Click (**Configuration**), and then click (**Edit this Schedule**).
5. Make the required changes.
6. Click **Save**.

Importing and Exporting Schedules

Infrastructure Automation provides the ability to import and export schedules using the **`import_export_schedules.rb`** script:

```
./var/www/miq/vmdb/tools/import_export_schedules.rb -h
```

Option	Description
-u, –user=<s>	userid (default: admin)
-d, –output-dir=<s>	Output directory (default: ./)
-s, –schedule=<s>	Schedule name or id
-o, –operation=<s>	Export or import (default: export)
-y, –import-yaml=<filename/uri>	Imported yaml
-h, –help	Show this message

Usage Example.

Import:

```
./var/www/miq/vmdb/tools/import_export_schedules.rb -o import -y ./filename.yaml
```

Export:

```
./var/www/miq/vmdb/tools/import_export_schedules.rb -s 158
```

Note that the **`import_export_schedules.rb`** script works for all types of schedules, including:

- Report
- Policy
- Alert
- SmartState Analysis
- Automation Task
- Service Template

Access Control

Browse to **Settings > Application Settings**. Click on the **Access Control** accordion to see a hierarchy of the configurable items for users, groups, roles, and tenants. You can add and modify users, groups, account roles, tenants, and projects.

Creating a Tenant

Tenants are used in Infrastructure Automation for controlling visibility of objects.

To create a tenant:

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Tenants**.
3. Click on the top-level **Tenant**, click (**Configuration**), and (**Add child Tenant to this Tenant**) to create a tenant.
4. Enter a name for the tenant in the **Name** field.
5. Enter a description for the tenant in the **Description** field.
6. Click **Add**.

Creating a Project

A tenant can also contain a self-contained child tenant known as a 'project'. A project cannot have a child tenant, but it is useful for allocating resources to a small group or team within a larger organization.

To create a project:

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Tenants**.
3. Click on the **Tenant** where you want to add a **Project**, click (**Configuration**), and (**Add Project to this Tenant**) to create a project.
4. Enter a name for the project in the **Name** field.
5. Enter a description for the project in the **Description** field.
6. Click **Add**.

Managing Tenant and Project Quotas

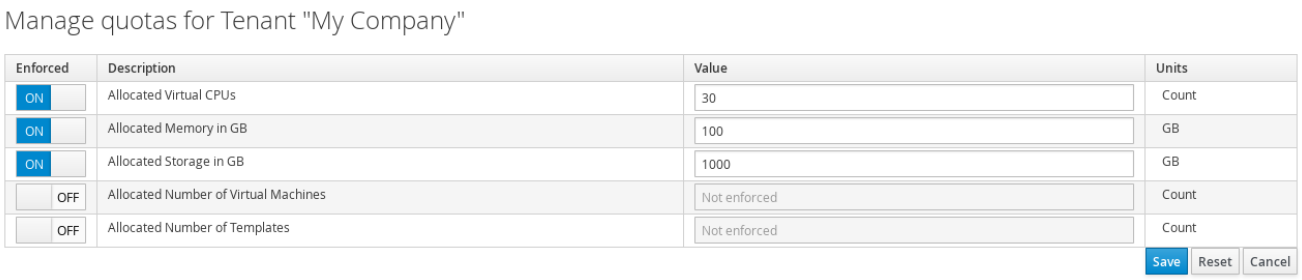
Use the following procedure to allocate or edit quotas for tenants and projects.

Note:

Quota is allocated based on the user's current group. If the user belongs to multiple groups, you must change to the desired group before allocating or editing group quota. See Groups for more details.

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Tenants**.
3. Click on the **Tenant** or **Project** where you want to add a quota, click (**Configuration**), and (**Manage quotas for the Selected Item**) to create a quota.
4. In the list of pre-built quotas, switch **Enforced** next to the quota item you want to enable to `Yes`.

5. In the **Value** field, enter the constraints you want to apply to the quota.



6. Click **Save**.

Tagging Tenants and Projects

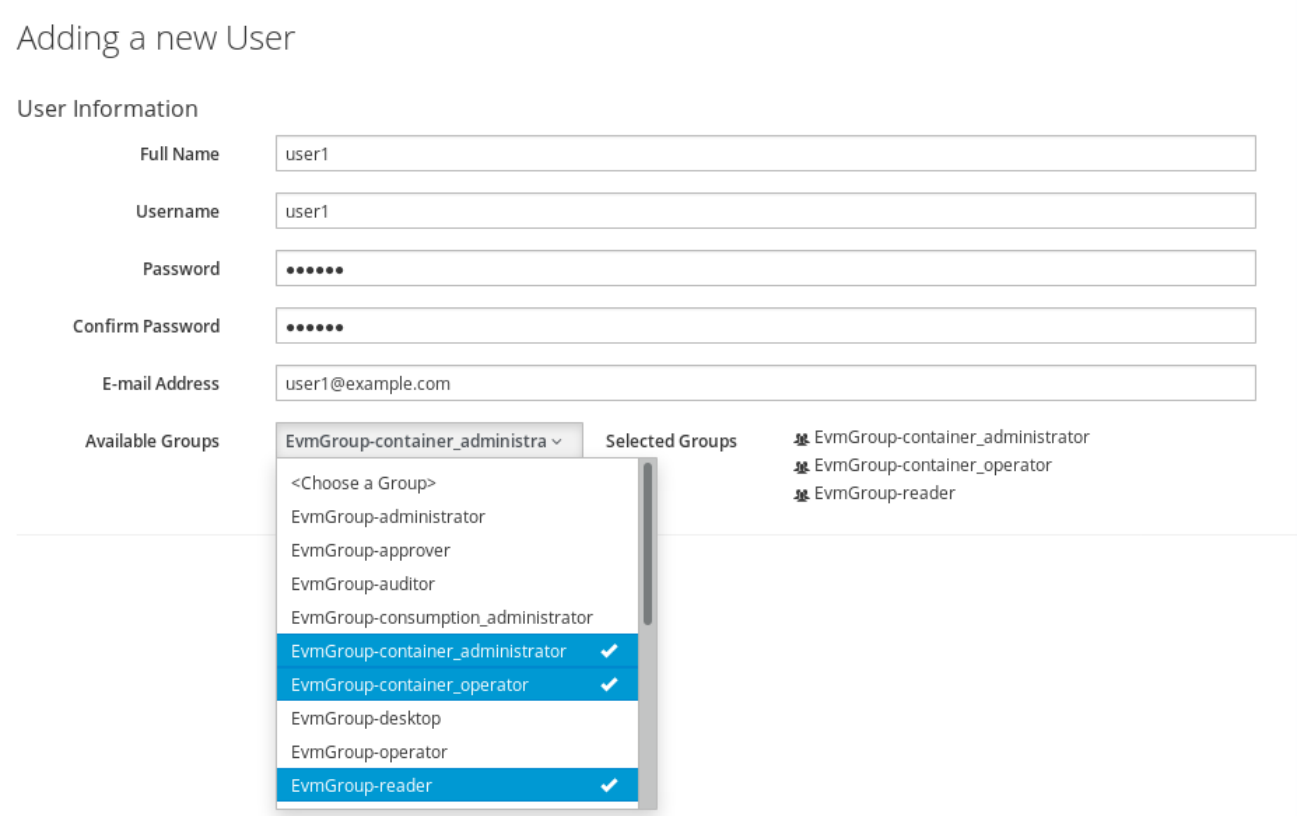
To tag tenants and projects:

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Tenants**.
3. Select the tenant or project, then click **Policy**, and select **Edit My Company Tags for this Tenant**.
4. In **Tag Assignment**, click **Select a customer tag to assign**, and select a tag from the list. In the next column, set a corresponding value.
5. Click **Save**.

Creating a User

To create a user:

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Users**.
3. Click (**Configuration**), and (**Add a new User**) to create a user.
4. Enter a **Full Name**, **Username**, **Password** with confirmation, and **Email Address** for the user. An email address is required; omitting the user email address can result in unsuccessful provisioning requests.



- If you are using LDAP, but did not enable **Get User Groups from LDAP** in your server's **Authentication** tab, you will need to define a user. The UserID must match exactly the user's name as defined in your directory service. Use all lowercase characters to ensure the user can be found in the VMDB. - When the user logs in, they use their LDAP password.

5. Select one or more groups from **Available Groups**.
6. Click **Add**.

Deleting a User

For security reasons, delete any user that no longer needs access to the information or functions of the server.

To delete a user:

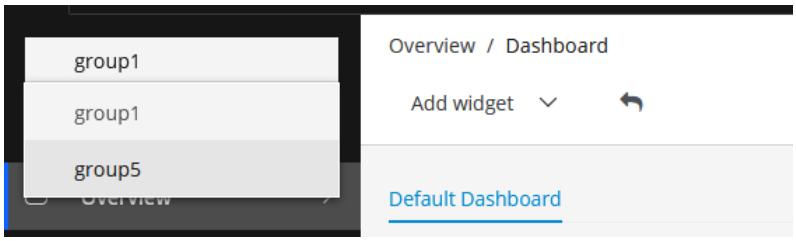
1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Users**.
3. Select the user you want to delete.
4. Click (**Configuration**), and (**Delete selected Users**) to delete a user.

Groups

User groups create filters and assign roles to users. You can either create your own groups, or leverage your LDAP directory service to assign groups of users to account roles. For a list of what each pre-defined account role can do, see Roles.

A user can exist in multiple groups. However, a group can only be assigned one account role.

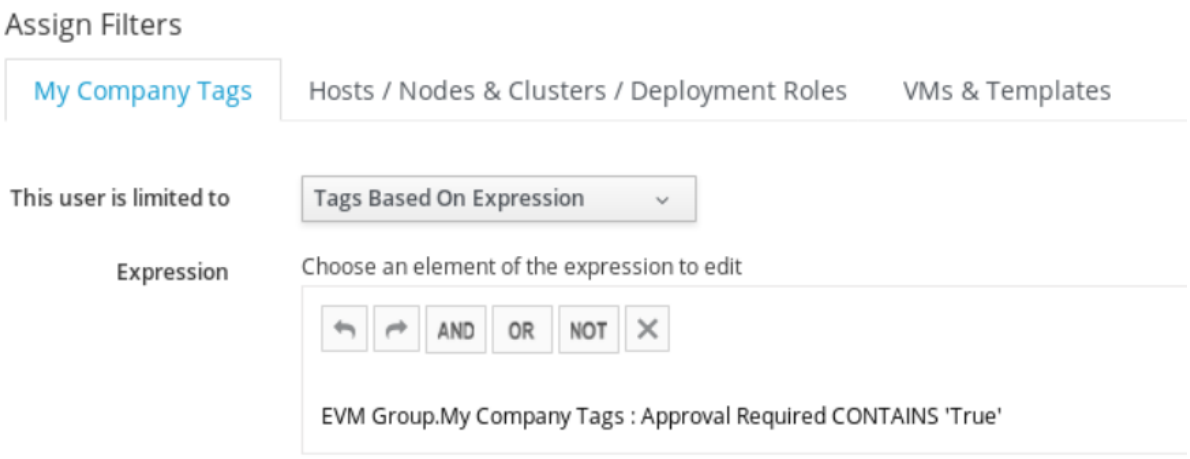
If you belong to multiple groups, you can change the current group. The Current Group affects quota and other settings for the user, including permissions for controlling the tasks that you can complete and the data that you can view. To change the group, click the group name in the upper left of the screen, and select your preferred group from the listed groups.



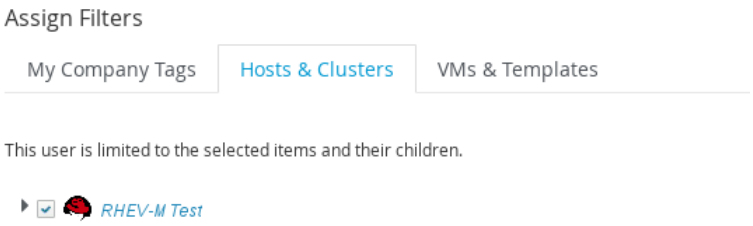
Creating a Group

To create a user group:

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Groups**.
3. Click (**Configuration**), and (**Add a new Group**) to create a group.
4. Enter a name for the group in the **Name** field. To ensure compatibility with tags, use underscores in place of spaces. For example, Infrastructure Automation-`test_group`.
5. Select a **Role** to map to this group. For a description of each Infrastructure Automation role, see Account Roles and Descriptions.
6. Select the **Project/Tenant** this group must belong to.
7. Limit what users in this group can view by selecting filters in the **Assign Filters** area.
    1. Click the **<My Company> Tags** tab to select the tags that users in this group can access. Resources with the selected tags attached can be accessed by this group. Select tags using one of the options in the **This user is limited to** list:
        - Select **Specific Tags**, then check the boxes for the tags that you want to limit this user to. The items that have changed will show in blue italicized font.
        - Select **Tags Based On Expression**, then create tags based on an expression using AND, OR, or NOT. This allows you to further limit the resources accessible to a user: for example, to specify a combination of tags that must exist on a resource.



2. Click the **Host & Clusters** tab.
    1. Check the boxes for the host and clusters that you want to limit this user to. The items that have changed will show in blue italicized font.

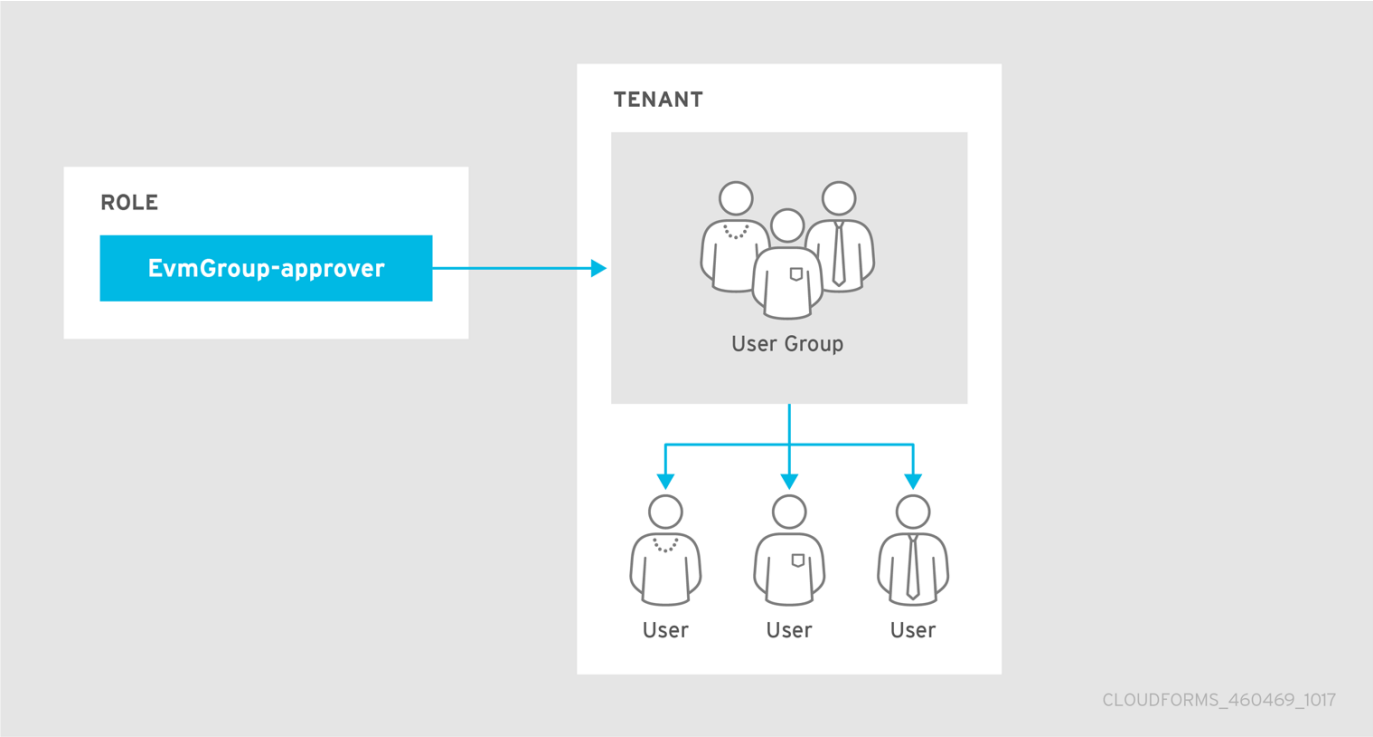


3. Click the **VMs & Templates** tab. This shows folders that you have created in your virtual infrastructure.
    1. Check the boxes for the folders that you want to limit this user to. The items that have changed will show in blue italicized font.
8. Click **Add**.

After creating a group, assign one or more users to the group by editing a user.

Roles

When you create a group, you must specify a role to give the group rights to resources in the console. The group's role determines the scope of access for the users that are members of the group.



Infrastructure Automation provides a default group of roles, but you can also create your own, or copy and edit the default groups.

Note:

If you have enabled **Get Role from LDAP** under **LDAP Settings**, then the role is determined by the LDAP user's group membership in the directory service.

To view details of a role and its level of access:

1. Browse to **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Roles**.
3. Click on a role from the list to display role information and the product features the role can access (marked by a checkmark). You can expand the categories under **Product Features** to see further detail.

The table below shows a summary of the functions available to each role.

Account Roles and Descriptions

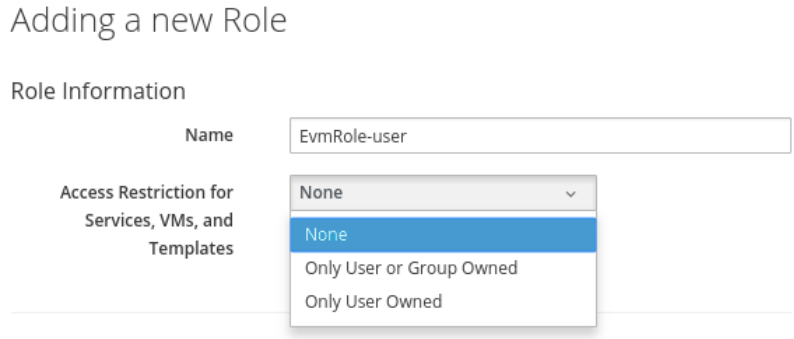
Role	Description
Administrator	Administrator of the virtual infrastructure. Can access all infrastructure functionality. Cannot change server configuration.
Approver	Approver of processes, but not operations. Can view items in the virtual infrastructure, view all aspects of policies and assign policies to policy profiles. Cannot perform actions on infrastructure items.
Auditor	Able to see virtual infrastructure for auditing purposes. Can view all infrastructure items. Cannot perform actions on them.
Container Administrator	Administrator with capabilities to configure, view and execute tasks on all containers and related underlying infrastructure. Has access to Nodes, Pods and Projects dashboards.
Container Operator	This role can view and execute tasks related to containers and related underlying infrastructure. The Container Operator has access to locked versions of the same dashboards as the Container Administrator.
Desktop	Access to VDI pages.
Operator	Performs operations of virtual infrastructure. Can view and perform all functions on virtual infrastructure items including starting and stopping virtual machines. Cannot assign policy, but can view policy simulation from Virtual Machine page.
Security	Enforces security for the virtual environment. Can assign policies to policy profiles, control user accounts, and view all parts of virtual infrastructure. Cannot create policies or perform actions on virtual infrastructure.
Super Administrator	Administrator of Infrastructure Automation and the virtual infrastructure. Can access all functionality and configuration areas.
Support	Access to features required by a support department such as diagnostics (logs). Can view all infrastructure items and logs. Cannot perform actions on them.
Tenant Administrator	Configures settings applicable to a Tenant. Sets Branding, maps groups/roles, configures LDAP credentials, and configures dashboard settings.
Tenant Quota Administrator	Configures quota limits for the tenant, applying usage constraints for CPU, Memory, Storage, Maximum number of VMs, and Maximum number of Templates.
User	User of the virtual infrastructure. Can view all virtual infrastructure items. Cannot perform actions on them.
User Limited Self Service	Limited User of virtual machines. Can make provision requests. Can access some functions on the virtual machine that the user owns including changing power state.

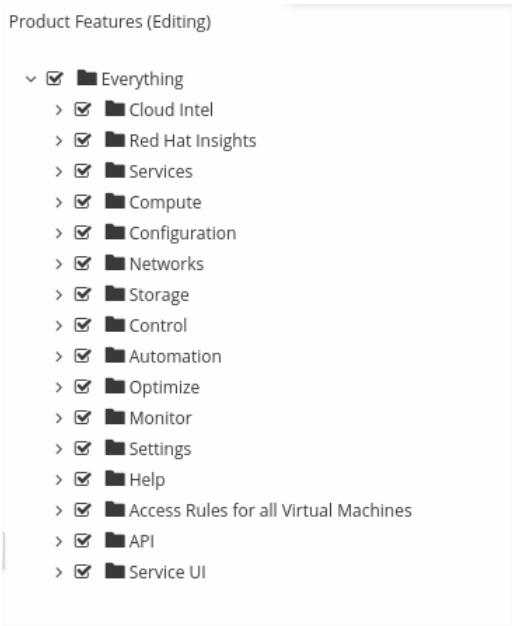
Role	Description
User Self Service	User of virtual machines. Can make provision requests. Can access some functions on the virtual machine that the user owns and that the user's LDAP groups own including changing power state.
Vm User	User of virtual machines. Can access all functions on the virtual machine including changing power state and viewing its console. Cannot assign policy, but can view policy simulation from virtual machine page.

Creating a Role

To create a role:

1. Click **Settings > Application Settings**.
2. Click on the **Access Control** accordion, then click **Roles**.
3. Click **Configuration**, and **Add a new Role**. Alternatively, you can copy an existing role to a new role by clicking **Copy the selected Role to a new Role**.
4. In the **Role Information** area, type a name for the new role. For **Access Restriction for Catalog Items, Orchestration Stacks, Key Pairs, Services, VMs, and Templates**, select if you want to limit users with this role to only see resources owned by the user or their group, owned by the user, or all resources (**None**):





5. Under **Product Features (Editing)**, navigate to the appropriate feature and enable or disable it:
6. Click **Add**.

Diagnostics

Browse to **Settings > Application Settings**. Click on the **Diagnostics** tab to see the status of the different Infrastructure Automation roles and workers for each server, view and collect logs, and gather data if there are any gaps in capacity and utilization information. The Diagnostics area is designed in a hierarchy.

- At the **region** level, you can see replication status, backup the VMDB, and run garbage collection on the VMDB.
- At the **zone** level, you can see Infrastructure Automation roles by servers and servers by roles. In addition, you can set log collection values for a specific zone, and collect gap data for capacity and utilization.
- At the **server** level, you can see the workers for each server, set log collection values for a specific server, and view current logs.

Region Diagnostics

Using the console, you can set the priority of server regional roles, review and reset replication, and create backups of your database either on demand or on a schedule.

Regions are used primarily to consolidate multiple VMDBs into one master VMDB for reporting while zones are used to define functional groups of servers. There can be only one region per VMDB, but multiple zones per region (or VMDB). Some server roles are aware of each other across Infrastructure Automation appliances at the region level. This means that redundancy and failover rules apply at the region level. You can also set priorities for the server roles that provide failover.

Server Role Priorities

If you have multiple servers in your environment with duplicate failover roles, then you can set the priority of the server role.

- Only server roles that support failover can be marked as primary. These roles only allow one server to be active at a time. These are **Notifier**, **Capacity & Utilization Coordinator**, **Event Monitor**, **Scheduler**, **Storage Inventory**, and **Provider Inventory**.
- All other server roles are additive. The more servers with that role in a zone the more work that can be performed.

There are three role priorities.

- **Primary**: There can only be one primary per zone or region per role. When an appliance is started, the system looks to see if any role is set to primary. If that is the case, the role is activated on that appliance and deactivated from the secondary. In the console, primary roles are shown in bold letters. The text turns red if the server goes down. You must actively set the primary priority.
- **Secondary**: This is the default priority. There can be multiple secondaries. When an appliance is started, if no primary is found in the zone, the first appliance to start takes the role. In the console, secondary roles are displayed normally with the word “secondary”.
- **Tertiary**: If all appliances with primary roles or secondary roles were down, one of the tertiary would be activated. The reason for tertiary is to ensure that if a server with crucial roles such as Provider Inventory or Event Monitor goes down, you have a way to associate those roles to different appliances by organizing the priorities. Tertiary roles simply show as active in the console.

Region Aware Server Roles

Role	More than one per Region	Can have Priority Set
Automation Engine	Y	N
Database Operations	Y	N
Notifier	N	Y
Reporting	Y	N
Scheduler	N	Y
User Interface	Y	N
Web Services	Y	N

Setting the Priority of a Failover Role

To set the priority of a failover role:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Depending on how you want to view your servers, click either the **Roles by Servers** tab or the **Servers by Roles** tab.
4. In the **Status** of **Roles for Servers** in **Zone Default Zone** area, click on the role that you want to set the priority for.
5. Click  (**Configuration**), and  (**Promote Server**) to make this the primary server for this role.
6. Click  (**Configuration**), and  (**Demote Server**) to demote the priority of this server for this role.

Zone Diagnostics

The console provides a way to see all the server roles that a server has been assigned and if these roles are running. This is especially helpful when you have multiple servers with different server roles. For each zone you can also set a central place for all logs to be collected, and collect capacity and utilization data that may be missing.

Viewing the Status of Server Roles

To view the status of server roles:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Depending on how you want to view your servers, click either **Roles by Servers** or the **Servers by Roles**.

Zone Aware Server Roles

Role	More than one per Region	Can have Priority Set
Automation Engine	Y	N
Capacity & Utilization Coordinator	N	Y
Capacity & Utilization Data Collector	Y	N
Capacity & Utilization Data Processor	Y	N
Database Operations	Y	N

Role	More than one per Region	Can have Priority Set
Event Monitor	N	Y
Provider Inventory	N	Y
Provider Operations	Y	N
Notifier	N	Y
Reporting	Y	N
Scheduler	N	Y
SmartProxy	Y	N
SmartState Analysis	Y	N
User Interface	Y	N
Web Services	Y	N

Removing an Inactive Server

To remove an inactive server:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Click on the name of the server in the tree view.
4. Click (**Delete Server**). This button is available only if the server is inactive.

Zone Log Collections

If you have multiple servers reporting to one central VMDB, then you can collect the configuration files and logs from the console of any of the servers. While you can set this either at the zone or server level, settings at the server level supersede the ones at the zone level.

Log depot options include:

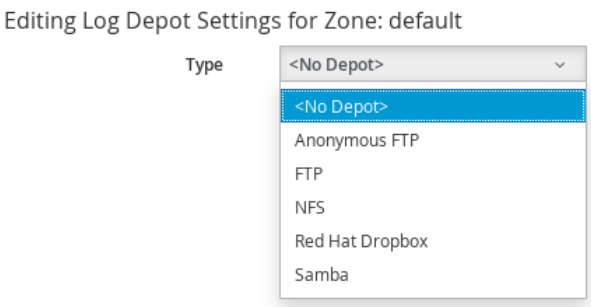
- Anonymous File Transfer Protocol (FTP)
- File Transfer Protocol (FTP)
- Network File System (NFS)
- Red Hat Dropbox
- Samba

See your network administrator if you need to set up one of these shares. You will also need a user that has write access to that location.

Setting the Location of the Log Depot

To set the location of the log depot:

1. Browse to **Settings > Application Settings**.
2. Click the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Click **Collect Logs**.
4. Click (**Edit**).
5. Select the **Type** of share.
6. Using the fully qualified domain name (**FQDN**) of the depot server, type in the appropriate settings for the **URI**.



7. If required, enter your user **ID** and **password** then click **Validate** to confirm the settings.
8. Click **Save**.

Collecting and Downloading Logs from All Servers in a Zone

To collect and download logs from all servers in a zone:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Click the **Collect Logs** tab.
4. Click (**Collect all logs**). All files in the logs directory as well as configuration files are collected from the selected zone.
5. Click **OK**. The status of the log retrieval shows in the Infrastructure Automation console.

Capacity and Utilization Repair

Under certain circumstances, it is possible that Infrastructure Automation is not able to collect capacity and utilization data. This could be due to password expiration, a change in rights to the cloud provider and this change didn't provide enough granularity to the Infrastructure Automation service account, or network connectivity. The gap data is collected directly by extracting the monthly performance data. Gap collection need to be completed for each zone individually. Therefore, the procedure below need to be repeated for each zone.

Repairing Capacity and Utilization Data

To repair capacity and utilization data:

1. Log in to a Infrastructure Automation appliance located in the zone for which you want to gather the data.
2. Browse to **Settings > Application Settings**.
3. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
4. Click **C & U Gap Collection**.
    1. Select the appropriate **Timezone**.

       Note:

       Do not select more than one week unless instructed to do so by Red Hat Support.

    2. Select a **Start Date**.
    3. Select an **End Date**.
5. Click **Submit**.

After the gap collection has completed for this zone, repeat these same steps for the next zone. You can check for completion by going to the clusters page and checking for the capacity and utilization data for the time period specified.

Server Diagnostics

Under **Diagnostics** for a server, you can view the status of Infrastructure Automation workers running on the server, set log collection setting for only that server, and view the server's current Infrastructure Automation and audit logs.

Workers

The **Workers** tab enables you to see the status of and restart Infrastructure Automation workers.

You can see additional information on and restart the following items:

- **C & U Metrics Collectors** that collects capacity and utilization data.
- **C & U Metrics Processors**, which processes the collected capacity and utilization data.
- **Event Handlers** put events from the Event Monitor into the VMDB and starts Infrastructure Automation processes if needed base on that information.
- **Event Monitors** that communicate with the external cloud provider to deliver up to date event information.
- **Generic Workers** that perform long running and priority processes.
- **Priority Workers** that perform high priority, short processes.
- **Schedule Workers** that maintains any items that run on a schedule.
- **Session Broker** that maintains a single connection to the cloud providers .
- **Refresh Workers** that runs the refresh processes.
- **Reporting Workers** that generate reports.
- **SmartProxy Workers** that run SmartState Analyses on virtual machine.
- **User Interface Worker** that allows users access to the console.
- **Web Services Worker** that maintains Infrastructure Automation Web services.
- **VM Analysis Collectors** that run and process SmartState Analyses on virtual machines.

Reloading Worker Display

To reload worker display:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click the **Workers** tab.
5. Click  (**Refresh Current Workers display**).

Restarting a Worker

To restart a worker:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click on the **Workers** tab.
5. Click on the worker you want to restart.
6. Click  (**Configuration**), then  (**Restart selected worker**).
7. Click **OK**.

Server and Audit Logs

Collecting Server Logs and Configuration Files

While you can designate a central location to collect logs for all servers in a specific zone, you can override those values for a specific server. To do this, designate a log depot location to store the files.

Log depot options include:

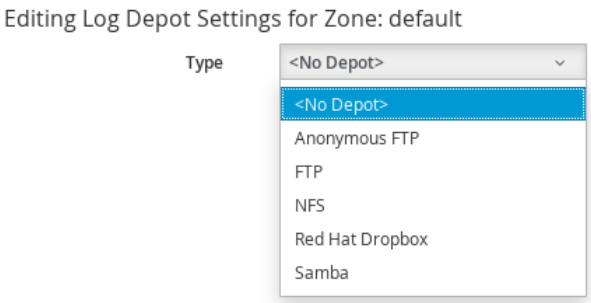
- Anonymous File Transfer Protocol (FTP)
- File Transfer Protocol (FTP)
- Network File System (NFS)
- Red Hat Dropbox
- Samba

See your network administrator to set up one of these shares. You also need a user that has write access to that location. Settings at the server level supersede the ones at the zone level.

Setting the Location of the Log Depot for a Specific Server

To set the location of the log depot for a specific server

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to collect logs for.
4. Click on the **Collect Logs** tab.
5. Click  (**Edit Log Depot Settings for the selected Server**).
6. Select the **Type** of share.



7. Using the fully qualified domain name (**FQDN**) of the depot server, type in the appropriate settings for the **URI**.
8. Enter your user ID and password, then click **Validate** to confirm the settings.
9. Click **Save**.

Collecting the Current Log Set of a Server

To Collect the Current Log Set of a Server

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to collect logs for.
4. Click on the **Collect Logs** tab.
5. Click (**Collect**), then click (**Collect current logs**). All current log files in as well as configuration files are collected.
6. Click **OK**. The status of the log retrieval shows in the Infrastructure Automation console.

Collecting All Log Sets from a Server

To Collect All Log Sets from a Server

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to collect logs for.
4. Click **Collect Logs**.
5. Click (**Collect**), then click (**Collect all logs**). All files in the logs directory as well as configuration files are collected.
6. Click **OK**. The status of the log retrieval shows in the Infrastructure Automation console.

Viewing the Server, Audit, and Production Logs

The server and audit logs roll over daily. The previous logs are stored as zipped files in the `/var/www/miq/vmdb/log` folder. The current logs can be easily viewed and downloaded from the settings menu; select **Configuration**, then click on the **Diagnostics** accordion.

Use the server log to see all actions taken by the server including communication with the SmartProxy and tasks.

Viewing the Server Log

To view the server log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click **CFME Log**.

The Infrastructure Automation server automatically retrieves the last 1000 lines of the log.

Reloading the Server Log

To reload the server log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.

3. Select the server that you want to view.
4. Click **CFME Log**.
5. Click (Reload the Log Display).

Downloading the Server Log

To download the server log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click **CFME Log**.
5. Click (**Download the Entire EVM Log File**).

Note:

Use the **Audit Log** to see changes to the user interface and authentication.

Viewing the Audit Log

To view the audit log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click **Audit Log**.

The server automatically retrieves the last 1000 lines of the log.

Reloading the Audit Log

To reload the audit log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click **Audit Log**.
5. Click (**Reload the Audit Log Display**).

Downloading the Audit Log

To download the audit log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the Zone that you want to view.
3. Select the server that you want to view.
4. Click Audit Log.
5. Click (**Download the Entire Audit Log File**).

Viewing the Production Log

Use the production log to see all actions performed using the console.

To view the production log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click **Production Log**.

The Infrastructure Automation server automatically retrieves the last 1000 lines of the log.

Reloading the Production Log

To reload the production log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the Zone that you want to view.
3. Select the server that you want to view.
4. Click **Production Log**.
5. Click  (**Reload the Product Log Display**).

Downloading the Production Log

To download the production log:

1. Browse to **Settings > Application Settings**.
2. Click on the **Diagnostics** accordion, then click the **Zone** that you want to view.
3. Select the server that you want to view.
4. Click **Production Log**.
5. Click  (**Download the Production Log File**).

Database Operations

Viewing Information on the VMDB

The **Database** accordion displays a summary of VMDB information, a list of all tables and indexes, settings for the tables, active client connection, and database utilization.

To view information on the VMDB:

1. Browse to **Settings > Application Settings**.
2. Click the **Database** accordion.
3. Click **VMDB** in the tree view on the left.
4. Click the appropriate tab to view the desired information:
   - **Summary**: displays statistics about the database.
   - **Tables**: displays a clickable list of all the tables.
   - **Indexes**: displays a clickable list of all the indexes.
   - **Settings**: displays a list of all tables, their descriptions, and other valuable Information.
   - **Client Connections**: displays all current connections to the VMDB.
   - **Utilization**: displays usage charges for the disk and index nodes.

Database Regions and Replication

Regions are used to create a central database for reporting and charting. Do not use the database at the top level for operational tasks such as SmartState analysis or capacity and utilization data collection. Assign each server participating in the region a unique number during the regionalization process, then set your subordinate regions to replicate to the top region.

Important

All Infrastructure Automation databases in a multi-region deployment must use the same security key.

Creating a Region

In principle, a region is created when you set up your Infrastructure Automation environment on the first appliance for the region. However, you can also create a region on an appliance where a database has already been set up. This process involves dropping and rebuilding the existing database to accommodate the new region number, and takes several minutes to complete.

Warning

Performing this procedure destroys any existing data and cannot be undone. Back up the existing database before proceeding. By default, new appliances are assigned region 0. Do not use this number when creating a region because duplicate region numbers can compromise the integrity of the data in the database.

1. Log in to the appliance as the **root** user.
2. Enter **`appliance_console`**, and press **Enter**.
3. Press any key.
4. Select **`Stop EVM Server Processes`**.

5. Enter **`Y`** to confirm.
6. Choose **`Configure Application`**.
7. Enter a database region number that has not been used in your environment. Do not enter duplicate region numbers because this can corrupt the data.
8. Enter **`Y`** to confirm.
9. The menu reappears after all processes are complete.
10. Select **`Start EVM Server Processes`**.
11. Enter **`Y`** to confirm.
12. Select **Quit** to exit the advanced settings menu.

Configuring Database Replication and Centralized Administration

To configure database replication, you must configure one Infrastructure Automation instance to act as a global copy, and one or more other instances to act as remote copies. This database replication relationship can only be configured on Infrastructure Automation instances that are of the same version. Configuring database replication in this version of Infrastructure Automation automatically enables centralized administration, eliminating the need for further configuration. Centralized administration in Infrastructure Automation supports life cycle management operations that can be initiated from the global copy and processed and executed on the remote copy.

Important

- You must configure at least one remote copy before you can configure the global copy. Changes to the role of a Infrastructure Automation instance take several minutes to take effect.
- The region number must be unique on each Infrastructure Automation instance where replication is configured. See Creating a Region for instructions on how to create a region.
- Enabling the **`EvmRole-user_global_region`** role will only provide access to specific operations that can be performed on a global region.

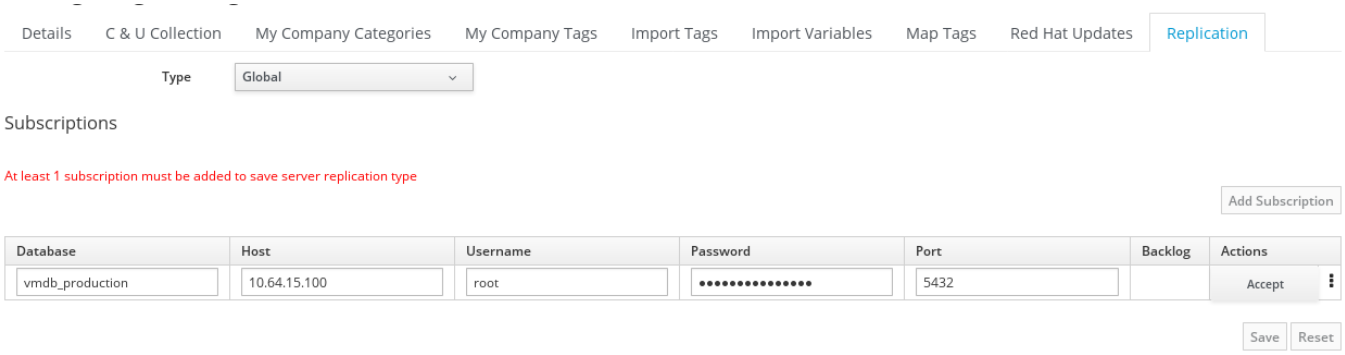
Configuring a Remote Copy

Configure a Infrastructure Automation instance to act as a remote copy from which data will be replicated to the global copy.

1. Browse to **Settings > Application Settings**.
2. Click the **Settings** accordion.
3. Click the region where the instance is located.
4. Click **Replication**.
5. Select **Remote** from the **Type** list.
6. Click **Save**.

Configuring the Global Copy

Configure a Infrastructure Automation instance to act as the global copy to which data is replicated from the remote copies.



1. Click **Configuration**.
2. Click the **Settings** accordion.
3. Click the region where the instance is located.
4. Click **Replication**.
5. Select **Global** from the **Type** list.
6. Click **Add Subscription**.
    1. Enter the name of the database on the remote copy in the **Database** field.

2. Enter the IP address or fully qualified domain name of the remote copy in the **Host** field.
3. Enter the name of a database user able to access the database in the **Username** field.
4. Enter the password of the database user in the **Password** field.
5. Enter the port by which the database is accessed in the **Port** field.
6. In **Actions**, click **Accept**. You can also **Update** your subscription if required, or click the menu on the right in order to **Validate** or **Discard**.

7. Click **Save**.

Note:

Once you configure a Infrastructure Automation instance to act as a global copy, and one or more other instances to act as remote copies: * centralized administration is automatically enabled after the initial data sync is complete. * you will see the type of region you are logging into (global or remote) on the Infrastructure Automation user interface login screen.

Database replication and centralized administration are now enabled on your instances. To open virtual machines of a remote region from the global region and perform required operations:

1. Browse to menu: **Compute < Infrastructure < Virtual Machines**, then click **VMs & Templates** and select a virtual machine in the global region.
2. In the **Multi Region** table on the summary screen, click **Connect to VM in its Region** to perform required operations in a new browser tab.

Resetting Database Replication

You can reset the replication relationship between the global copy and remote copies by temporarily removing and re-enabling the subscription from the global copy.

1. Browse to **Settings > Application Settings**.
2. Click the **Settings** accordion.
3. Click the region where the instance is located.
4. Click **Replication**.
5. Remove the subscription:
    1. Click the actions button for the subscription to reset.
    2. Click **OK**.
6. Click **Save**.
7. Re-add the subscription:
    1. Click **Add Subscription**.
    2. Enter the name of the database on the remote copy in the **Database** field.
    3. Enter the IP address or fully qualified domain name of the remote copy in the **Host** field.
    4. Enter the name of a database user able to access the database in the **Username** field.
    5. Enter the password of the database user in the **Password** field.
    6. Enter the port by which the database is accessed in the **Port** field.
8. Click **Save**.

Backing Up and Restoring Infrastructure Automation in a Kubernetes Based Environment (podified)

Prerequisites:

- Velero and Restic must be installed and working in your cluster. For more information on Velero, see the Velero documentation. For more information on Restic, see the Restic documentation

Note: The operator adds labels and/or annotations to everything that needs to be backed up. You can easily and quickly backup only what is required. The operator will add annotations for restic volume backups for example, `backup.velero.io/backup-volumes: <volume-label>`.

1. The backup label key can be configured by adding the following to the CR. The default label is applied if no value is specified for `backupLabelName`.

```
...
spec:
  ...
  backupLabelName: <your label name>
```

2. Create a backup by running the command:

```
velero backup create <your backup name> --include-namespaces <your namespace> -l <your label name>=t
```

3. In order to restore a backup, first ensure the Infrastructure Automation CRD exists. Then, restore the backup by running the commands:

```
oc new-project <your namespace>
velero restore create --from-backup <your backup name>
```

Backing Up and Restoring a Database for an Appliance Based Environment

Running a Single Database Backup

Infrastructure Automation supports database backups to **Network File System (NFS)**, **Samba**, **Amazon Web Service(AWS) S3** and **OpenStack Swift** storage.

Note:

Database backups to **AWS S3** require valid AWS credentials to perform this operation.

To run a single database backup:

1. Browse to **Settings > Application Settings**.
2. Click the **Diagnostics** accordion and click the **Region** name.
3. Click the **Database** tab.
4. If you have created a backup schedule and want to use the same depot settings, select the schedule in the **Backup Schedules** box.
5. If you do not want to use the settings from a backup schedule, select a type of server for storing the backups from the Type drop-down list in the **Database Backup Settings** box. You can use **Network File System** (NFS), **Samba** or **AWS S3**.
   - If you select **Samba**, enter the **Depot Name**, **URI**, **Username**, and a valid **Password**. Click **Validate** to check the settings.
   - If you select **AWS S3**, enter the **Depot Name**, **URI**, and select the **AWS Region** from the drop-down list. Enter the **Username**, and a valid **Password**. Click **Validate** to check the settings.
   - If you select **OpenStack Swift**, enter the **Depot Name**, **URI**, **Security Protocol** and **API Port**. Enter the **Username**, and a valid **Password**. Click **Validate** to check the settings.
   - If you select **Network File System**, enter the **URI**.
6. Click **Submit** to run the database backup.

Restoring a Database from a Backup

If a database is corrupt or fails, restore it from a backup. You can restore a backup from a local file, NFS, or Samba.

To restore a database from a backup:

1. Save the database backup file as **`/tmp/evm_db.backup`**. Infrastructure Automation looks specifically for this file when restoring a database from a local backup.
2. If you are restoring a database backup on a high availability environment, stop the replication manager service (**`repmgr13.service`**). This is not required in other Infrastructure Automation configurations.

   ```
   # systemctl stop repmgr13.service
   ```
3. Log in to the appliance as the **root** user.
4. Enter **`appliance_console`**, and press **Enter**.
5. Select **`Stop EVM Server Processes`** to stop processes on all servers that connect to this VMDB.
6. Enter **`Y`** to confirm.
7. After all processes are stopped, press **Enter** to return to the menu.
8. Press **Enter** again to manually configure settings.
9. Select **`Restore Database From Backup`**, then specify the location to restore the backup from in the **`Restore Database File`** menu:
   1. If you saved the database backup file locally as **`/tmp/evm_db.backup`**, select **`Local file`**. You can also restore from a **`Network File System (nfs)`** or **`Samba (smb)`**.
   2. Specify the location of the backup file.

      Note:

      The appliance console menu may respond slowly if connections are open and the server is still shutting down. If this occurs, wait a minute and try again.
10. Enter **`Y`** to keep the database backup after restoring from it. Enter **`N`** to delete it.
11. Press **`Y`** to confirm.
12. After the backup completes, press **Enter** to return to the menu.
13. Press **Enter** again to manually configure settings.
14. Select **`Start EVM Server Processes`** to restart all processes on servers that connect to this VMDB.
15. Enter **`Y`** to confirm.

16. If you are restoring a database backup on a high availability environment, start the replication manager service (**repmgr13.service**). This is not required in other Infrastructure Automation configurations.

```
# systemctl start repmgr13.service
```

Performing a Binary Backup and Restoring the Database

Preserve data at the file system level by performing a binary backup. This includes all databases, users and roles, and other objects.

Note:

This procedure uses the **pg_basebackup** utility to perform a remote database backup and create a full replacement of the PostgreSQL data directory, capturing the exact state of the database when the backup finishes. For more information on the **pg_basebackup** utility, see the PostgreSQL documentation.

Performing a Binary Backup

Create a binary backup and store it as a **gzip** tar file inside the Infrastructure Automation backup directory.

A PostgreSQL superuser or user with *Replication* permissions are required to perform this procedure.

1. SSH into the database server as the **root** user or provide PostgreSQL superuser credentials.
2. Run the **pg_basebackup** command to create the backup:

   ```
   # pg_basebackup -X fetch -h hostname -U root -Ft -z -D filename
   ```

   Note:

   -h *hostname* specifies the IP address of the database server.

   -D *filename* specifies the name of the directory created to contain the backup.

Restoring a Database from the Binary Backup

Restore your PostgreSQL binary backup using the following steps. This process will require stopping both EVM and PostgreSQL services before restoring data.

1. Copy the existing backup to the target VM:

   ```
   # scp filename/base.tar.gz root@hostname:/var/www/miq
   ```

2. SSH to the target VM:

   ```
   # ssh root@hostname
   ```

3. Stop both the EVM and PostgreSQL servers:

   ```
   # systemctl stop evmserverd
   # systemctl stop $APPLIANCE_PG_SERVICE
   ```

4. Rename the existing data directory:

   ```
   # mv /var/lib/pgsql/data /var/lib/pgsql/data.backup
   ```

5. Create a clean data directory:

   ```
   # mkdir /var/lib/pgsql/data
   ```

6. Unzip the tar file to the new directory:

   ```
   # tar -xzf /var/www/miq/base.tar.gz -C /var/lib/pgsql/data
   ```

7. Correct permissions:

   ```
   # chown postgres:postgres /var/lib/pgsql/data
   # chmod 700 /var/lib/pgsql/data
   ```

8. Restart the PostgreSQL and EVM servers:

   ```
   # systemctl start $APPLIANCE_PG_SERVICE
   # systemctl start evmserverd
   ```

Running Database Garbage Collection

The database server collects garbage automatically, but Red Hat may occasionally direct you to run database garbage collection manually in order to reclaim unused space in your VMDB.

To run database garbage collection:

1. Browse to **Settings > Application Settings**.
2. Click the **Diagnostics** accordion and click the **Region** name.
3. Click the **Database** tab.
4. In the **Run Database Garbage Collection Now** box, click **Submit**.

Changing the Database Password

Use the following procedures to change the password on appliances containing an internal database, and on worker appliances.

To change the password for an external database, use the procedure in Changing the Password on the Worker Appliances.

Note:

See Appliance Types in the *Deployment Planning Guide* for a summary of different types of appliances.

Changing the Password on the Database Appliance

Infrastructure Automation provides a default database password for the internal PostgreSQL database.

To change the password, you need to stop the Infrastructure Automation service, change the password for the PostgreSQL database, run a command to change the password in the configuration file that **evmserverd** uses to access the server, and restart the Infrastructure Automation appliance.

1. Stop the Infrastructure Automation service:
    1. SSH into the appliance.
    2. To stop the Infrastructure Automation service, run the following command:

       ```
       service evmserverd stop
       ```
2. Use **pgadmin** to change the password for your Infrastructure Automation database (default is **vmdb_production**). If you do not have **pgadmin**, you can change the password by running:

   ```
   psql -U root -d vmdb_production
   ```
    1. At the vmdb# prompt, type:

       ```
       ALTER USER root WITH PASSWORD 'newpassword';
       ```
    2. To exit **psql**, type:

       ```
       \q
       ```
3. Change the password in the configuration file that **evmserverd** uses to access the server:

   ```
   /var/www/miq/vmdb/tools/fix_auth.rb --databaseyml --password newpassword
   ```
4. Restart the Infrastructure Automation service:

   ```
   service evmserverd start
   ```
5. Verify that you can log in to the Infrastructure Automation console.

Changing the Password on the Worker Appliances

1. Stop the Infrastructure Automation service:
    1. SSH into the appliance.
    2. To stop the Infrastructure Automation service, run the following command:

       ```
       service evmserverd stop
       ```
2. Change the password in the configuration file that **evmserverd** uses to access the server:

   ```
   /var/www/miq/vmdb/tools/fix_auth.rb --databaseyml --password newpassword
   ```
3. Restart the Infrastructure Automation service:

   ```
   service evmserverd start
   ```

   In a high availability environment, if using the same PostgreSQL user for replication, you must also change the password in the `/var/lib/pgsql/.pgpass` file on every database node. Additionally, if the password for the user being used for region-to-region replication is changing, users must also change the password in the replication subscription screen. See [Configuring the Global Copy](#configuring-the-global-copy).

Adding a New Appliance to an Existing Region with a Non-default Password

1. Create the new appliance.
2. Start the appliance, but do not go into any of the configuration options, instead **SSH** into the new appliance.
3. In the **/var/www/miq/vmdb** directory, create a file called **REGION**. Its only contents should be the number of the Region that it is joining. (You could also just copy the **REGION** file from the VMDB appliance.)
4. Edit the **database.yml** file in the **/var/www/miq/vmdb** directory. (You may want to save from the original.)
    1. Replace the contents of the **“production”** section with the contents of the **“base”** section.
    2. Edit the **“host”** parameter to match the IP of the appliance hosting the VMDB.
    3. Save the new **database.yml**.

5. Run the following command to change the password in the configuration file that **evmserverd** uses to access the server:

   ```
   /var/www/miq/vmdb/tools/fix_auth.rb --databaseyml --password newpassword
   ```

6. Restart the new worker appliance:

   ```
   service evmserverd restart
   ```

Creating a Database Dump

Creating a database dump from the appliance console can be used to provide additional information when raising a support case. Options exist to generate a database dump excluding specified tables and to split the database dump into multiple parts.

Database dumps require name or location information based on the **Dump Output File Name** selection:

Local File	Location
Network File System (NFS)	Location
Samba (SMB)	Location
Amazon S3	Name
File Transfer Protocol (FTP)	Location
ftp to dropbox.redhat.com	Name

To create a database dump:

1. SSH into the appliance console as **root**.
2. Press any key to continue.
3. Type **appliance_console** and press **Enter**.
4. From the Advanced Settings, select option 5, **Create a Database Dump**.
5. Select a **Dump Output File Name** from the list.
6. Enter the location or name to save the dump file to.
7. (Optional) Select tables to exclude from the dump:
   1. Enter **Y** to exclude tables. Select **N** to skip the operation.
   2. Enter the table names in a space-separated list:

      ```
      metrics_* vim_performance_states event_streams
      ```

8. (Optional) Split the database dump output into multiple parts:
   1. Enter **Y** to split the database dump output. Select **N** to skip the operation.
   2. Enter the size to split by, ex **250M** or **1G**.

The database dump is created and sent to the specified location.

Enabling Events Syndication

You can enable events syndication in the **Advanced** settings:

1. From the menu, go to **Settings > Application Settings**.
2. Expand the **Settings** accordian. When you select the **Settings** accordion, you are automatically taken to the server list under **Zones**.
3. Select the zone where the EVM server is located, and select the EVM server.
4. Click the **Advanced** tab.

   You can click the **Advanced** tab at the zone level to set advanced configuration settings for the entire zone. Or, you can click on the server, and then click the **Advanced** tab to set advanced configuration settings at the server (appliance) level, which overrides zone-level settings.

5. Find the following configuration:

   ```
   :event_streams:
     :history:
       :keep_events: 6.months
       :purge_window_size: 1000
     :syndicate_events: false
   ```

6. Set the value for `:syndicate_events:` to **true**.
7. Click **Save**.

Enabling Inventory Syndication

You can enable inventory syndication in the **Advanced** settings:

1. From the menu, go to **Settings** > **Application Settings**.
2. Expand the **Settings** accordion. When you select the **Settings** accordion, you are automatically taken to the server list under **Zones**.
3. Select the zone where the EVM server is located and select the EVM server.
4. Click the **Advanced** tab.

   You can click the **Advanced** tab at the zone level to set advanced configuration settings for the entire zone. Or, you can click on the server, and then click the **Advanced** tab to set advanced configuration settings at the server (appliance) level, which overrides zone-level settings.
5. Find the following configuration:

```
:ems_refresh:
 :full_refresh_threshold: 100
 :syndicate_inventory: false
 :raise_vm_snapshot_complete_if_created_within: 15.minutes
 :refresh_interval: 24.hours
```

6. Set the value for `syndicate_inventory` to `true`.

Enabling Metrics Syndication

You can enable metrics syndication in the `Advanced` settings:

1. From the menu, go to **Settings** > **Application Settings**.
2. Expand the **Settings** accordion. When you select the **Settings** accordion, you are automatically taken to the server list under **Zones**.
3. Select the zone where the EVM server is located and select the EVM server.
4. Click the **Advanced** tab.

   You can click the **Advanced** tab at the zone level to set advanced configuration settings for the entire zone. Or, you can click on the server, and then click the **Advanced** tab to set advanced configuration settings at the server (appliance) level, which overrides zone-level settings.
5. Find the following configuration:

```
:performance:
  :host_overhead:
    :memory: 2.01.percent
    :cpu: 0.15.percent
  :syndicate_metrics: false
```

6. Set the value for **:syndicate_metrics:** to `true`.
7. Use **C & U Collection Settings** to select the clusters and datastores that you want to collect usage data for.

SmartProxies

The embedded SmartProxy can analyze virtual machines that are registered to a host and templates that are associated with a provider.

Installing the SmartProxy from the Console

The server comes with one SmartProxy version already available. It can also be installed on an ESX Server version 3.0.2, 3.5 or 4.

Contact Red Hat before installing a new SmartProxy on an ESX Server.
Requirements:

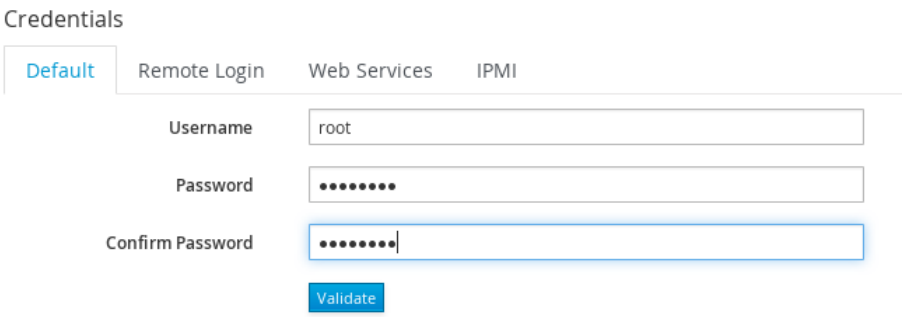
- On ESX, SSH (Secure Shell) must be enabled. This is usually port 22.
- 300 MB free disk space to install and run the SmartProxy.
- Administrator or root credentials.
- The host must already be in the VMDB either by discovery or manually.

Entering Credentials and Operating System for the Target Host

Set the credentials and operating system for the target host to prepare for the installation of SmartProxy.

To Enter Credentials and Operating System for the Target Host:

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Select the host you want to edit.
3. Click **Configuration**, then  **Edit Selected items**.
4. In **Credentials**, click the **Default** tab and enter your login credentials. If you are using domain credentials, the format for User ID must be in the format of **<domainname>\<username>**. For ESX hosts, if SSH login is disabled for the default user, click the **Remote Login** tab and enter a user with remote login access.



If the target is a Windows host, disconnect all network connections between the Windows proxy and the target. If an existing connection uses a different set of credentials than those set in the console, the installation may fail.

5. Click **Validate** to verify the credentials.
6. If you added the host manually instead of **Host Discovery** or **Provider Refresh** finding it, select the host's operating system from the **Host Platform** drop-down box to ensure the host platform is available.
7. Click **Save**.

When remotely installing on Windows hosts, the SmartProxy file is first copied to a Windows proxy. That computer then installs the file to the target host. The Windows proxy is the same as when you select the Default Repository SmartProxy box. You can locate this by navigating to the settings menu and selecting **Configuration**, then clicking on the desired server, then the **Server** tab, and exploring the **Server Control** area.

About

Click the question mark icon ( ) at the top right of the Infrastructure Automation user interface to open a dropdown menu with links to Infrastructure Automation documentation and the Red Hat Customer Portal, and for information about the current Infrastructure Automation session.

- Click **Documentation** to open a list of guides. Click the title of a document to view it as a PDF in a new web browser tab.
- Click **Red Hat Customer Portal** to open the Red Hat Customer Portal in a new web browser tab.
- Click **About** to open a pop-up window showing information about the Infrastructure Automation version and the session of the logged-in user.

Default Roles

This section outlines the default roles provided in Infrastructure Automation and the product features to which they are granted access.

EvmRole-super_administrator

The **`EvmRole-super_administrator`** is granted access to the following product features.

Product Feature
everything

EvmRole-super_administrator

EvmRole-administrator

The **`EvmRole-administrator`** is granted access to the following product features.

Product Feature
about
all_vm_rules
automation_manager
embedded_automation_manager
availability_zone
host_aggregate
compute
flavor
floating_ip
chargeback

Product Feature
catalog
cloud_network
cloud_object_store_container
cloud_object_store_object
control_explorer
dashboard
datacenter
storage
storage_pod
ems_cloud
ems_network
ems_cluster
ems_infra
ems_infra_admin_ui
ems_physical_infra
container_dashboard
ems_container
ems_container_deployment
configuration_job
container_project
container_route
container_service
container_replicator
container_group
container
container_node
persistent_volume
container_build
container_image_registry
container_image
container_topology
cloud_topology
infra_topology
host
infra_networking
miq_ae_class_explorer
miq_ae_customization_explorer
miq_ae_class_import_export
miq_ae_class_log
miq_ae_class_simulation
miq_task_all_ui
miq_task_my_ui
my_settings
policy_simulation
policy_log
miq_report
miq_request
miq_template
orchestration_stack
physical_server
physical_infra_topology
planning
policy_import_export
provider_foreman_explorer
pxe
resource_pool
security_group
service
timeline
usage
utilization
vm_explorer
vm
vm_cloud_explorer

Product Feature
vm_infra_explorer
sui_services
sui_notifications

EvmRole-administrator

EvmRole-approver

The **`EvmRole-approver`** is granted access to the following product features.

Product Feature
about
all_vm_rules
compute
chargeback
chargeback_reports
customization_template_view
iso_datastore_view
control_explorer
dashboard
ems_cluster_show
ems_cluster_show_list
ems_cluster_perf
ems_cluster_tag
ems_cluster_timeline
ems_infra_show
ems_infra_show_list
ems_infra_tag
ems_infra_timeline
ems_physical_infra_console
ems_physical_infra_tag
ems_physical_infra_view
ems_physical_infra_tag
host_show
host_show_list
host_perf
host_tag
host_timeline
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
miq_report_run
miq_report_saved_reports
miq_report_schedules
miq_report_view
miq_request_control
miq_request_view
miq_task_my_ui
miq_template_check_compliance
miq_template_perf
miq_template_policy_sim
miq_template_show
miq_template_show_list
miq_template_snapshot
miq_template_tag
miq_template_timeline
physical_infra_topology_view
physical_server_view
policy_log
policy_simulation
pxe_image_type_view
pxe_server_view
resource_pool_show
resource_pool_show_list

Product Feature
resource_pool_tag
service_view
storage_show
storage_show_list
storage_pod_show
storage_pod_show_list
storage_perf
storage_tag
timeline
usage
vm_check_compliance
vm_console
vm_webmks_console
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_vnc_console
vm_vmrc_console
cockpit_console
vm_perf
vm_policy_sim
vm_show
vm_show_list
vm_snapshot
vm_tag
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_tags
sui_orders_view
sui_notifications

EvmRole-approver

EvmRole-auditor

The **`EvmRole-auditor`** is granted access to the following product features.

Product Feature
about
automation_manager
embedded_automation_manager
compute
chargeback
chargeback_reports
customization_template_view
iso_datastore_view
control_explorer
dashboard
ems_cluster_view
ems_cluster_tag
ems_infra_view
ems_infra_tag
ems_infra_check_compliance
ems_physical_infra_console
ems_physical_infra_tag
ems_physical_infra_view
host_show
host_show_list
host_perf
infra_networking_view

Product Feature
infra_networking_tag
instance_view
instance_check_compliance
instance_policy_sim
instance_tag
image_view
image_check_compliance
image_policy_sim
image_tag
iso_datastore_view
host_view
host_check_compliance
host_tag
miq_task_my_ui
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
miq_report_run
miq_report_saved_reports
miq_report_schedules
miq_report_view
miq_template_view
miq_template_check_compliance
miq_template_policy_sim
miq_template_snapshot_view
miq_template_tag
physical_infra_topology_view
physical_server_view
planning
policy_log
policy_profile
policy_simulation
pxe_image_type_view
pxe_server_view
resource_pool_show
resource_pool_show_list
resource_pool_tag
service_view
storage_show
storage_show_list
storage_perf
storage_tag
storage_pod_show
storage_pod_show_list
timeline
usage
utilization
vm_view
vm_check_compliance
vm_compare
vm_console
vm_drift
vm_webmks_console
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_vnc_console
vm_vmrc_console
cockpit_console
vm_perf
vm_show
vm_show_list
vm_snapshot_view
vm_tag

Product Feature
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_tags
sui_notifications

EvmRole-auditor

EvmRole-desktop

The **`EvmRole-desktop`** is granted access to the following product features.

Product Feature
about
all_vm_rules
automation_manager
compute
dashboard
ems_physical_infra
miq_request_admin
miq_request_view
miq_template_clone
miq_template_compare
miq_template_drift
miq_template_edit
miq_template_refresh
miq_template_miq_request_new
miq_template_perf
miq_template_publish
miq_template_show
miq_template_show_list
miq_template_timeline
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
physical_server
physical_infra_topology
physical_server_view
provider_foreman_explorer
vm_clone
vm_compare
vm_console
vm_webmks_console
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_vnc_console
vm_vmrc_console
cockpit_console
vm_drift
vm_edit
vm_refresh
vm_reset
vm_guest_restart
vm_miq_request_new
vm_perf
vm_publish
vm_show
vm_show_list
vm_guest_shutdown
vm_start

Product Feature
vm_stop
vm_suspend
vm_pause
vm_shelve
vm_shelve_offload
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_start
sui_vm_stop
sui_vm_suspend
sui_orders_view
sui_orders_operations
sui_notifications

EvmRole-desktop

EvmRole-operator

The **`EvmRole-operator`** is granted access to the following product features.

Product Feature
about
all_vm_rules
automation_manager
embedded_automation_manager
compute
chargeback
chargeback_reports
dashboard
datastore
ems_cluster_analyze
ems_cluster_compare
ems_cluster_drift
ems_cluster_show
ems_cluster_show_list
ems_cluster_perf
ems_cluster_tag
ems_cluster_timeline
ems_infra_new
ems_infra_delete
ems_infra_discover
ems_infra_edit
ems_infra_refresh
ems_infra_scale
ems_infra_show
ems_infra_show_list
ems_infra_tag
ems_infra_timeline
ems_physical_infra_new
ems_physical_infra_console
ems_physical_infra_delete
ems_physical_infra_discover
ems_physical_infra_edit
ems_physical_infra_refresh
ems_physical_infra_tag
ems_physical_infra_view
physical_server_view
physical_infra_topology_view
host_new
host_analyze

Product Feature
host_compare
host_discover
host_drift
host_edit
host_refresh
host_show
host_show_list
host_perf
host_tag
host_timeline
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
miq_report_run
miq_report_saved_reports
miq_report_schedules
miq_report_view
miq_task_my_ui
miq_template_analyze
miq_template_check_compliance
miq_template_compare
miq_template_drift
miq_template_edit
miq_template_perf
miq_template_refresh
miq_template_show
miq_template_show_list
miq_template_snapshot
miq_template_sync
miq_template_tag
miq_template_timeline
pxe
resource_pool_show
resource_pool_show_list
resource_pool_tag
service_view
provider_foreman_explorer
storage_delete
storage_scan
storage_show
storage_show_list
storage_perf
storage_tag
timeline
usage
vm_analyze
vm_check_compliance
vm_collect_running_processes
vm_compare
vm_console
vm_webmks_console
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_vnc_console
vm_vmrc_console
cockpit_console
vm_drift
vm_edit
vm_perf
vm_refresh
vm_reset
vm_guest_restart
vm_show

Product Feature
vm_show_list
vm_guest_shutdown
vm_snapshot
vm_start
vm_stop
vm_suspend
vm_pause
vm_shelve
vm_shelve_offload
vm_sync
vm_tag
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_tags
sui_vm_start
sui_vm_stop
sui_vm_suspend
sui_notifications

EvmRole-operator

EvmRole-security

The **`EvmRole-security`** is granted access to the following product features.

Product Feature
about
all_vm_rules
compute
chargeback
chargeback_reports
control_explorer
dashboard
datastore
ems_cluster_show
ems_cluster_show_list
ems_cluster_perf
ems_cluster_tag
ems_cluster_timeline
ems_infra_show
ems_infra_show_list
ems_infra_tag
ems_infra_timeline
ems_physical_infra_tag
ems_physical_infra_view
physical_server_timeline
host_show
host_show_list
host_perf
host_tag
host_timeline
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
miq_report_run
miq_report_saved_reports
miq_report_schedules
miq_report_view
miq_task_my_ui
miq_template_check_compliance

Product Feature
miq_template_compare
miq_template_drift
miq_template_perf
miq_template_policy_sim
miq_template_show
miq_template_show_list
miq_template_snapshot_add
miq_template_snapshot_delete
miq_template_snapshot_delete_all
miq_template_snapshot_revert
miq_template_tag
miq_template_timeline
policy_log
policy_simulation
resource_pool_show
resource_pool_show_list
resource_pool_tag
service_view
storage_show
storage_show_list
storage_perf
storage_tag
timeline
usage
vm_check_compliance
vm_compare
vm_drift
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_perf
vm_policy_sim
vm_show
vm_show_list
vm_snapshot_add
vm_snapshot_delete
vm_snapshot_delete_all
vm_snapshot_revert
vm_tag
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_snapshot_create
sui_vm_snapshot_delete
sui_vm_tags
sui_notifications

EvmRole-security

EvmRole-support

The **`EvmRole-support`** is granted access to the following product features.

Product Feature
about
all_vm_rules
compute
chargeback
chargeback_reports
control_explorer
dashboard
datastore
ems_cluster_show

Product Feature
ems_cluster_show_list
ems_cluster_perf
ems_cluster_tag
ems_cluster_timeline
ems_infra_show
ems_infra_show_list
ems_infra_tag
ems_infra_timeline
ems_physical_infra_console
ems_physical_infra_tag
ems_physical_infra_view
host_show
host_show_list
host_perf
host_tag
host_timeline
miq_task_my_ui
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
miq_report_run
miq_report_saved_reports
miq_report_schedules
miq_report_view
miq_template_check_compliance
miq_template_compare
miq_template_drift
miq_template_perf
miq_template_policy_sim
miq_template_show
miq_template_show_list
miq_template_snapshot
miq_template_tag
miq_template_timeline
physical_infra_topology_view
physical_server_view
policy_log
policy_simulation
resource_pool_show
resource_pool_show_list
resource_pool_tag
service_view
storage_show
storage_show_list
storage_perf
storage_tag
timeline
usage
vm_check_compliance
vm_collect_running_processes
vm_console
vm_webmks_console
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_vnc_console
vm_vmrc_console
cockpit_console
vm_compare
vm_drift
vm_perf
vm_policy_sim
vm_show
vm_show_list

Product Feature
vm_snapshot
vm_tag
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_tags
sui_notifications

EvmRole-support

EvmRole-user

The **`EvmRole-user`** is granted access to the following product features.

Product Feature
about
all_vm_rules
compute
chargeback
chargeback_reports
dashboard
datastore
ems_cluster_show
ems_cluster_show_list
ems_cluster_perf
ems_cluster_tag
ems_cluster_timeline
ems_infra_show
ems_infra_show_list
ems_infra_tag
ems_infra_timeline
ems_physical_infra_console
ems_physical_infra_show
ems_physical_infra_show_list
ems_physical_infra_tag
ems_physical_infra_timeline
host_show
host_show_list
host_perf
host_tag
host_timeline
miq_task_my_ui
my_settings_default_views
my_settings_time_profiles
my_settings_visuals
miq_report_run
miq_report_saved_reports
miq_report_schedules
miq_report_view
miq_request_admin
miq_request_view
miq_template_check_compliance
miq_template_compare
miq_template_drift
miq_template_perf
miq_template_show
miq_template_show_list
miq_template_snapshot
miq_template_tag
miq_template_timeline
physical_infra_topology_view

Product Feature
physical_server_view
resource_pool_show
resource_pool_show_list
resource_pool_tag
service_view
storage_show
storage_show_list
storage_perf
storage_tag
timeline
usage
vm_check_compliance
vm_console
vm_webmks_console
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_vnc_console
vm_vmrc_console
cockpit_console
vm_compare
vm_drift
vm_perf
vm_show
vm_show_list
vm_snapshot
vm_tag
vm_timeline
vm_chargeback
sui_services_view
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_tags
sui_orders_view
sui_orders_operations
sui_notifications

EvmRole-user

EvmRole-user_limited_self_service

The **`EvmRole-user_limited_self_service`** is granted access to the following product features.

Product Feature
about
all_vm_rules
catalog_items_view
compute
miq_request_admin
miq_request_view
my_settings_default_views
my_settings_visuals
service_edit
service_delete
service_reconfigure
service_tag
service_retire_now
service_view
svc_catalog_provision
vm_clone
vm_cloud_explorer
vm_filter_accord
vm_guest_restart

Product Feature
vm_guest_shutdown
vm_infra_explorer
vm_miq_request_new
vm_publish
vm_reset
vm_retire_now
vm_show
vm_show_list
vm_snapshot_view
vm_start
vm_stop
vm_suspend
vm_pause
vm_shelve
vm_shelve_offload
vm_tag
sui_core
sui_services
sui_vm
sui_orders
sui_svc_catalog_view
sui_svc_catalog_cart
sui_cart

EvmRole-user_limited_self_service

EvmRole-user_self_service

The **`EvmRole-user_self_service`** is granted access to the following product features.

Product Feature
about
all_vm_rules
automation_manager
embedded_automation_manager
ems_physical_infra_console
catalog_items_view
compute
miq_template_clone
miq_template_drift
miq_template_edit
miq_template_perf
miq_template_refresh
miq_template_show
miq_template_show_list
miq_template_snapshot
miq_template_sync
miq_template_tag
miq_template_timeline
my_settings_default_views
my_settings_visuals
miq_request_admin
miq_request_view
provider_foreman_explorer
service_edit
service_delete
service_reconfigure
service_tag
service_retire_now
service_view
svc_catalog_provision
vm_console
vm_webmks_console
vm_clone

Product Feature
vm_cloud_explorer
vm_console
vm_webmks_console
vm_drift
vm_edit
vm_filter_accord
vm_guest_restart
vm_guest_shutdown
vm_infra_explorer
vm_miq_request_new
vm_perf
vm_publish
vm_refresh
vm_reset
vm_retire_now
vm_show
vm_show_list
vm_snapshot
vm_start
vm_stop
vm_suspend
vm_pause
vm_shelve
vm_shelve_offload
vm_sync
vm_tag
vm_timeline
vm_chargeback
vm_vmrc_console
vm_vnc_console
cockpit_console
sui

EvmRole-user_self_service

EvmRole-vm_user

The **`EvmRole-vm_user`** is granted access to the following product features.

Product Feature
about
all_vm_rules
automation_manager
embedded_automation_manager
compute
miq_request_admin
miq_request_view
miq_template_analyze
miq_template_check_compliance
miq_template_clone
miq_template_drift
miq_template_edit
miq_template_perf
miq_template_show
miq_template_show_list
miq_template_snapshot
miq_template_sync
miq_template_tag
miq_template_timeline
my_settings_default_views
my_settings_visuals
provider_foreman_explorer
vm_analyze
vm_check_compliance

Product Feature
vm_clone
vm_cloud_explorer
vm_collect_running_processes
vm_compare
vm_console
vm_webmks_console
vm_drift
vm_explorer
vm_guest_restart
vm_guest_shutdown
vm_infra_explorer
vm_miq_request_new
vm_perf
vm_policy_sim
vm_publish
vm_refresh
vm_reset
vm_retire_now
vm_show
vm_show_list
vm_snapshot
vm_start
vm_stop
vm_suspend
vm_pause
vm_shelve
vm_shelve_offload
vm_sync
vm_tag
vm_timeline
vm_chargeback
vm_vmrc_console
vm_vnc_console
cockpit_console
sui_vm_details_view
sui_vm_console
sui_vm_web_console
sui_vm_tags
sui_vm_retire
sui_vm_start
sui_vm_stop
sui_vm_suspend
sui_orders_view
sui_orders_operations
sui_notifications

EvmRole-vm_user

EvmRole-tenant_administrator

The **`EvmRole-tenant_administrator`** is granted access to the following product features.

Product Feature
about
automation_manager
embedded_automation_manager
availability_zone
host_aggregate
all_vm_rules
compute
cloud_network
cloud_subnet
flavor
floating_ip

Product Feature
chargeback
catalog
cloud_tenant
control_explorer
dashboard
datacenter
storage
ems_cloud
ems_network
ems_cluster
ems_infra
ems_physical_infra
host
load_balancer
miq_ae_class_explorer
miq_ae_customization_explorer
miq_ae_class_import_export
miq_ae_class_log
miq_ae_class_simulation
miq_task_all_ui
miq_task_my_ui
my_settings
network_port
network_router
policy_simulation
policy_log
miq_report
miq_request
miq_template
orchestration_stack
planning
policy_import_export
provider_foreman_explorer
pxe
rbac_group
rbac_role_view
rbac_tenant
rbac_user
resource_pool
security_group
service
timeline
usage
utilization
vm_explorer
vm
vm_cloud_explorer
vm_infra_explorer
sui_services
sui_notifications

EvmRole-tenant_administrator

EvmRole-tenant_quota_administrator

The **`EvmRole-tenant_quota_administrator`** is granted access to the following product features.

Product Feature
about
automation_manager
embedded_automation_manager
availability_zone
host_aggregate
all_vm_rules

Product Feature
flavor
compute
chargeback
catalog
cloud_tenant
control_explorer
dashboard
datacenter
storage
ems_cloud
ems_cluster
ems_infra
ems_physical_infra
host
miq_ae_class_explorer
miq_ae_customization_explorer
miq_ae_class_import_export
miq_ae_class_log
miq_ae_class_simulation
miq_task_all_ui
miq_task_my_ui
my_settings
policy_simulation
policy_log
miq_report
miq_request
miq_template
orchestration_stack
planning
policy_import_export
provider_foreman_explorer
pxe
rbac_tenant_view
rbac_tenant_manage_quotas
resource_pool
security_group
service
timeline
usage
utilization
vm_explorer
vm
vm_cloud_explorer
vm_infra_explorer
sui_services
sui_notifications

EvmRole-tenant_quota_administrator

EvmRole-consumption_administrator

The **`EvmRole-consumption_administrator`** is granted access to the following product features.

Product Feature
dashboard
chargeback
miq_report

EvmRole-consumption_administrator

EvmRole-container_administrator

The **`EvmRole-container_administrator`** is granted access to the following product features.

Product Feature
vms_filter_accord
instances_filter_accord
datacenter_controller
storage
storage_pod
dashboard
miq_report
consumption
chargeback
pictures
control_explorer
generic_object
generic_object_definition
my_settings
tasks
about
ems_container
container_group
container_node
container_replicator
container_image
container_image_registry
persistent_volume
container_build
container_template
container_service
container_route
container_project
container
container_topology
container_dashboard
ems_infra_dashboard
instance_view
vm_view
miq_cloud_networks
miq_arbitration_settings
miq_arbitration_rules
redhat_access_insights_admin
ems_container_ad_hoc_metrics
monitor
monitor_alerts
alert_status
alert_action
ems_infra
rbac_user
ops_settings

EvmRole-container_administrator

EvmRole-container_operator

The **`EvmRole-container_operator`** is granted access to the following product features.

Product Feature
vms_filter_accord
instances_filter_accord
dashboard_view
miq_report_saved_reports_view
miq_report_view
miq_report_control
chargeback_reports
my_settings
tasks
ems_container_view

Product Feature
ems_container_check_compliance
container_group_view
container_group_check_compliance
container_node_view
container_node_check_compliance
container_replicator_check_compliance
container_image_view
container_image_scan
container_image_check_compliance
container_image_registry_view
persistent_volume_view
container_build_view
container_template_view
container_service_view
container_route_view
container_project_view
container_filter_accord
container_view
container_control
container_topology
container_dashboard
instance_view
vm_view
ems_container_ad_hoc_metrics
monitor
monitor_alerts
alert_status
alert_action
ems_infra
rbac_user
ops_settings

EvmRole-container_operator

EvmRole-reader

The **`EvmRole-reader`** is granted access to the following product features.

Product Feature
auth_key_pair_cloud_view
automation_manager_configuration_script_view
automation_manager_configured_system_view
automation_manager_providers_view
availability_zone_view
catalog_items_view
cloud_network_view
cloudobject_store_container_view
cloud_object_store_object_view
cloud_subnet_view
cloud_tenant_view
cloud_topology
cloud_volume_backup_view
cloud_volume_snapshot_view
cloud_volume_view
configuration_job_view
configuration_script_view
configured_systems_filter_accord_view
container_build_view
container_dashboard
container_filter_accord
container_group_view
container_image_registry_view
container_image_view
container_node_view

Product Feature
container_project_view
container_replicator_view
container_route_view
container_service_view
container_template_view
container_topology
container_view
customization_template_view
dashboard_view
embedded_automation_manager_credentials_view
embedded_configuration_script_payload_view
embedded_configuration_script_source_view
ems_block_storage_view
ems_cloud_view
ems_cluster_view
ems_container_view
ems_infra_view
ems_physical_infra_view
ems_network_view
ems_object_storage_view
flavor_view
floating_ip_view
host_aggregate_view
host_view
image_view
infra_networking_view
infra_topology
instance_view
iso_datastore_view
load_balancer_view
miq_ae_class_log
miq_ae_domain_view
miq_report_saved_reports_view
miq_report_schedule_view
miq_report_view
miq_request_view
miq_template_snapshot_view
miq_template_view
my_settings
network_port_view
network_router_view
network_topology
orchestration_stack_view
orchestration_templates_view
persistent_volume_view
physical_infra_topology_view
physical_server_view
planning
policy_log
provider_foreman_view
pxe_image_type_view
pxe_server_view
rbac_group_view
rbac_role_view
rbac_tenant_view
rbac_user_view
redhat_access_insights_overview
resource_pool_view
security_group_view
service_view
st_catalog_view
storage_view
tasks
timeline

Product Feature
utilization
virtual_template_show
vm_cloud_explorer
vm_explorer
vm_infra_explorer
vm_snapshot_view
vm_view
sui_services_view
sui_orders_view
sui_notifications

EvmRole-reader

- Managing Infrastructure and Inventory
- Clusters
- Performing SmartState Analysis on Clusters
  - Comparing Clusters
- Creating a Cluster Comparison Report
- Viewing a Cluster
- Tagging Clusters
- Viewing Capacity and Utilization Charts for a Cluster
- Viewing Cluster Timeline
- Detecting Drift on Clusters
- Creating a Drift Report for Clusters
- Removing Clusters
- Hosts
  - Filtering Hosts
- Setting a Default Host Filter
- Creating a Host Filter
- Performing SmartState Analysis on Hosts
  - Comparing Hosts
- Host Comparison Sections
- Using the Host Comparison Sections
- Creating a Host Comparison Report
- Refreshing Multiple Hosts
- Adding a Single Host
- Editing Hosts
- Viewing a Host
- Tagging Multiple Hosts
- Removing Hosts
  - Scaling Down Compute Hosts
- Refreshing Relationships and Power States for a Host
- Viewing Capacity and Utilization Charts for a Host
- Viewing the Host Timeline
- Host Virtual Summary
- Viewing Host Device Information
- Viewing Host Network Information
- Viewing Storage Adapters
- Detecting Drift on Hosts
- Creating a Drift Report for Hosts
- Virtual Machines
  - Filtering Virtual Machines and Templates
- Creating a Virtual Machine or Template Filter
- Loading a Report Filter or Search Expression
- Changing Views for Virtual Machines and Templates
- Sorting Virtual Machines and Templates
- Creating a Virtual Machine or Template Report
- Searching for Virtual Machines or Templates
  - Analyzing Virtual Machines and Templates
  - Red Hat Enterprise Virtualization Prerequisites
- SmartState Analysis on Red Hat Enterprise Virtualization Manager 3.1 and Above - Storage Support Notes
- SmartState Analysis on Red Hat Enterprise Virtualization Manager 3.0 - Storage Support Notes
  - VMware vSphere Prerequisites
  - Comparing Virtual Machines and Templates
- Virtual Machine and Templates Comparison Sections
- Using the Virtual Machine Comparison Sections
- Creating a Virtual Machine Comparison Report
- Controlling the Power State of Red Hat Virtualization Virtual Machines
- Refreshing Virtual Machines and Templates

- Extracting Running Processes from Virtual Machines and Templates
- Setting Ownership for Virtual Machines and Templates
- Removing Virtual Machines and Templates from the VMDB
- Tagging Virtual Machines and Templates
- Viewing a VMware Virtual Machine's Storage Profile
- Viewing Running Processes after Collection
- Editing Virtual Machine or Template Properties
- Setting Ownership of a Virtual Machine or Template
- Right Sizing a Virtual Machine
- Viewing Capacity and Utilization Charts for a Virtual Machine
- Viewing the Virtual Machine or Template Timeline
- Virtual Machine or Template Summary
- Viewing the Operating System Properties
- Viewing Virtual Machine or Template Snapshot Information
- Viewing User Information for a Virtual Machine or Template
- Viewing Group Information for a Virtual Machine or Template
- Viewing Genealogy of a Virtual Machine or Template
- Comparing Genealogy of a Virtual Machine or Template
- Tagging Virtual Machines or Templates with a Common Genealogy
- Detecting Drift on Virtual Machines or Templates
- Creating a Drift Report for a Virtual Machine or Template
- Viewing Analysis History for a Virtual Machine or Template
- Viewing Disk Information for a Virtual Machine or Template
- Reconfiguring Virtual Machines
- Adding a Disk to a Virtual Machine
- Removing a Virtual Machine Disk
- Resizing a Virtual Machine Disk
- Increasing or Decreasing a Virtual Machine's Memory Size
- Reconfigure a Virtual Machine's Processors
- Adding or Removing Virtual Machine Network Adapters
  - Adding a Network Adapter to a VMware Virtual Machine
  - Removing a Network Adapter from a VMware Virtual Machine
- Attach or Remove an ISO (VMware Virtual Machines Only)
- Viewing Event Logs for a Virtual Machine or Template
- Remote Consoles
  - Configuring Console Access to VMware ESXi Hosts At A Network Layer
    - Using SSH to Configure VMware ESXi Hosts to Enable Console Access
    - Using the VMware vCenter Web Interface to Configure ESXi Hosts to Enable Console Access
    - Configuring the VMware ESXi Host Firewall Ports for Console Access
  - Configuring WebMKS Support in Infrastructure Automation
  - Opening a Console for a Virtual Machine
- Managing Virtual Machine Snapshots
- Creating a Virtual Machine Snapshot
- Deleting a Virtual Machine Snapshot
- Reverting a Virtual Machine to a Snapshot
- Creating a Template Based on a Virtual Machine
- Retiring Virtual Machines
- Migrating Virtual Machines
- Resource Pools
- Removing a Resource Pool
- Tagging a Resource Pool
- Viewing the Resource Pool Summary
- Resource Pools Accordion
- Datastores
- Performing SmartState Analysis on Datastores
- Viewing a Datastore
- Tagging a Datastore
- Viewing Capacity and Utilization Charts for a Datastore
- Removing a Datastore
- Data Optimization
  - Utilization Trends
    - Viewing a Utilization Trend Summary
    - Viewing Detail Lines of a Utilization Trend
    - Viewing a Report of a Utilization Trend
- PXE Servers
- Availability Zones
- Viewing an Availability Zone
- Viewing Availability Zone Relationships
- Cloud Tenants

    - Tenant Mapping
- Viewing Cloud Tenant Relationships
- Volumes
    - Amazon Elastic Block Store Manager Volumes
- Creating Volumes
- Creating a Snapshot of a Volume
- Attaching a Volume to an Instance
- Detaching a Volume from an Instance
- Editing a Volume
- Deleting a Volume
    - OpenStack Block Storage Manager Volumes
- Creating Volumes
- Creating a Backup of a Volume
- Restoring a Volume from a Backup
- Restoring a Cloud Volume from a Backup
- Deleting a Cloud Volume Backup
- Creating a Snapshot of a Volume
- Attaching a Volume to an Instance
- Detaching a Volume from an Instance
- Editing a Volume
- Deleting a Volume
    - IBM Power Systems Virtual Servers Block Storage Manager Volumes
- Creating Volumes
- Attaching a Volume to an Instance
- Detaching a Volume from an Instance
- Deleting a Volume
- Flavors
- Creating a Flavor
- Viewing a Flavor
- Viewing Flavor Relationships
- Deleting a Flavor
- Cloud Networks
    - Creating and Administering Cloud Networks
        - Adding a Cloud Network
        - Editing Cloud Network Details
        - Deleting a Cloud Network
    - Creating and Administering Subnets
        - Adding a Subnet
        - Editing a Cloud Subnet
    - Configuring Network Routers
        - Adding a Network Router
        - Editing Network Router Details
        - Adding an Interface to a Network Router
        - Removing a Network Router Interface
        - Deleting a Network Router
    - Creating Floating IPs
        - Adding Floating IPs.
        - Managing Port Association of a Floating IP
        - Deleting a Floating IP
    - Security Groups
        - Editing Security Group Details
        - Viewing Security Groups
        - Tagging Security Groups
        - Deleting a Security Group
- Instances and Images
    - Filtering Instances and Images
- Using an Instance or Image Filter
- Creating an Instance or Image Filter
- Loading a Report Filter or Search Expression
- Changing Views for Instances and Images
- Sorting Instances and Images
- Creating an Instance or Image Report
- Searching for Instances or Images
- Analyzing Instances and Images with SmartState Analysis
    - Restrictions on Displaying Files Collected
    - Comparing Instances and Images
- Creating an Instance Comparison Report
- Refreshing Instances and Images
- Extracting Running Processes from Instances and Images

- Setting Ownership for Instances and Images
- Removing Instances and Images from the VMDB
- Tagging Instances and Images
- Reviewing an Instance or Image
- Viewing Running Processes after Collection
- Managing Security Groups for Cloud Instances
- Editing Instance or Image Properties
- Controlling the Power State of an Instance
- Right Sizing an Instance
  - Resizing an Instance
  - Migrating a Live Instance
  - Evacuating an Instance
- Viewing Capacity and Utilization Charts for an Instance
- Viewing the Instance or Image Timeline
- Viewing the Instance or Image Summary
- Viewing User Information for an Instance or Image
- Viewing Group Information for an Instance or Image
- Viewing Genealogy of an Instance or Image
- Detecting Drift on Instances or Images
- Creating a Drift Report for an Instance or Image
- Viewing Analysis History for an Instance or Image
- Viewing Event Logs for an Instance or Image
- Orchestration Stacks
- Tagging Orchestration Stacks
- Retiring Orchestration Stacks
- Removing Orchestration Stacks
- Key Pairs
- Adding a New Key Pair
- Removing a Key Pair
- Downloading Key Pairs
- Object Stores
- Tagging Object Stores
- VMware Networking Switches
- Tagging VMware Networking Switches
- Container Entities
  - The Containers Overview Page
  - Viewing a Container Entity Summary
  - Using the Topology Widget
    - Viewing the Topology for Container Providers
    - Viewing the Topology for Container Provider Projects
    - Limiting the Number of Containers Shown in the Topology View
  - Analyzing Container Images with SmartState Analysis
  - Configuring Automatic Tagging for Container Entities

Managing Infrastructure and Inventory

Clusters

Clusters provide high availability and load balancing for a group of hosts. The **Clusters** page under **Compute > Infrastructure** displays the clusters discovered in your enterprise environment.

Note:

Any filter applied will be in effect here.



Use the **Clusters Taskbar** to manage the analysis and tagging of your clusters. These buttons manage multiple clusters at one time. To manage one cluster, click on that cluster in the main area of the screen.

Performing SmartState Analysis on Clusters

Analyze a cluster to gather historical data to compare with previous points in time.

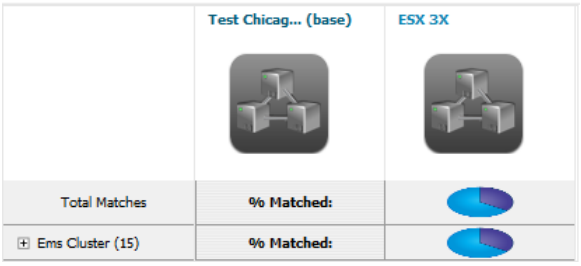
1. Browse to menu: **Compute > Infrastructure > Clusters**.
2. Check the clusters to analyze.
3. Click **Configuration**, and then (**Perform SmartState Analysis**).
4. Click **OK**.

The SmartState Analysis begins and returns the current data.

Comparing Clusters

Infrastructure Automation provides features to compare properties of clusters.

1. Browse to menu: **Compute > Infrastructure > Clusters**.
2. Check the clusters to compare.
3. Click **Configuration**, and then (**Compare Selected items**). The comparison displays in a default expanded view and lists a limited set of properties.



4. To delete a cluster from the comparison, click (**Remove this Cluster from the Comparison**).
5. To go to a compressed view, click (**Compressed View**). To return to an expanded view, click (**Expanded View**).
6. To change the base cluster that all other clusters compare to, click its label at the top of its column.
7. To go to the cluster summary screen, click its virtual thumbnail or icon.
8. There are three buttons in the taskbar to limit the type of views:
   - Click (**All attributes**) to see all attributes.
   - Click (**Attributes with different values**) to see only the attributes that are different across clusters.
   - Click (**Attributes with the same values**) to see only the attributes that are the same across clusters.
9. To limit the mode of the view, there are two taskbar buttons.
   - Click (**Details Mode**) to see all details for an attribute.

- Click ± (**Exists Mode**) to only see if an attribute exists compared to the base or not. This only applies to attributes that can have a Boolean property. For example, a user account exists or does not exist, or a piece of hardware that does or does not exist.

This creates a comparison between clusters. Export this data or create a report from your comparison for analysis using external tools.

Creating a Cluster Comparison Report

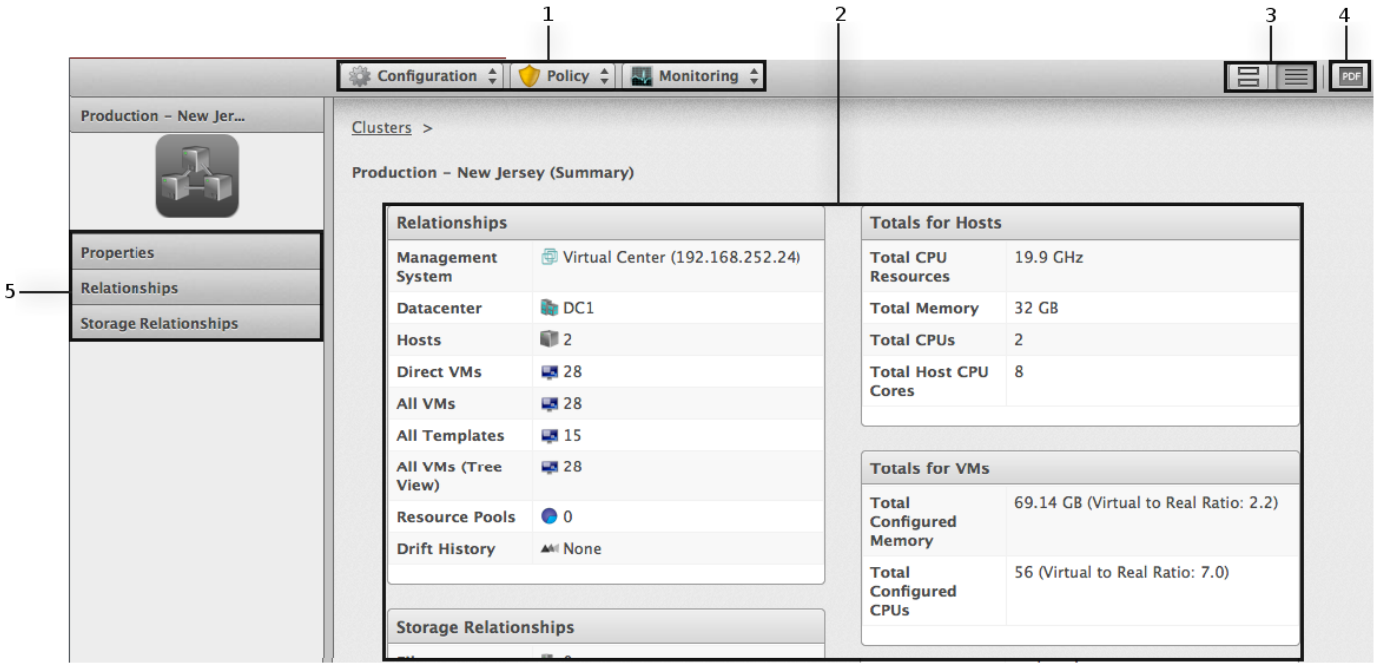
Create a quick report of to compare clusters in CSV, TXT, or PDF formats.

1. Create the comparison to analyze.
2. Click (**Download**).
3. Click the output button for the type of report.
   - Click (**Download comparison report in TXT format**) for a text file.
   - Click (**Download comparison report in CSV format**) for a comma-separated file.
   - Click (**Download comparison report in PDF format**) for a PDF file.

Viewing a Cluster

You can click on a specific cluster to view its details. The screen provides you with a cluster taskbar, a cluster accordion, and a cluster summary.

Cluster Management Screen.



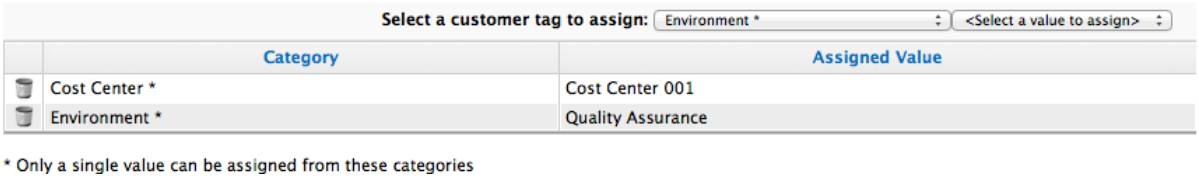
1. **Cluster Taskbar**: Choose between **Configuration**, **Policy** and **Monitoring** options for the selected cluster
2. **Cluster Summary**: See cluster summary such as **Relationships**, **Totals for Hosts**, **Totals for VMs**
3. **Cluster Summary Views**: Choose between graphical or text view of the cluster summary
4. **Cluster Summary PDF**: Generates cluster summary in PDF format
5. **Cluster Accordion**: See details about **Properties**, **Relationships**, **Storage Relationships** for the selected cluster

Tagging Clusters

Use tags to categorize clusters.

1. Browse to menu: **Compute > Infrastructure > Clusters**.

2. Check the Clusters to tag.
3. Click **Policy**, and then **Edit Tags**.

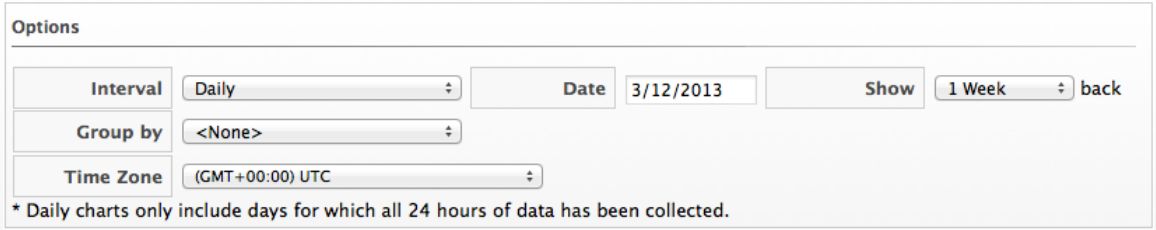


4. Select a customer tag from the first dropdown, and then a value for the tag.
5. Select more tags or click **Save** to save your changes.

Viewing Capacity and Utilization Charts for a Cluster

View capacity and utilization for a cluster.

1. Browse to menu: **Compute > Infrastructure > Clusters**.
2. Click the cluster to view Capacity and Utilization data.
3. Click (**Monitoring**), and then (**Utilization**) or from the accordion menu, click **Properties > Capacity & Utilization**.



4. From **Interval**, select to view hourly or daily data points and the dates to view data. Use **Group by** to group the lines by SmartTags. Use **Time Profiles** to select a time range for the data.



The **Capacity & Utilization** charts display

Note:

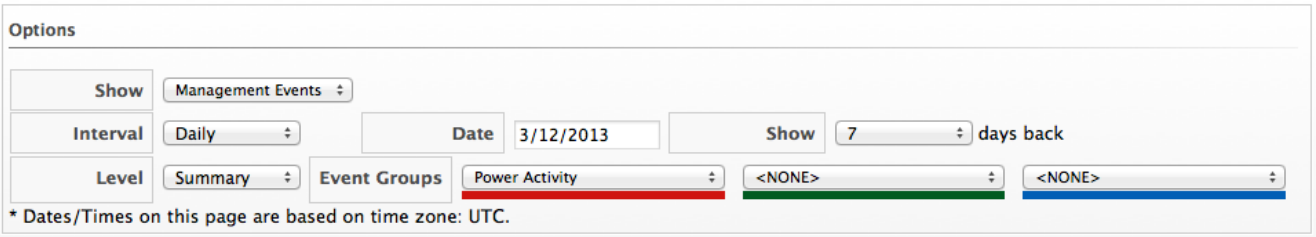
Daily charts only include full days of data. If a day does not include all the 24 data points for a day, the data does not show for that day.

For information about data optimization including utilization trend reports, see Data Optimization.

Viewing Cluster Timeline

Use the cluster timeline to see a graphical depiction of operational and configuration events over time.

1. Browse to menu: **Compute > Infrastructure > Clusters**.
2. Click the cluster to view the timeline.
3. Click (**Monitoring**), and then (**Timelines**) or from the accordion menu, click menu: **Properties > Timelines**.
4. From **Options**, customize the period of time to display, and the types of events to see.



   - Use the **Interval** dropdown to select hourly or daily data points.
   - Use **Date** to type the date for the timeline to display.
   - If you select to view a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days.
   - The three **Event Group** dropdowns allow the selection of different groups of events to display. Each has its own color.
   - From the **Level** dropdown, select a **Summary** event if needed, or a **Detail** list of events. For example, the detail level of a *Power On* event might include the power on request, the starting event, and the actual **Power On** event. If you select **Summary**, the timeline only displays the Power On event.
5. To see more detail on an item in the timeline, click on it. A balloon appears with a clickable link to the resource.

Detecting Drift on Clusters

Over time, a cluster's configuration might change. Drift is the comparison of a cluster to itself at different points in time. The cluster requires analysis at least twice to collect information. Detecting drift provides users with the following benefits:

- See the difference between the last known state of a cluster and its current state
- Review the configuration changes that happen to a particular cluster between multiple points in time.
- Capture the configuration drifts for a single cluster across a time period.

Detect drift on clusters:

1. Browse to menu: **Compute > Infrastructure > Clusters**.
2. Click on the cluster to view drift.
3. Click **Relationships** in the cluster accordion.
4. Click **Drift History**.
5. Check the analyses to compare.
6. Click (**Drift Analysis**) at the top of the screen. The results are displayed.
7. Check the **Comparison** sections on the left to view in your comparison.
8. Click the plus sign next to the section name to expand it.
   - An item displayed on red text shows a change from the base analysis. An item displayed in black text shows no change from the base analysis.
   - A (**Changed from previous**) shows there has been a change since the last analysis.
   - A (**Same as previous**) means there has been no change since the last analysis.
   - Click (**Remove from drift**) at the bottom of a column to remove a specific analysis. The drift is then recalculated and the new results display.

9. Click (**Expanded View**) to see the expanded view. Click (**Compressed View**)] to compress the information.
10. Click the minus sign next to the section name to collapse it.
11. To limit the type of views, there are three buttons in the taskbar.
    - Click (**All attributes**) to see all attributes of the sections selected.
    - Click (**Attributes with different values**) to see only the attributes different across drifts.
    - Click (**Attributes with the same values**) to see only the attributes the same across drifts.

The drift displays for your cluster. Download the data or create a report from the drift for analysis using external tools.

Creating a Drift Report for Clusters

Use the drift report feature to export information about your cluster's drift.

1. Create a drift of a cluster.
2. Click (**Download**).
3. Click the output button for the type of report you want.
    - Click (**Download drift report in TXT format**) for a text file.
    - Click (**Download drift report in CSV format**) for a comma-separated file.
    - Click (**Download drift report in PDF format**) for a PDF file.

Removing Clusters

If a cluster has been decommissioned or requires troubleshooting, it might require removal from the VMDB.

1. Browse to menu: **Compute > Infrastructure > Clusters**.
2. Check the clusters to remove.
3. Click **Configuration**, and then (**Remove Clusters from the VMDB**).
4. Click **OK**.

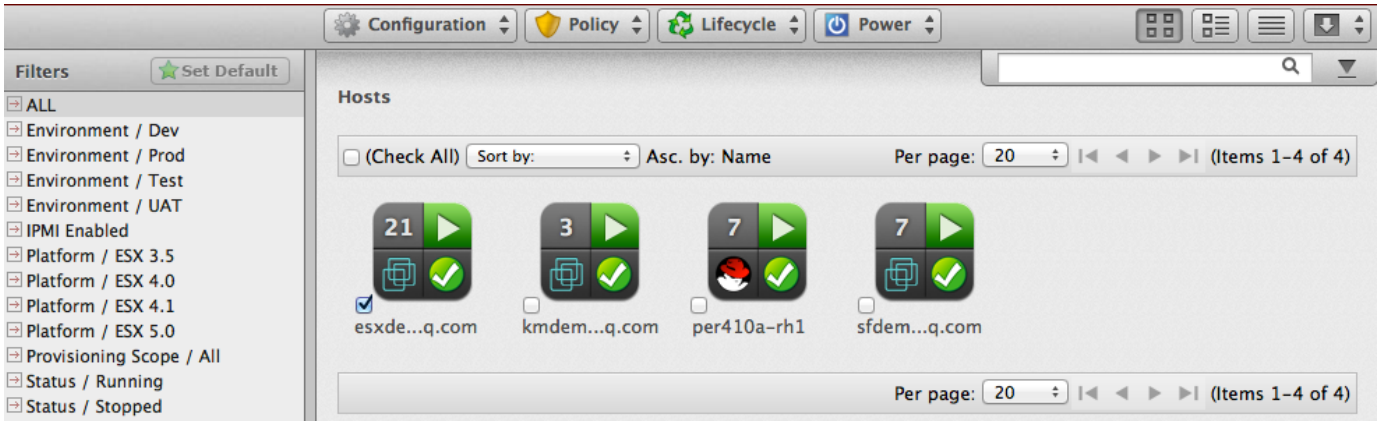
The clusters are deleted. Any virtual machines or hosts associated with these clusters remain, but are no longer associated with them.

Hosts

The **Hosts** page under **Compute > Infrastructure** displays the hosts discovered in your enterprise environment.

Note:

Any applied filters will be in effect here.



After adding or sorting your hosts, click on one to examine it more closely and see its virtual machines, SmartProxy settings, and properties.



1. Top left quadrant: Number of virtual machines on this host
2. Bottom left quadrant: Virtual machine software
3. Top right quadrant: Power state of host
4. Bottom right quadrant: Authentication status

Icon	Description
	Validated: Valid authentication credentials have been added.
	Invalid: Authentication credentials are invalid
	Unknown: Authentication status is unknown or no credentials have been entered.

Filtering Hosts

The Host Filter accordion is provided to easily navigate through the hosts. Use the ones provided or create your own. In addition, you can set a default filter.

Setting a Default Host Filter

Set the default filter for viewing your hosts.

1. From the **Filters** accordion on the left, click on the filter to use.
2. Click **Set Default** at the top of the filters list.

The default filter is set and marked by a green star next to its name.

Creating a Host Filter

Create a filter for viewing your hosts.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click (**Advanced Search**) to open the expression editor.
3. Use the expression editor to choose the appropriate options for your criteria.
4. Click **Save**.
5. Type in a name for the search expression in **Save this search as**.

   Note:

   This title depends on the type of resource you are searching.
6. Click **Save**.

The filter is saved and displays in the **My Filters** area of the **Filter** accordion.

Performing SmartState Analysis on Hosts

Perform a SmartState analysis on a host to collect additional information about it, such as patches, CPU, and memory.

Note:

- SmartState analysis on hosts is processed by the **Provider Operations** role. It is enabled by default.
- For ESX or ESXi hypervisors, consider the following: ESX hosts utilize a service console for host management and can be accessed using SSH. ESXi hosts lack a service console and therefore SSH cannot be used to obtain information sets for patches, services, Linux packages, user groups, SSH Config, and FS Files.
- `root` or administrator credentials are required to get patch information.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Check the hosts to analyze.
3. Click **Configuration**, and then (**Perform SmartState Analysis**).
4. Click **OK**.

Comparing Hosts

Infrastructure Automation allows you to compare hosts and check operating systems, host software and version information, and hardware.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Check the hosts to compare.
3. Click **Configuration**, and then (**Compare selected Hosts**). The comparison displays in a default expanded view, which lists a limited set of properties.
4. To remove a host from the comparison, click (**Remove this Host from the comparison**) at the bottom of the column.
5. To go to a compressed view, click (**Compressed View**). To return to an expanded view, click (**Expanded View**).
6. To limit the mode of the view, there are two buttons in the taskbar.
   - Click (**Details Mode**) to see all details for an attribute.
   - Click (**Exists Mode**) to limit the view to if an attribute exists compared to the base or not. This only applies to attributes that can have a Boolean property. For example, a user account exists or does not exist, or a piece of hardware that does or does not exist.
7. To change the base host that compare to the other hosts, click its label at the top of its column.
8. To go to the summary screen for a host, click its virtual thumbnail or icon.

Host Comparison Sections

Section	Description
Host Properties	Use this section to see basic information of the host, such as hostname, product, build number, hardware, and network adapters.
Security	Use this to see users and groups for the host, and firewall rules.
Configuration	Use this to see the operating system, applications, services, patches, vSwitches, vLANS, and advanced settings.
My Company Tags	Use this to see all tags.

Using the Host Comparison Sections

The following procedure describes how to use the host comparison sections.

1. On the left of a comparison screen, select the categories of properties to display.
2. Click the plus sign next to the sections name to expand it.
3. The following descriptions pertain to the **Expanded View** . Either the value of a property or an icon representing the property displays depending on the properties type.
   - A property displayed in the same color as the base means that the compared host matches the base for that property.
   - A property displayed in a different color from the base means that the compared host does not match the base for that property.
4. If you are in the **Compressed View** , the values of the properties do not display. All items are described by the icons shown below.
   - A (**checkmark**) means the compared host matches the base for that property. Hover over it and the value of the property displays.
   - A (**x**) means the compared host does not match the base for that property. Hover over it and the value of the property displays.
5. Click the plus sign next to the section name to collapse it.

This comparison is viewable in multiple ways. Export the data or create a report from your comparison for analysis using external tools.

Creating a Host Comparison Report

Create a quick report to compare clusters in CSV, TXT, or PDF formats.

1. Create the comparison to analyze.
2. Click (**Download**).
3. Click the output button for the type of report.
   - Click (**Download comparison report in TXT format**) for a text file.
   - Click (**Download comparison report in CSV format**) for a comma-separated file.
   - Click (**Download comparison report in PDF format**) for a PDF file.

Refreshing Multiple Hosts

Manually refresh a host for its properties and related infrastructure components.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Check the hosts to refresh.
3. Click **Configuration**, and then (**Refresh Relationships and Power States**).
4. Click **OK**.

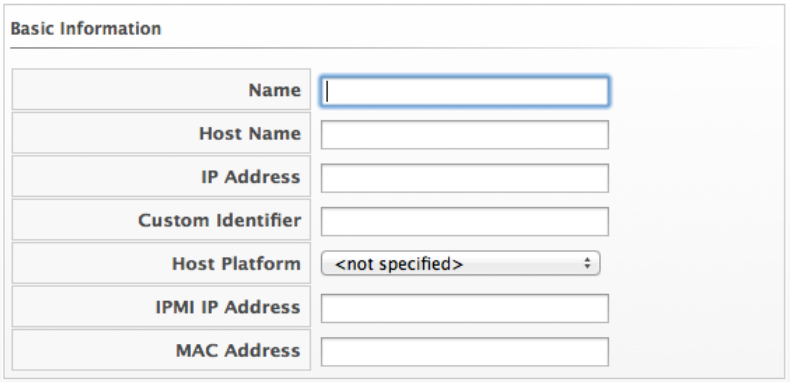
When a host is refreshed and a new virtual machine is discovered on that host, Infrastructure Automation checks to see if the virtual machine is already registered with another host. If this is the case, the host that the virtual machine is associated with switches to the new host. If the SmartProxy is monitoring a provider, this happens automatically. If not, the next refresh of the host addresses this.

Adding a Single Host

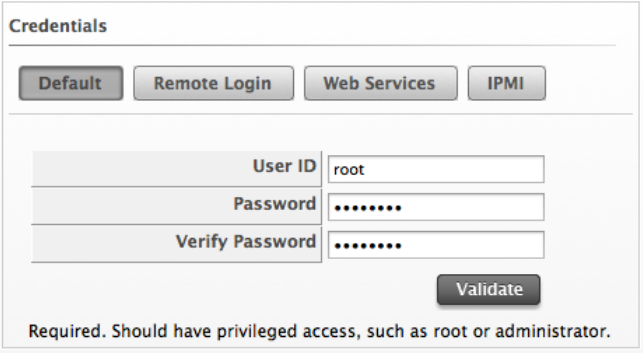
To analyze a host for more detailed information, add it to the VMDB first. If the host has not been found during **Host Discovery** or **Provider Refresh**, and the host's IP address is known, use the **Add a New Host** button.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click **Configuration**, then click (../images/1862.png)**Add a New item**.
3. Type the **Name**, **Host Name**, and **IP Address** of the host to add. **Name** is how the device is labeled in the console. Select the type of operating system from the **Host Platform** dropdown. If the Host has been found during **Discovery** or **Refresh** and the host's operating system has been identified, the **Host Platform** selector remains disabled. If adding an IPMI server for provisioning, add in the IP address of that host.

   Important: The **Host Name** must use a unique fully qualified domain name.

4. In the **Credentials** box, the **Default** tab provides fields to type a user name with elevated security credentials and the user's password. If using domain credentials, the format for **User ID** is in the format of `[domainname]\[username]`. On ESX hosts, if the `SSH` login is disabled for the **Default** user, type in a user with remote login access on the **Remote Login** tab.

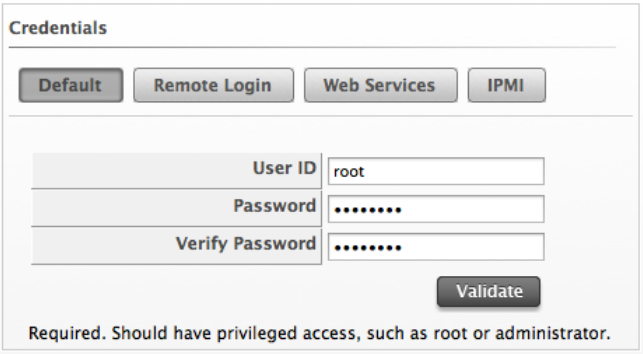


5. Click **Validate** to check the credentials.
6. Click **Save**.

Editing Hosts

If multiple hosts have the same settings or credentials, edit them at the same time.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click **Configuration**.
3. Check the Hosts to edit.
4. Click (**Edit Selected items**).
5. Use **Credentials** to provide login credentials required for this host.



- On the **Default** tab, type a user name with elevated security credentials and the users password. If you are using domain credentials, the format for User ID must be in the format of `[domainname]\[username]`.
- On `ESX` hosts, if `SSH` login is disabled for the **Default** user, type in a user with remote login access on the **Remote Login** tab. If this is not supplied, *Default* credentials will be used.
- Use **Web Services** to supply credentials for any web service calls made directly to the host system. If this is not supplied, **Default** credentials are used.

  Note:

  Login credentials are required for performing SmartState Analysis on the host's virtual machines and templates.

  For each type of credential used, the following information is required:

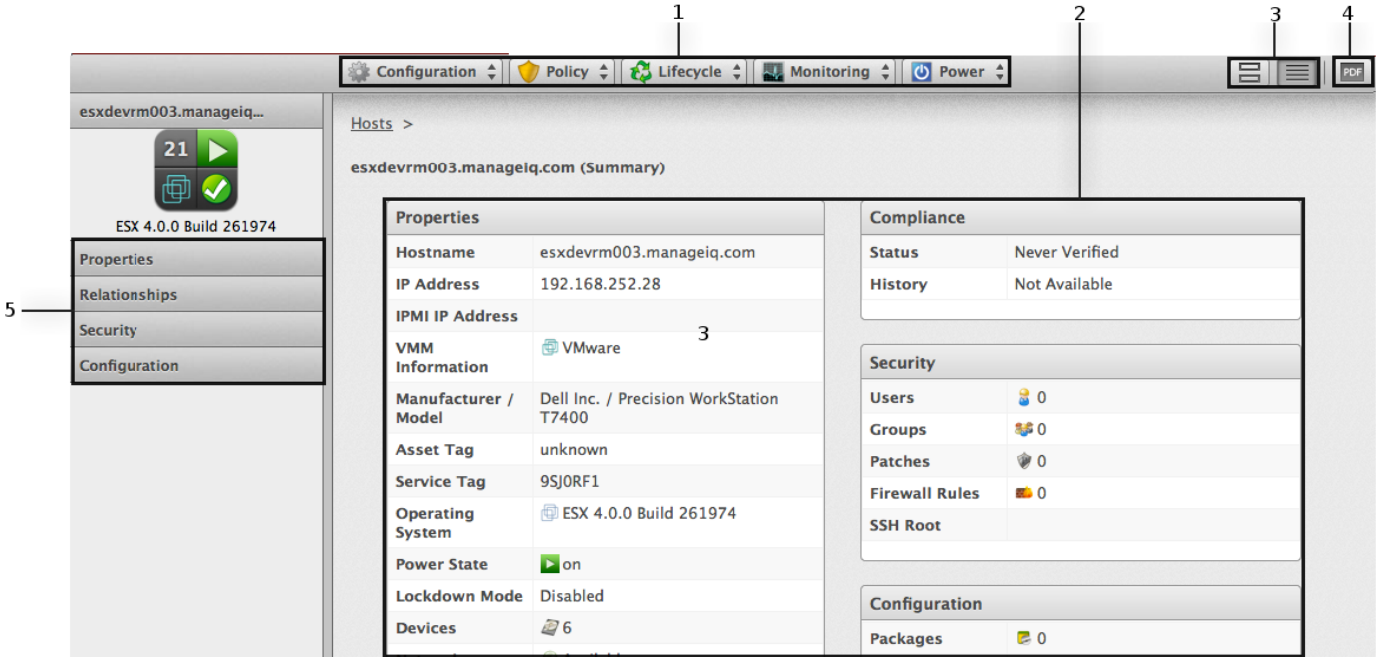
- Use **User ID** to specify a login ID.
- Use **Password** to specify the password for the User ID.
- Use **Verify Password** to confirm the password.

6. Test the credentials by using the *Select Host to validate against* drop down and click **Validate**.
7. Click **Save**.

Viewing a Host

You can click on a specific host to review it. The screen shows a host virtual thumbnail, a host taskbar, a host accordion, and a host summary.

Host Management Screen.

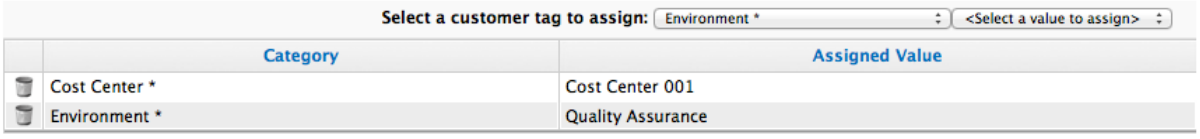


1. **Host Taskbar**: Use the host taskbar to take actions on the selected host
2. **Host Summary**: Use the host summary to see the properties of a host, drill down to a host's information, and view its installed virtual machines
3. **Host Summary Views**: Choose between graphical or text view of the provider summary
4. **Host PDF**: Generates host summary in PDF format
5. **Host Accordion**: See details about **Properties**, **Relationships**, **Security** and **Configuration** for the selected host

Tagging Multiple Hosts

To categorize hosts together, apply tags to multiple hosts at the same time.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Check the hosts to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a customer tag from the first dropdown, and then a value for the tag.



* Only a single value can be assigned from these categories

5. Select more tags or click **Save** to save your changes.

Removing Hosts

If a host is decommissioned or requires troubleshooting, it might require removal from the VMDB.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Check the hosts to remove.
3. Click **Configuration**, and then (**Remove items from the VMDB**).
4. Click **OK**.

The hosts are removed. The virtual machines remain in the VMDB, but are no longer associated with their respective hosts.

Scaling Down Compute Hosts

Through Infrastructure Automation, you can perform a *Compute scale down* on a Red Hat OpenStack infrastructure provider. This process involves decreasing its Compute nodes used by an OpenStack infrastructure provider. Doing so involves putting a Compute node into *maintenance mode* and removing it from the provider afterwards. Once a node is in maintenance mode, it can be repurposed (for examle, re-provision it as a Controller node), repaired, or decommissioned altogether.

Before scaling down, evacuate or migrate any instances hosted on the node you are removing. For instructions on either procedure, see Migrating a Live Instance or Evacuating an Instance.

After migrating or evacuating instances from the node, set the node to maintenance mode. To do so:

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click the OpenStack compute node to be removed from the provider.
3. Click **Configuration**, and then (**Toggle Maintenance Mode**).

   Note:

   This option can only be used with OpenStack providers with at least two Compute nodes.

Repeat this procedure for every node you want to remove from the cloud provider.

After setting a Compute node to maintenance mode, you can scale down its provider:

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click the provider to be scaled down.
3. Click **Configuration**, and then (**Scale Down**).
4. From the **Scale Infrastructure Provider Down** section, check the nodes to be removed from the provider. You can only do this for nodes where **Maintenance** is set to **true**.
5. Click **Scale Down**.

Refreshing Relationships and Power States for a Host

Refresh the relationships and power states of the items associated with your hosts from the Host Taskbar.

Note:

`root` or administrator credentials are required to get patch information.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click on the host to refresh.
3. Click **Configuration**, and then (**Refresh Relationships and Power States**) on the Host Taskbar.

Infrastructure Automation determines the state (running, stopped, or paused) of all virtual machines registered to the host.

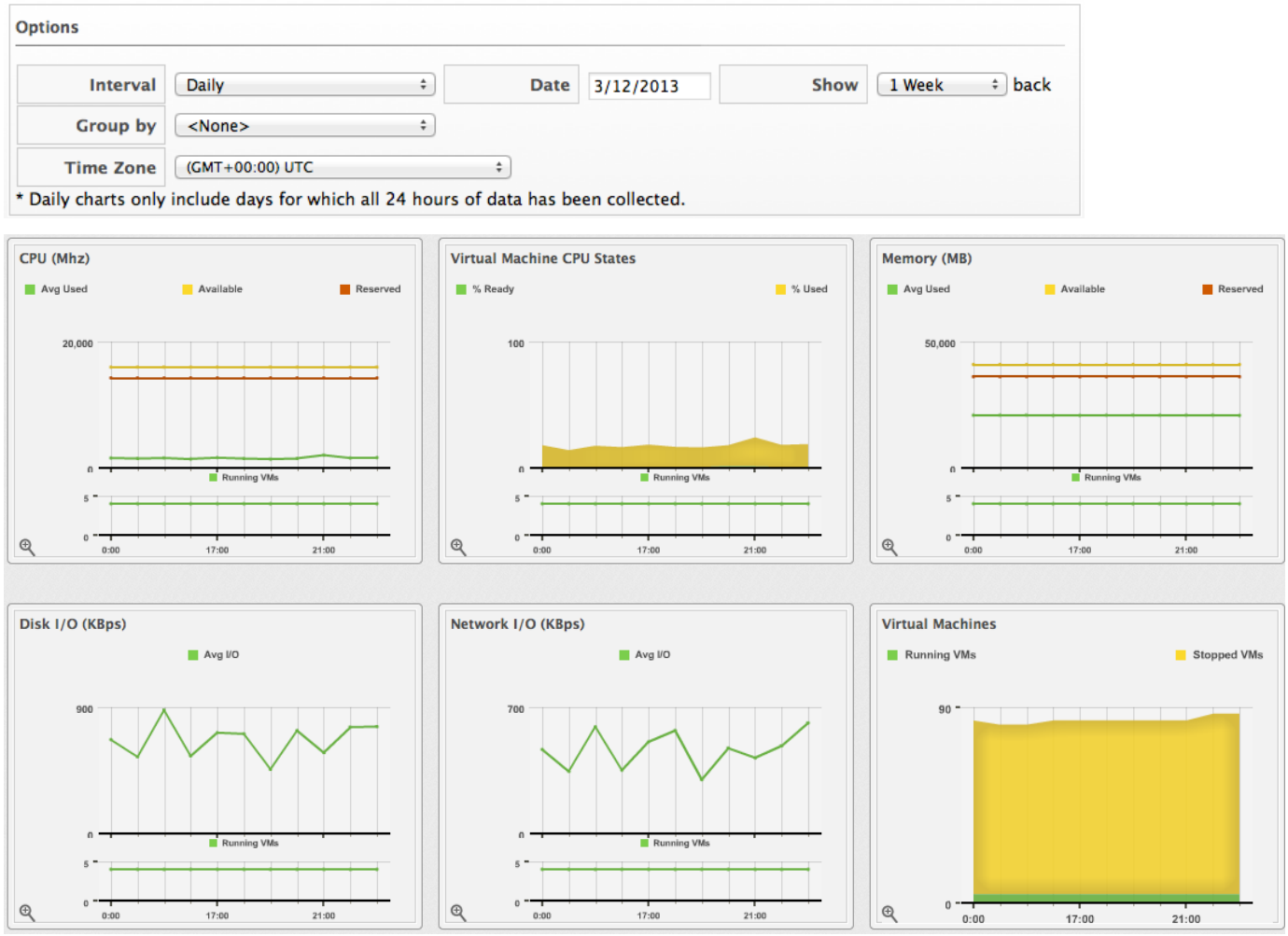
Viewing Capacity and Utilization Charts for a Host

View Capacity & Utilization data for hosts that are part of a cluster.

Note:

Your Infrastructure Automation server requires network visibility to the provider assigned the **Server Role** of **Capacity & Utilization Collector** to enable this feature.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click the Host to view capacity data.
3. Click (**Monitoring**), and then (**Utilization**) or from the Host accordion, click **Properties > Capacity & Utilization**.
4. From **Interval**, select to view hourly or daily data points and the dates to view data. Use **Group by** to group the lines by SmartTags. Use **Time Profiles** to select a time range for the data.



The charts are displayed for CPU, memory, disk, network, and running virtual machines.

Note:

Daily charts only include full days of data. If a day does not include all the 24 data points for a day, the data does not show for that day.

For information about data optimization including utilization trend reports, see Data Optimization.

Viewing the Host Timeline

View the timeline of events for the virtual machines registered to a host.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click the host to view the timeline.
3. Click (**Monitoring**), and then (**Timelines**) or from the host accordion, click **Properties > Timelines**.
4. From **Options**, customize the period of time to display and the types of events to see.
   - Use **Show** to select types of events to show on the timeline.
   - Use the **Interval** dropdown to select hourly or daily data points.
   - Use **Date** to type the date the timeline displays.
   - If you select to view a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days. If selecting **Hourly**, select the interval to see.
   - From the **Level** dropdown, select either a **Summary** event or a **Detail** list of events. For example, the detail level of a **Power On** event might include the power on request, the starting event, and the actual **Power On** event. If you select **Summary**, only the Power On event appears in the timeline.
   - The three **Event Group** dropdowns allow selection of different groups of events to display. Each group has its own color.
5. To see more detail on an item in the timeline, click on it. A balloon appears with a clickable link to the resource.

Host Virtual Summary

Clicking on a specific host shows the Host's Virtual Thumbnail and an **`operating system-sensitive`** screen of host information, called the Host Summary. Where applicable, click on a subcategory of the Host Summary to see more detail on that section.

A **Refresh** provides some basic information on the Host. To get more detail, enter credentials for the host and perform a SmartState Analysis.

The Summary divides into the following categories.

- **Properties** include information such as base operating system, hostname, IP addresses, devices attached to the system, and storage adapters. Some categories can be clicked on for additional detail. For example, click **Network** to view the network adapters connected to the host.

Properties	
Hostname	per410a-t5.manageiq.com
IP Address	192.168.252.4
IPMI IP Address	
VMM Information	VMware
Manufacturer / Model	Dell Inc. / PowerEdge R410
Asset Tag	unknown
Service Tag	BBR8TK1
Operating System	ESXi 4.0.0 Build 504850
Power State	on
Lockdown Mode	Disabled
Devices	6
Network	Available
Storage Adapters	5
Number of CPUs	2
Number of CPU Cores	8
CPU Cores Per Socket	4
Memory	40 GB
Manangement Engine GUID	a55d2e44-8828-11e2-9abc-005056af00b4

- **Relationships** include information on the provider, cluster, datastores, resource pools, and installed virtual machines.

Relationships	
Management System	Virtual Center (192.168.252.14)
Cluster	None
Datastores	13
Resource Pools	0
VMs	80
VM Templates	7
Drift History	2

- **Security** shows the number of users, groups, patches installed, and firewall rules on the host. Click on any of these items to see further details.

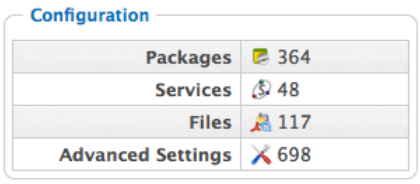
  Note:

  Run a SmartState Analysis on the host to retrieve this information.

- **Storage Relationships** shows the relationship the host has to LUNs, volumes, and file shares. The **Storage Inventory Role** must be enabled in the zone for these items to be populated.
- **Configuration** shows the number of packages and services installed. Click on any of these items to see more details.

  Note:

  Run a SmartState Analysis on the host to retrieve this information.



- **Smart Management** shows all tags assigned to this host.
- **Authentication Status** shows all the types of credentials entered for this host and the whether those credentials are valid.

Viewing Host Device Information

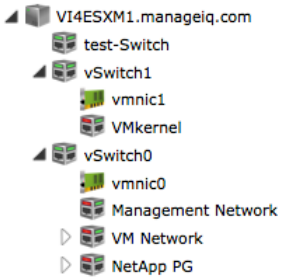
Access information on the hardware devices including processor, CPU type and speed, and memory for each host.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click the host to view the network information.
3. From the host accordion, click menu: **Properties > Devices**.

Viewing Host Network Information

Access information on networking including switches, network interfaces, and local area networks for each host.

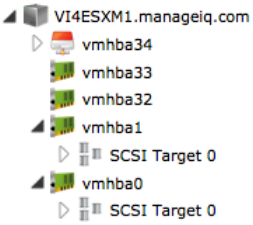
1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click the host to view the network information.
3. From the host accordion, click menu: **Properties > Network**.



Viewing Storage Adapters

Access information on the storage adapters including storage type for each host.

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click the host to view the network information.
3. From the host accordion, click menu: **Properties > Storage Adapters**.



Detecting Drift on Hosts

Over time, the configuration of a host might change. Drift is the comparison of a host to itself at different points in time. The host requires analysis at least twice to collect information. Detecting drift provides you the following benefits:

- See the difference between the last known state of a host and its current state.
- Review the configuration changes that happen to a particular host between multiple points in time.
- Capture the configuration drifts for a single host across a time period.

Detect drift on hosts:

1. Browse to menu: **Compute > Infrastructure > Hosts**.
2. Click on the host to view drift.
3. Click **Relationships** in the host accordion.
4. Click **Drift History**.
5. Check the analyses to compare.
6. Click  (**Drift**) at the top of the screen. The results display.
7. Check the **Comparison** sections on the left to view in your comparison.
8. Click **Apply**.
9. Click the plus sign next to the sections name to expand it.
    - An item displayed on red text shows a change from the base analysis. An item displayed in black text shows no change from the base analysis.
    - A  (**Changed from previous**) shows a change since the last analysis.
    - A  (**Same as previous**) means no change since the last analysis.
    - Click  (**Remove from drift**) at the bottom of a column to remove a specific analysis. The drift recalculates and the new results display.
10. Click  (**Expanded View**) to see the expanded view. Click  (**Compressed View**) to compress the information.
11. Click the minus sign next to the sections name to collapse it.
12. To limit the type of views, you have three buttons in the taskbar:
    - Click  (**All attributes**) to see all attributes of the sections you selected.
    - Click  (**Attributes with different values**) to see only the attributes that are different across the drifts.
    - Click  (**Attributes with the same values**) to see only the attributes that are the same across drifts.

The drift comparison displays. Download the data or create a report from your drift for analysis using external tools.

Creating a Drift Report for Hosts

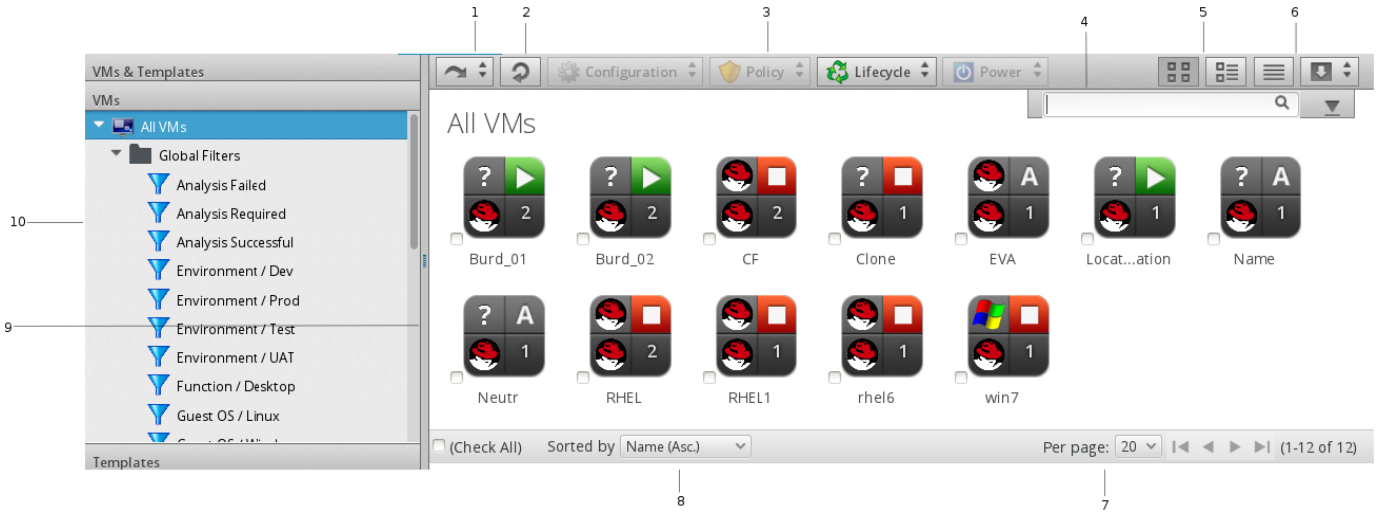
Use the drift report feature to export information about your host's drift.

1. Create the comparison to analyze.
2. Click  (**Download**).
3. Click the output button for the type of report you want.
    - Click  (**Download drift report in TXT format**) for a text file.
    - Click  (**Download drift report in CSV format**) for a comma-separated file.
    - Click  (**Download drift report in PDF format**) for a PDF file.

Virtual Machines

The heterogeneous virtual machine container and guest support combined with the ability to analyze information inside the virtual machine - such as disk space, patch level or installed applications - provides in-depth information across the virtual environment. This rich set of information enables Infrastructure Automation users to improve problem resolution times and effectively manage virtual machines.

The **Virtual Machines** pages display all virtual machines that were discovered by your server. Note that if you have applied a filter to a user, it will be in effect here. The **Virtual Machines** taskbar is a menu driven set of buttons that provide access to functions related to virtual machines.



1. History button
2. Refresh screen button
3. Taskbar
4. Name search bar/Advanced Search button
5. View buttons
6. Download buttons
7. Navigation bar
8. Sort dropdown
9. Main area in Grid View
10. Provider/Filter Navigation

The console uses **Virtual Thumbnails** to describe virtual machines and templates. Each thumbnail contains four quadrants by default. This allows you to glance at a virtual machine for a quick view of its contents.



1. Top left quadrant: Operating system of the Virtual Machine
2. Bottom left quadrant: Virtual Machine Hosts software
3. Top right quadrant: Power state of Virtual Machine or Status icon
4. Bottom right quadrant: Number of Snapshots for this Virtual Machine

Icon	Description
	Template: Virtual Template
	Retired: When a virtual machine or instance is no longer required, it can be retired. Once a virtual machine or instance reaches its retirement date and time, it is immediately shut down and not allowed to restart. If an attempt to restart is made, Infrastructure Automation will shut down the virtual machine or instance.
	Archived: An archived virtual machine has no host or datastore associated with it. Archiving is done to move virtual machines to a low cost storage, either on demand or during retirement, if requested, to avoid incurring extra cost on a virtualized infrastructure due to virtual machine sprawl.
	Orphaned: An orphaned virtual machine has no host but has a datastore associated with it. Orphaned virtual machines are those that have been removed from their providers but still exist on the storage. An orphaned virtual machine is unable to identify the associated host. A virtual machine also shows as orphaned if it exists on a different host than the host expected by the provider's server.
	Disconnected: A disconnected virtual machine is one that has lost connection to either the provider's storage, host, or both. A disconnect is usually a result of network issues on the provider side. For instance, if during virtual machine provisioning the storage is not set up or deleted, the virtual machine will still exist on the provider, but will not run on the host as it has lost connection to its provider's storage.
	On: Virtual Machine is powered on.
	Off: Virtual Machine is powered off.
	Suspended: Virtual Machine has been suspended.

The **Virtual Machines** page has three accordions organizing your virtual machines and templates in different ways. All of these accordions share a set of common controls:

- Use **VMs and Templates** to view your virtual machines and templates organized by Provider. In addition, you can see archived and orphaned items here.
- Use **VMs** to view, apply filters, and collect information about all of your virtual machines.
- Use **Templates** to view, apply filters, and collect information about all of your templates.

Through the console, you are able to view your virtual machines in multiple ways. For your virtual machines, you can:

- Filter virtual machines
- Change views
- Sort
- Create a report
- Search by MyTags
- Search by collected data

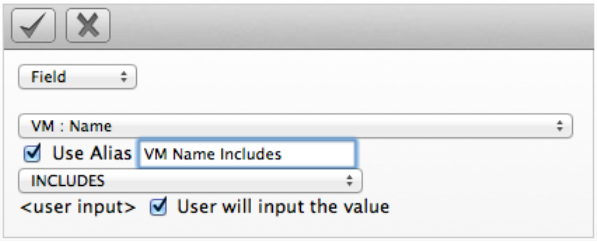
Filtering Virtual Machines and Templates

The **Virtual Machine Filter** accordion is provided so that you can easily navigate through groups of virtual machines. You can use the ones provided or create your own through **Advanced Filtering** capabilities.

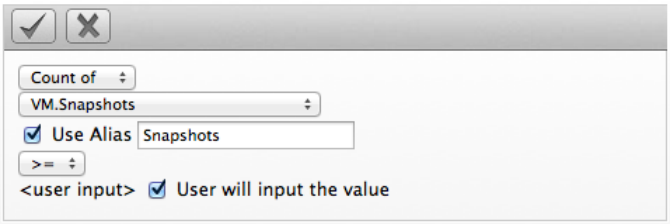
1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Go to the **VMs** or **Templates** accordion.
3. Click on the desired filter from the left pane.

Creating a Virtual Machine or Template Filter

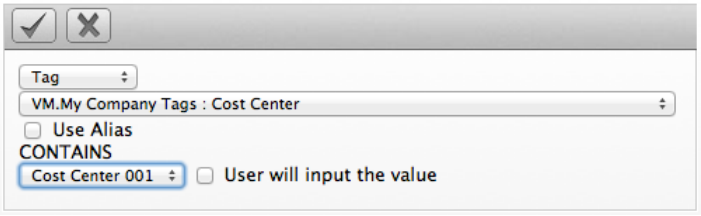
1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Go to the **VMs** or **Templates** accordion.
3. Click **All VMs** or **All Templates**, then click (**Advanced Search**) to open the expression editor.
4. Use the expression editor to choose the appropriate options for your criteria. Based on what you choose, different options will show.
   - For all of the types of searches, you have the options of creating an alias and requested user input. Select **Use Alias** to create a user friendly name for the search. If you are requested user input for the search, this text will show in the dialog box where the input is requested.
   - Click **Field** to create criteria based on field values.



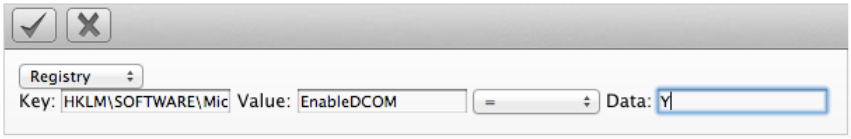
   - Click **Count of** to create criteria based on the count of something, such as the number of snapshots for a virtual machine, or the number of virtual machines on a host.



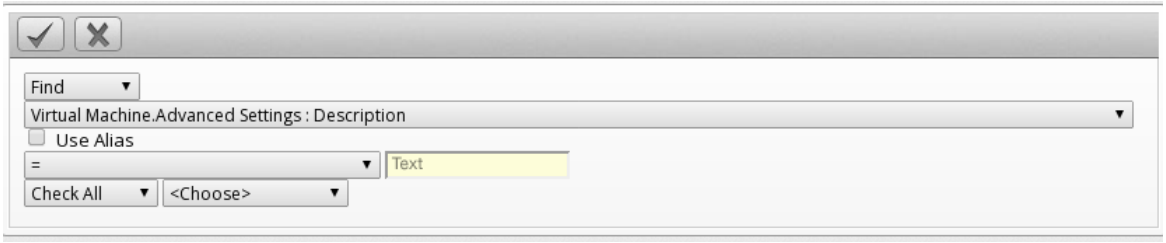
   - Click **Tag** to create criteria based on tags assigned to your virtual infrastructure, such as for power states or production tagging.



- Click **Registry** to create criteria based on registry values, such as the DCOM status of a Windows system. Note this criteria applies only to Windows operating systems.



- Click **Find** to seek a particular value, and then check a property.



5. Click ✓ (**Commit Expression Element Changes**) to add the expression.
6. Click **Save**.
7. Type in a name for the search expression in **Save this VM** search as. (Note that this title depends on the type of resource you are searching.) To set the filter to show globally, check **Global search**.
8. Click **Save**.

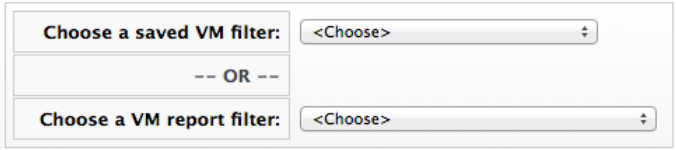
The filter is saved and will show in the **My Filters** area of the **Filter** accordion. If you checked **Global search**, the filter will show under **Global Filters**.

Loading a Report Filter or Search Expression

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to search either **VMs** or **Templates**.
3. Click » (**Advanced Search**) to open the expression editor.
4. Click **Load**.
5. Select either a saved virtual machine search or a virtual machine report filter.

   Note:

   The set of items to select will depend on the type of resource you are searching.



6. Click **Load** to load the search expression.
7. If you want to edit the expression, click on it and make any edits for the current expression.

   - Click ✓ (**Commit expression element changes**) to add the changes.
   - Click ↶ (**Undo the previous change**) to remove the change you just made.
   - Click ↷ (**Redo the previous change**) to put the change that you just made back.
   - Click AND (**AND with a new expression element**) to create a logical AND with a new expression element.

- Click OR (**OR with a new expression element**) to create a logical OR with a new expression element.
- Click NOT (**Wrap this expression element with a NOT**) to create a logical NOT on an expression element or to exclude all the items that match the expression.
- Click ✖ (**Remove this expression element**) to take out the current expression element.

8. Click **Load**.
9. Click **Apply**.

Changing Views for Virtual Machines and Templates

While you can set the default view for different pages from the settings menu, then **Configuration > My Settings > Default Views**, the current view can also be controlled from the Virtual Machines pages.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to view.
3. Click the appropriate button for the desired view.
   - Click for **Grid View**.
   - Click for **Tile View**.
   - Click for **List View**.

Sorting Virtual Machines and Templates

Virtual machines and templates can be sorted by Name, Cluster, Host, Datastore, Compliance, Last Analysis Time, Total Snapshots, or Region.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to sort.
3. To sort virtual machines or templates when in grid or tile view:
   - From the **Sort by** dropdown, click the attribute to sort.
4. To sort virtual machines or templates when in list view:
   - Select the **List View**.
   - Click on the **Column Name** to sort. For example, click on **Cluster** to sort by the name of the cluster.

Creating a Virtual Machine or Template Report

For a listing of virtual machines and templates, you can create a quick report in CSV, TXT, or PDF formats.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items for report creation.
3. Click (**Download**).
   - Click for a **TXT** file.
   - Click for a **CSV** file.
   - Click for a **PDF** file.

Searching for Virtual Machines or Templates

To the right of the taskbar on the *Virtual Machines* page, you can enter names or parts of names for searching. You can search in the following ways:

- Type characters that are included in the name. For example, if you type **`sp1`**, all Virtual Machines whose names include **`sp1`** appear, such as **`Windows2003sp1`** and **`Sp1clone`**.

- Use * at the end of a term to search for names that begin with specific characters. For example, type **v*** to find all virtual machines whose names begin with the letter **v**.
- Use ***** at the beginning of a term to search for names that end with specific characters. For example, type ***sp2** to find all virtual machines whose names end with **sp2**.
- Erase all characters from the search box to go back to viewing all virtual machines.

Search for virtual machines or templates:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to search.



3. In the **Name Filter** bar in the upper right corner of the window, type your criteria.
4. Click (**Search by Name within results**) or press **Enter**.
5. Type in other criteria to filter on what is currently displayed.
6. Click (**Search by Name within results**) or press **Enter**.

Analyzing Virtual Machines and Templates

Analyze a virtual machine to collect metadata such as user accounts, applications, software patches, and other internal information. If Infrastructure Automation is not set up for automatic analysis, perform a manual analysis of a virtual machine. To perform a SmartState analysis, Infrastructure Automation requires a running SmartProxy with visibility to the virtual machine's storage location. If the virtual machine is associated with a host or provider, ensure the virtual machine is registered with that system to be properly analyzed; the server requires this information since a snapshot might be created.

Note:

SmartState Analysis of a virtual machine requires access to its host. To perform a successful analysis, edit the virtual machine's host and enter the host's authentication credentials.

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to analyze.
3. Check the **Virtual Machines** and **Templates** to analyze.
4. Click (**Configuration**), then (**Perform SmartState Analysis**).
5. Click **OK**.

Red Hat Enterprise Virtualization Prerequisites

SmartState Analysis on Red Hat Enterprise Virtualization Manager 3.1 and Above - Storage Support Notes

Note the following requirements when performing SmartState Analysis on Red Hat Enterprise Virtualization Manager 3.1 and above.

- NFS
  - The Infrastructure Automation appliance requires a mount to the NFS datastore.
- iSCSI / FCP
  - For Red Hat Enterprise Virtualization 3.1 and 3.2, clusters must use full Red Hat Enterprise Linux hosts and not Red Hat Enterprise Virtualization Hypervisor hosts. You can use either type of host in Red Hat Enterprise Virtualization 3.3 and above.
  - Each Infrastructure Automation appliance performing SmartState Analysis requires sharable, non-bootable DirectLUN access to each attached iSCSI/FCP storage domain. In order to perform smart analysis, the appliance must mount the data storage as a DirectLUN disk.
  - A Infrastructure Automation appliance **must** reside in each datacenter with the iSCSI / FCP storage type.
  - If your storage domain is using Logical Volume Manager (LVM), you will need to manually activate your volume groups and logical volumes before performing the SmartState Analysis.
    1. Edit the **/etc/lvm/lvm.conf** file on the appliance and add volume groups to activate as read-only:

```
activation {
    read_only_volume_list = ["volume_group_0", "volume_group_1", etc...]
}
```

Note: the volume group names are the storage domain UUIDs

2. Activate volume groups:

   ```
   vgchange -ay
   ```

- Other Notes
  - The **Edit Management Engine Relationship** option enables the VM SmartState Analysis job to determine the datacenter where a Infrastructure Automation appliance is running and thus to identify what storage it has access to in a Red Hat Enterprise Virtualization environment.
    - After setting up a Infrastructure Automation appliance and performing a refresh of the provider, find the Infrastructure Automation appliance in the **Virtual Machine** accordion list and view its summary screen.
    - Click **Configuration > Edit Management Engine Relationship**.
    - Select the server that relates to this instance of the Infrastructure Automation appliance.

If you attach a DirectLUN disk after configuring the Infrastructure Automation database, access the appliance in a terminal and run `pvscan` to detect the DirectLUN disk. Alternatively, in Infrastructure Automation 5.2.1 and above, you can restart the appliance to detect the disk automatically.

SmartState Analysis on Red Hat Enterprise Virtualization Manager 3.0 - Storage Support Notes

There are two additional steps required to perform a SmartState Analysis on Red Hat Enterprise Virtualization Manager 3.0 using **`iSCSI`** or **`FCP`** storage. NFS storage does not have these requirements.

1. Enable **`DirectLUN`** support for the host and Infrastructure Automation appliance that performs the analysis.
   - Enable **`DirectLUN`** on host.
   - Enable **`DirectLUN`** on the Infrastructure Automation appliance. To do this, edit the desired Red Hat Enterprise Virtualization storage and get the **`LUNID`** value. Then, on the Infrastructure Automation appliance virtual machine in the Red Hat Enterprise Virtualization user interface, right-click and select **Edit > Custom Properties** and enter the following in the **Custom Properties** edit box:

     ```
     directlun=<LUN ID>:readonly
     ```

     If you have multiple storage domains, separate them by a comma, similar to:

     ```
     directlun=<LUN ID 1>:readonly,<LUN ID 2>:readonly,<LUN ID N>:readonly
     ```

     Note:

     The Infrastructure Automation appliance must reside in the same data center as the storage you are trying to connect. If you have multiple data centers with **`iSCSI`** or **`FCP`** storage, you need a Infrastructure Automation appliance in each data center to support virtual machine scanning.
2. **Set Server Relationship** - This is required to allow the virtual machine SmartState analysis job to determine which data center a Infrastructure Automation appliance is running and therefore identify what storage it has access to in a Red Hat Enterprise Virtualization environment.
   1. After setting up a Infrastructure Automation appliance and performing a refresh of the Provider, find the Infrastructure Automation appliance in the **Virtual Machine** accordion list and view its summary screen.
   2. Click  (**Configuration**), and then  (**Edit Server Relationship**).
   3. Select the server that relates to this instance of the Infrastructure Automation appliance.
3. If your storage domain is using Logical Volume Manager (LVM), you will need to manually activate your volume groups and logical volumes before performing the SmartState Analysis.
   1. Edit the **`/etc/lvm/lvm.conf`** file on the appliance and add volume groups to activate as read-only:

      ```
      activation {
          read_only_volume_list = ["volume_group_0", "volume_group_1", etc...]
      }
      ```

      Note: the volume group names are the storage domain UUIDs
   2. Activate volume groups:

      ```
      vgchange -ay
      ```

   For more information, please refer to the Red Hat Enterprise Linux documentation.

VMware vSphere Prerequisites

Comparing Virtual Machines and Templates

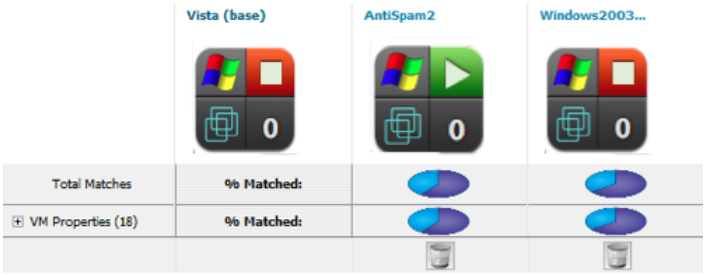
The Infrastructure Automation Server allows you to compare multiple virtual machines. This allows you to see how different virtual machines are from their original template. This helps detect missing patches, unmanaged user accounts, or unauthorized services.

Use the comparison feature to:

- Compare multiple virtual machines from different hosts.
- Compare multiple virtual machines side-by-side.
- Quickly see similarities and differences among multiple virtual machines and a base.
- Narrow the comparison display to categories of properties.
- Print or export in the comparison results to a PDF or CSV file.

Compare virtual machines and templates:

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to analyze.
3. Check the items to compare.
4. Click btn:[(Configuration)], and then btn:[(Compare Selected items)]. The comparison displays in a compressed view with a limited set of properties listed.



5. To delete an item from the comparison, click btn:[(Remove this item from Inventory)] at the bottom of the items column. This option is only available when comparing more than two virtual machines.
6. To view many items on one screen, go to a compressed view by clicking btn:[(Compressed View)]. To return to an expanded view, click btn:[(Expanded View)].
7. To limit the mode of the view, there are two buttons in the task bar.
   - Click btn:[(Details Mode)] to see all details for an attribute.
   - Click btn:[(Exists Mode)] to limit the view to if an attribute exists compared to the base or not. This only applies to attributes that can have a boolean property. For example, a user account exists or does not exist, or a piece of hardware that does or does not exist.
8. To change the base virtual machine that all the others are compared to, click its label at the top of its column.
9. To go to the summary screen for a virtual machine, click its btn:[Virtual Thumbnail] or icon.

Virtual Machine and Templates Comparison Sections

The following table describes the different sections for comparison information.

Section	Description
Properties	Use this section to see basic information on the file location of the virtual machine, its name, and the virtual machine monitor vendor. Hardware, disk, CD/DVD drives, floppy drive, network adapter, and volume information is also included.
Security	Use this to see users and groups for the virtual machine, including those which may be unauthorized compared to a template.
Configuration	Use this to see Guest Applications, Win32 services, Linux Init Processes, Kernel Drivers, File System Drivers, and Patches.
My Company Tags	Use this to see all tags.

Using the Virtual Machine Comparison Sections

Use the comparison sections to view various comparison data and display the data in different ways.

1. On the left of a comparison screen, select what categories of properties to display.
2. Click **Apply**.
3. Click the plus sign next to the sections name to expand it.

4. The following descriptions pertain to the **Expanded View** . Whether you see the value of a property or an icon representing the property depends on the properties type.
    - A property displayed in the same color as the base means that the compared virtual machine matches the base for that property.
    - A property displayed in a different color from the base means that the compared virtual machine does not match the base for that property.
5. If you are in the **Compressed View** , the values of the properties will not be displayed. The icons shown below will describe all items.
    - A (**checkmark**) means that the compared virtual machine matches the base for that property. If you hover over it, the value of the property will display.
    - A (**x**) means that the compared virtual machine does not match the base for that property. If you hover over it, the value of the property will display.
6. Click the minus sign next to the sections name to collapse it.

Your comparison can be viewed in multiple ways. Export the data or create a report from your comparison for analysis using external tools.

Creating a Virtual Machine Comparison Report

Output the data from a comparison report in TXT, CSV or PDF formats.

1. Create the comparison for the report.
2. Click the output button for the chosen report type.
    - Click (**Download comparison report in TXT format**) for a text file.
    - Click (**Download comparison report in CSV format**) for a CSV file.
    - Click (**Download comparison report in PDF format**) for a PDF file.

Controlling the Power State of Red Hat Virtualization Virtual Machines

Follow this procedure to control the power states of Red Hat Virtualization VMs through the Infrastructure Automation console.

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Click on a VM to change the power state.
3. Click **Power Operations**, then click the button for the desired power operation. Available operations will depend on the current power state of the VM.
    - Click (**Shutdown Guest**) to shutdown the guest OS on the VM.
    - Click (**Restart Guest**) to restart the guest OS on the VM.
    - Click (**Power On**) to power on the VM.
    - Click (**Power Off**) to power off the VM.
    - Click (**Suspend**) to suspend the VM.
    - Click (**Reset**) to reset the VM.
4. Click **OK**.

Refreshing Virtual Machines and Templates

Refresh your virtual machines to get the latest data the provider or host can access. This includes information such as the power state, container, and hardware devices attached to the virtual machine.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to analyze.
3. Check the items to refresh.
4. Click **Configuration**, and then (**Refresh Relationships and Power States**) on the **Virtual Machine Taskbar**.

The console returns a refreshed list of the data associated with the selected virtual machines.

Extracting Running Processes from Virtual Machines and Templates

Infrastructure Automation can collect processes running on Windows virtual machines. To do this, enter domain credentials for the zone where the virtual machine is located.

The virtual machine must be running and must have an IP address in the VMDB, usually obtained from a SmartState Analysis.

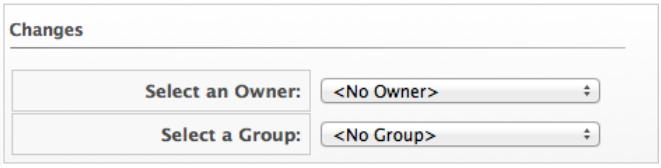
1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Check the Virtual Machines to collect the processes.
3. Click **Configuration**, and then (**Extract Running Processes**) on the Taskbar.
4. Click **OK**.

The server returns the running processes. View the summary of the virtual machine to see the details.

Setting Ownership for Virtual Machines and Templates

You can set the owner of a group of virtual machines and templates by either individual user or group. This allows you an additional way to filter and can be used to enforce quotas.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to change.
3. Check the items to set ownership.
4. Click **Configuration**, and then (**Set Ownership**) on the **Virtual Machine Taskbar**.
5. From the **Select an Owner** dropdown, select a user, and from the **Select a Group** dropdown, select a group.



6. Click **Save**.

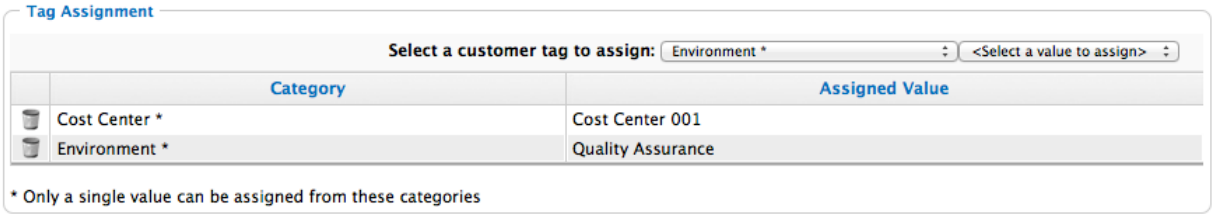
Removing Virtual Machines and Templates from the VMDB

If a virtual machine has been decommissioned or you need to perform some troubleshooting, you might need to remove a specific virtual machine from the VMDB. This does not however remove the virtual machine or template from its Datastore or Provider.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to remove.
3. Check the items to remove.
4. Click **Configuration**, and then (**Remove from the VMDB**) button.
5. Click **OK**.

Tagging Virtual Machines and Templates

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to tag.
3. Check the items to tag.
4. Click (**Policy**), and then (**Edit Tags**).
5. Select a customer tag from the first dropdown, and then a value for the tag.



Viewing a VMware Virtual Machine's Storage Profile

VMware storage profiles allow you to assign policies to datastores. Storage profiles are used to tag virtual machines to ensure they operate in compliance with settings in the datastore.

Infrastructure Automation retrieves VMware virtual machine storage profile information in the inventory, and associates the virtual machines and disks with them.

To view a virtual machine's storage profile:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click a VMware virtual machine to open its summary page.
3. The VMware **Storage Profile** is listed under **Properties**.

You can assign a storage profile when provisioning a VMware virtual machine in Infrastructure Automation, by using the virtual machine as a template to clone. See Provisioning Virtual Machines in *Provisioning Virtual Machines and Hosts* for instructions.

Viewing Running Processes after Collection

1. Click a virtual machine with collected processes.
2. From the **Diagnostics** area, click **Running Processes**.

Name	PID	Memory %	Memory Usage	CPU %	CPU Time
AppleMobileDeviceService.e	1,360	0.54	2,883,584	0.00	0
EarthAgent.exe	1,412	1.36	7,278,592	0.00	409
NSClient++.exe	1,524	1.70	9,093,120	0.00	176
SoftwareUpdate.exe	2,028	3.37	18,075,648	0.00	0
SpntSvc.exe	1,640	1.13	6,041,600	0.00	695
StOPP.exe	2,052	0.50	2,658,304	0.00	0
StWatchDog.exe	3,672	0.40	2,125,824	0.00	0
VMwareService.exe	1,664	14.34	76,918,784	0.00	108
VMwareTray.exe	2,368	0.14	749,568	0.00	0
VMwareUser.exe	3,276	0.13	696,320	0.00	2
csrss.exe	332	1.21	6,475,776	0.00	5
csrss.exe	3,304	1.24	6,647,808	0.00	3

The most recent collection of running processes is displayed. Sort this list by clicking on the column headers.

Editing Virtual Machine or Template Properties

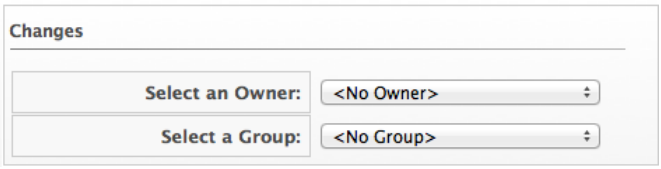
Edit the properties of a virtual machine or template to set parent and child virtual machines. SmartState Analysis also can detect this.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to edit.
3. Click the item to edit properties.
4. Click **Configuration**, and then (**Edit selected item**) on the Taskbar.
5. From the **Parent VM** dropdown, select the parent virtual machine.
6. From **Child VM** selection, select virtual machines that are based on the current virtual machine from the list of **Available VMs**.
7. Click **Save**.

Setting Ownership of a Virtual Machine or Template

Set the owner of a virtual machine or template by either individual user or group. This allows you an additional way to filter configuration items.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the items to analyze.
3. Click the item to set ownership.
4. Click **Configuration**, and then (**Set Ownership**) on the taskbar.
5. From the **Select an Owner** dropdown, select a user.



6. From the **Select a Group** dropdown, select a group.
7. Click **Save**.

Right Sizing a Virtual Machine

Infrastructure Automation uses collected statistics to recommend the best size for a virtual machine. Infrastructure Automation uses the information from the **Normal Operating Range** to calculate the recommendations.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click a virtual machine for right-sizing.
3. Click **Configuration**, and then (**Right-Size Recommendations**) button.

A new page appears with three levels of Memory and CPU recommendations, Conservative, Moderate, and Aggressive, next to the Normal Operating Range statistics.

Viewing Capacity and Utilization Charts for a Virtual Machine

You can view capacity and utilization data for virtual machines that are part of a cluster. Note that daily charts only include full days of data. If all 24 data points for a day are not available, daily charts are not displayed. For some capacity and utilization data, Infrastructure Automation calculates and shows trend lines in the charts which are created using linear regression. The calculation uses the capacity and utilization data collected by Infrastructure Automation during the interval you specify.

Note:

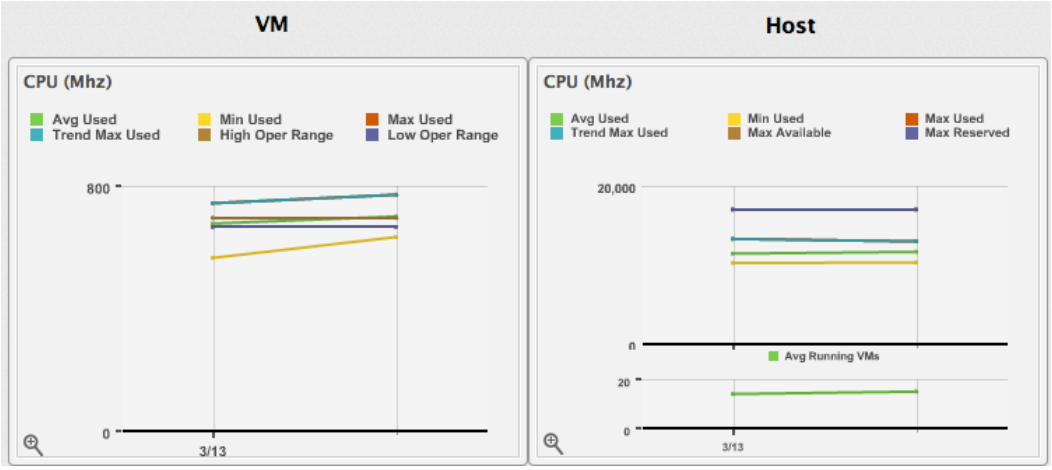
You must have a server with network visibility to your provider assigned the server role of **Capacity & Utilization Collector** to use this feature.

The virtual machine must be powered on to collect the data.

1. From **Compute > Infrastructure > Virtual Machines**, click the accordion that you want to view capacity data for.
2. Click the item you want to view.
3. Click (**Monitoring**), and then (**Utilization**).
4. From **Interval**, select to view **Daily**, **Hourly**, or **Most Recent Hour** data points. When choosing **Daily**, you can also select the **Date**, and how far back you want to go from that date. When selecting **Hourly**, you can select the date for which you want to view hourly data. If you are using **Time Profiles**, you will be able to select that as an option, also.



5. From **Compare to**, select **Parent Host** or **Parent Cluster**. The capacity and utilization charts for both items will show simultaneously.



Note:

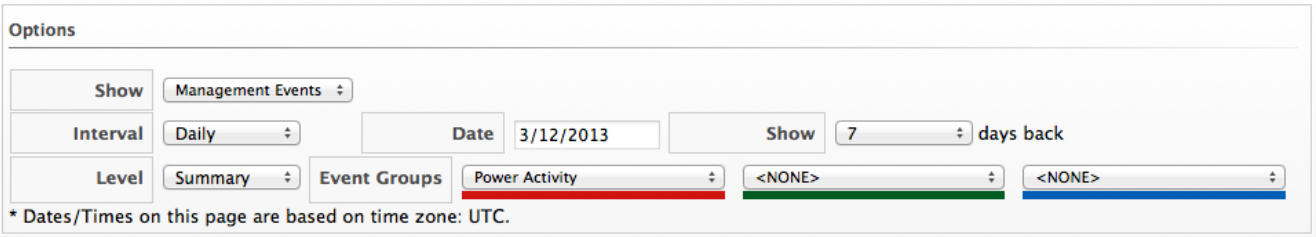
Daily charts only include full days of data. This means Infrastructure Automation does not show daily data for a day without a complete 24 data point range for a day.

For information about data optimization including utilization trend reports, see Data Optimization.

Viewing the Virtual Machine or Template Timeline

View the timeline of events for a virtual machine or template if registered to a Host.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the virtual machine to view the timeline.
3. Click (**Monitoring**), and then (**Timelines**) on the taskbar.
4. From **Options**, customize the period of time to display, and the types of events to view.



- Use the **Interval** dropdown to select hourly or daily data points.
- Use **Date** to type the date of the timeline to display.
- If viewing a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days.
- The three **Event Group** dropdowns allow selection of different event groups to display. Each has its own color.
- From the **Level** dropdown, select either a **Summary** event or a **Detail** list of events. For example, the detail level of a **Power On** event might include the power on request, the starting event, and the actual **Power On** event. If you select **Summary**, you only see the **Power On** event in the timeline.

5. To see more detail on an item in the timeline, click on it. A balloon appears with a clickable link to the resource.

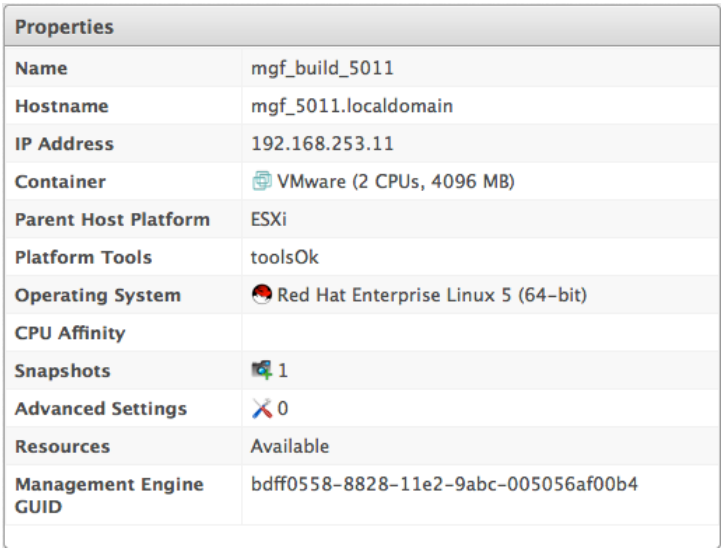
Virtual Machine or Template Summary

When you click on a specific virtual machine or template, you will see the **Virtual Thumbnail**, and an operating system-specific screen of the item, called the **Summary**. Where applicable, click on a subcategory of the **Summary** to see more detail on that section.

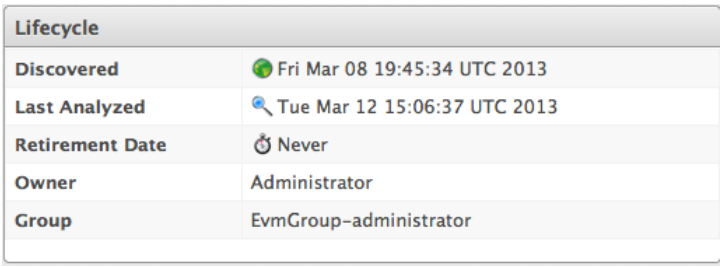
Note:

When you perform a SmartState Analysis on a virtual machine or template, you get more detailed information in these categories.

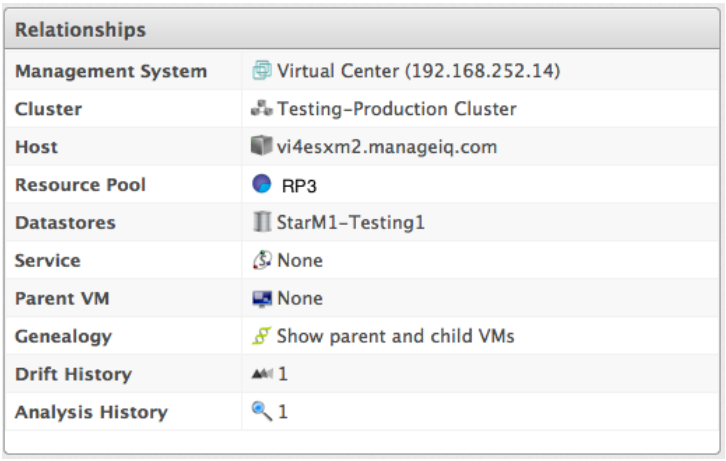
- **Properties** include information such as the base operating system, hostname, IP addresses, Virtual Machine vendor, CPU Affinity, devices attached to the system, and snapshots. This includes the ability to analyze multiple partitions, multiple disks, Linux logical volumes, extended partitions, and Windows drives. Some categories can be clicked on for additional detail. For example, click **Container** to view notes associated with a virtual machine.



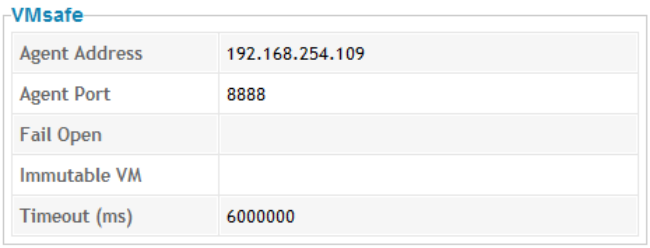
- **Lifecycle** shows the date of discovery and the last analysis. If a retirement date and time or owner has been set, these display as well.



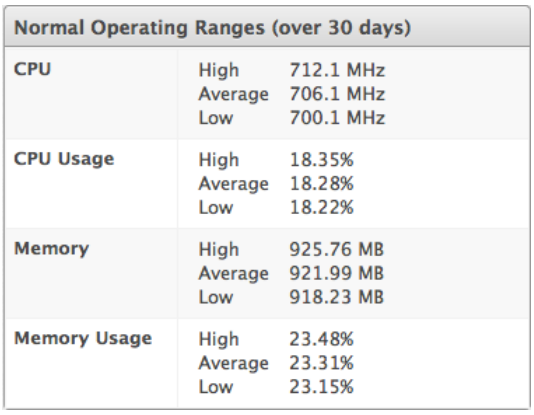
- **Relationships** include information on the parent host, genealogy such as parent and child virtual machines, and drift.



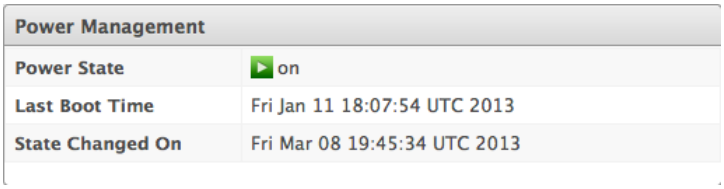
- **Storage Relationships** shows relationships to Filers, LUNs, Volumes and File Shares.
- **VMsafe** shows properties of the VMsafe agent if it is enabled.



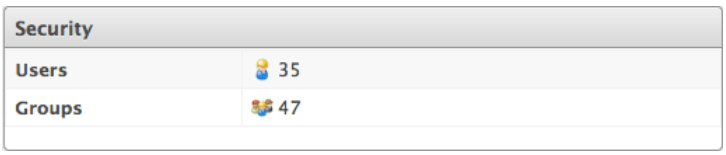
- **Normal Operating Ranges** shows the values for the normal operating range for this virtual machine. These statistics are used in calculating right sizing recommendations.



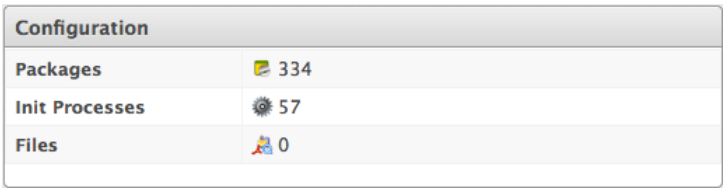
- **Power Management** displays the current power state, last boot time, and last power state change. **State Changed On** is the date that the virtual machine last changed its power state. This is a container view of the power state, therefore a restart of the operating system does not cause the container power state to change and will not update this value.



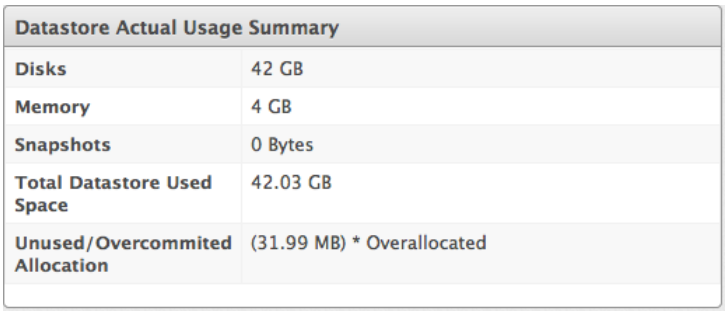
- **Security** includes information on users, groups, and security patches. Recall that the items shown on the **Summary** screen change based on the guest operating system.



- **Configuration** includes information on applications, services, packages, and init processes. This section changes depending on the base operating system.



- **Datastore Allocation Summary** shows how many and how much disk space has been allocated to this virtual machine as well as disk alignment and thin provisioning information.
- **Datastore Actual Usage Summary** shows how much disk and memory the virtual machine is actually using.



- **Diagnostics** provides a link to viewing running processes and the information from the latest collected event logs.
- **Smart Management** shows all tags assigned to this virtual machine.

Viewing the Operating System Properties

View details of the operating system from the **Virtual Machine Summary** or the accordion. For Windows systems, see **Account Policies** for the virtual machine.

1. From **Compute > Infrastructure > Virtual Machines**, click on the item to view its **Summary**.
2. From the **Properties** section, click **Operating System**.

An expanded view of the operating systems properties and **Account Policies** displays. This varies based on the operating system.

Viewing Virtual Machine or Template Snapshot Information

View the list of snapshots to see a history of their creation and size. Infrastructure Automation provides the description, size, and creation time of the snapshot as well as a view of the genealogy of the snapshots.

Note:

Snapshot size is only available after the successful completion of a **SmartState Analysis**.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the appropriate accordion containing the item you wish to view the snapshots of.
3. Click on the item to view its **Summary**.
4. From the **Summary**, click *Snapshots* in the **Properties** area.
5. The list of snapshots show in a tree format and captures their genealogy.

Viewing User Information for a Virtual Machine or Template

Infrastructure Automation's **SmartState Analysis** feature returns user information. Drill into the user to get details on the user's account, including group memberships.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view user information.
3. Click on the item to view its **Summary**.
4. From the **Security** section of the **Virtual Machine Summary**, click **Users**.
5. Click the user to view details.

Viewing Group Information for a Virtual Machine or Template

Infrastructure Automation's **SmartState Analysis** feature returns group information. Explore a group to get a list of its users.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view user information.
3. Click on the item to view its **Summary**.
4. From the **Security** section of the **Virtual Machine Summary**, click **Groups**.
5. Click the group to view users.

Viewing Genealogy of a Virtual Machine or Template

Infrastructure Automation detects the lineage of a virtual machine. View a virtual machine's lineage and compare the virtual machines that are part of its tree. This also allows tagging of virtual machines that share genealogy.

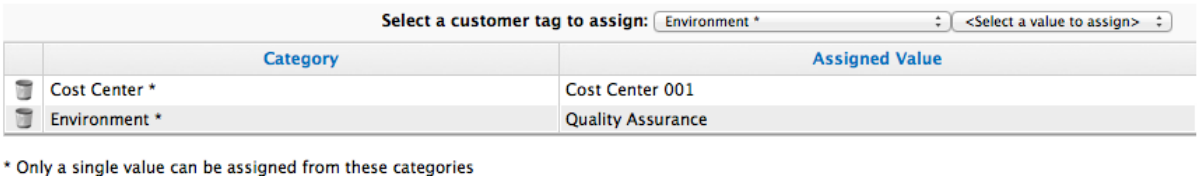
1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view genealogy.
3. Click on the item to view its **Summary**.
4. From the **Relationships** area in the **Summary**, click **Genealogy**.

Comparing Genealogy of a Virtual Machine or Template

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view genealogy.
3. Click on the item to view its **Summary**.
4. From the **Relationships** area in the **Summary**, click **Genealogy**.
5. Check the items to compare.
6. Click (**Compare Selected VMs**).

Tagging Virtual Machines or Templates with a Common Genealogy

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view genealogy.
3. Check the items to tag.
4. Click (**Policy**), and then (**Edit Tags**).
5. Select a customer tag from the first dropdown, and then a value for the tag.



Detecting Drift on Virtual Machines or Templates

The configuration of a virtual machine might change over time. **Drift** is the comparison of a virtual machine to itself at different points in time. The virtual machine needs analysis at least twice to collect this information. Detecting drift provides you the following benefits:

- See the difference between the last known state of a machine and its current state.

- Review the configuration changes that happen to a particular virtual machine between multiple points in time.
- Review the host and datastore association changes that happen to a particular virtual machine between multiple points in time.
- Review the classification changes that happen to a virtual machine between two time checks.
- Capture the configuration drifts for a single virtual machine across a time period.

Detect drift on virtual machines or templates:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click on the item to view its **Summary**.
3. From the **Relationships** area in the **Summary**, click **Drift History**.
4. Check the analyses to compare.
5. Click  (**Select up to 10 timestamps for Drift Analysis**) at the top of the screen. The results display.
6. Check the **Drift** sections on the left to view in your comparison.
7. Click **Apply**.
8. The following descriptions pertain to the **Expanded View**  . Whether you see the value of a property or an icon representing the property depends on the properties type.
    - A property displayed in the same color as the base means the compared analysis matches the base for that property.
    - A property displayed in a different color from the base means the compared analysis does not match the base for that property.
9. If you are in the **Compressed View**  , the values of the properties are not displayed. All items are described by the icons shown below.
    - A  (**checkmark**) means that the compared analysis matches the base for that property. If you hover over it, the value of the property will display.
    - A  (**triangle**) means the compared analysis does not match the base for that property. If you hover over it, the value of the property displays. Click the minus sign next to the sections name to collapse it.
10. To limit the scope of the view, you have three buttons in the **Resource** button area.
    - Click  (**All attributes**) to see all attributes of the sections you selected.
    - Click  (**Attributes with different values**) to see only the attributes that are different across the drifts.
    - Click  (**Attributes with the same values**) to see only the attributes that are the same across drifts.
11. To limit the mode of the view, there are two buttons in the **Resource** button area.
    - Click  (**Details Mode**) to see all details for an attribute.
    - Click  (**Exists Mode**) to only see if an attribute exists compared to the base or not. This only applies to attributes that can have a Boolean property. For example, a user account exists or does not exist, or a piece of hardware that does or does not exist.

This creates a drift analysis. Download the data or create a report from your drift for analysis using external tools.

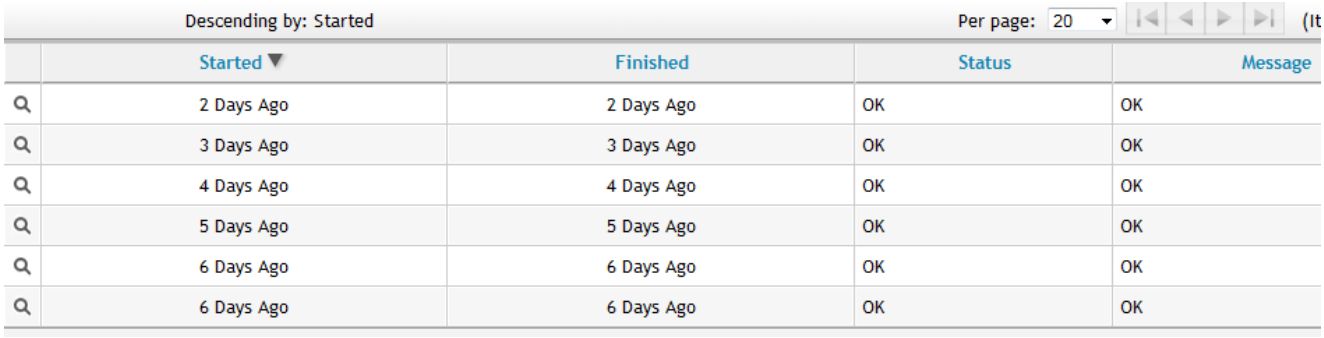
Creating a Drift Report for a Virtual Machine or Template

1. Create the comparison to analyze.
2. Click  (**Download**).
3. Click the output button for the type of report you want.
    - Click  (**Download drift report in text format**) for a text file.
    - Click  (**Download drift report in CSV format**) for a csv file.
    - Click (**Download drift report in PDF format**) for a PDF file.

Viewing Analysis History for a Virtual Machine or Template

Each time a SmartState Analysis is performed on a virtual machine, a record is created of the task. This information is accessed either from the virtual machine accordion or the virtual machine summary. Use this detail to find when the last analysis was completed and if it completed successfully. If the analysis resulted in an error, the error is shown here.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view genealogy.
3. Click on the item to view its **Summary**.
4. From the **Relationships** area in the **Summary**, click **Analysis History**. A history of up to the last 10 analyses is displayed.



Descending by: Started Per page: 20 (It

	Started ▼	Finished	Status	Message
	2 Days Ago	2 Days Ago	OK	OK
	3 Days Ago	3 Days Ago	OK	OK
	4 Days Ago	4 Days Ago	OK	OK
	5 Days Ago	5 Days Ago	OK	OK
	6 Days Ago	6 Days Ago	OK	OK
	6 Days Ago	6 Days Ago	OK	OK

5. Click on a specific analysis to see its details.

Viewing Disk Information for a Virtual Machine or Template

Each time a SmartState Analysis is performed on a virtual machine or template, information on the disks associated with the item is collected. This includes free and used space information as well as the type of disk and file system.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click on the item to view its **Summary**.
3. From **Datastore Allocation Summary**, click **Disks**.

A list of the disks for the item with type, file system, size, and usage information is displayed.

Reconfiguring Virtual Machines

Memory, processors, disks, ISOs, and CD/DVD Drives can be reconfigured on existing VMware and Red Hat Virtualization virtual machines using the **Reconfigure this VM** button.

You can reconfigure multiple components on a virtual machine using one request, or make each reconfiguration separately. Confirm your changes using the **Submit** button.

The following restrictions apply to adding and removing Red Hat Virtualization virtual machine disks: - Supported by Red Hat Virtualization 4.0 and above. - Only a single bootable disk is supported. - The virtual machine requires at least one existing disk to allow adding additional disks. This is because the destination storage cannot be specified from the Infrastructure Automation dialog, so the storage associated with the existing disk is reused.

Adding a Disk to a Virtual Machine

A disk can be added to a VMware or Red Hat Virtualization virtual machine with the following steps:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Click **Add Disk**.
5. Specify the disk type, mode, size, and dependency options.
6. Click **Submit**.

After the disk has been added, you can view the new disk by navigating to **Compute > Infrastructure > Virtual Machines**. . Open the target virtual machine, then click **Devices** to view the new disk.

Removing a Virtual Machine Disk

A disk can be removed from a VMware or Red Hat Virtualization virtual machine with the following steps:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Click **Delete** next to the disk to remove.
5. Click **Submit**.

Resizing a Virtual Machine Disk

Note:

This functionality is available on VMware virtual machines only.

You can extend a VMware virtual machine's disk with the following steps:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**), and then (**Reconfigure this VM**).
4. Click **Resize** next to the disk you want to resize to show resizing options.
5. Specify the desired memory size and units (MB or GB). A disk can only be extended in size.
6. Click **Confirm Resize** to select the values. Alternatively, selecting **Cancel Resize** shows the original values.
7. Click **Submit**.

The disk resizing request is then processed for the virtual machine.

Increasing or Decreasing a Virtual Machine's Memory Size

You can increase or decrease a VMware or Red Hat Virtualization virtual machine's memory size with the following steps:

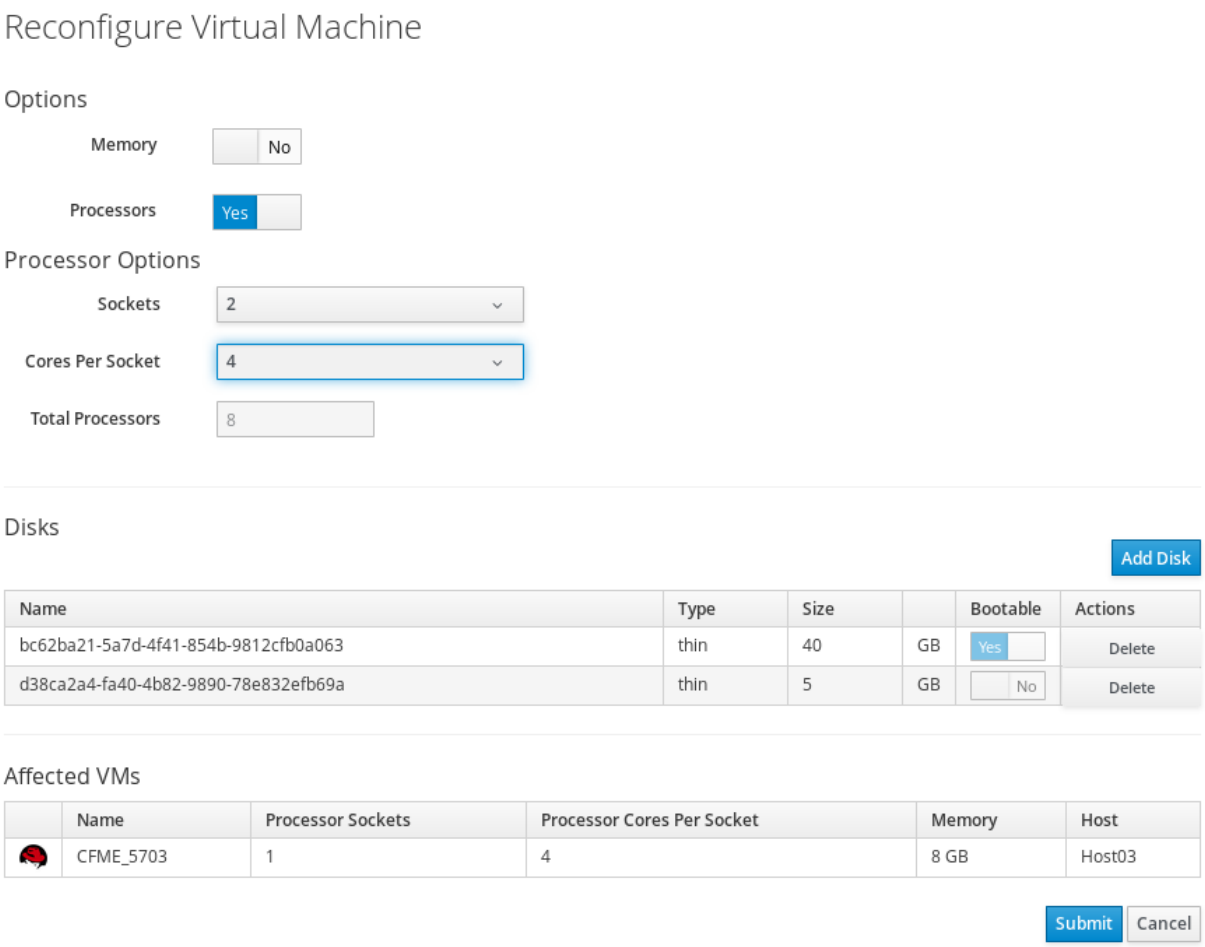
1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Select **Yes** next to **Memory** to show memory options.
5. Specify the desired memory size and units (MB or GB).
6. Click **Submit**.

The memory add request is then processed for the virtual machine.

Reconfigure a Virtual Machine's Processors

You can reconfigure a VMware or Red Hat Virtualization virtual machine's processors with the following steps:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then **Reconfigure this VM**.
4. Select **Yes** next to **Processors** to show processor options.



5. Specify the number of sockets and the number of cores per socket. The reconfiguration screen will calculate the **Total Processors**.
6. Click **Submit**.

The virtual machine's processors are then reconfigured.

Adding or Removing Virtual Machine Network Adapters

Note:

This functionality is available on VMware virtual machines only.

You can add or remove network adapters on a VMware virtual machine from the **Reconfigure this VM** button.

Adding a Network Adapter to a VMware Virtual Machine

To add a network adapter to a virtual machine:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Click **Add Network** next to the disk you want to resize to show resizing options.
5. Select the type of adapter from the list under **vLan**.
6. Click **Confirm Add**.
7. Click **Submit**.

A request to add the network adapter is then processed for the virtual machine. You can view the details for the new network adapter by navigating to the **Reconfigure this VM** area. When the network adapter is added, a name and MAC address are assigned to the adapter.

Removing a Network Adapter from a VMware Virtual Machine

To remove a network adapter from a virtual machine:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Click **Delete** next to the network adapter you want to remove.
5. Click **Submit**.

Attach or Remove an ISO (VMware Virtual Machines Only)

You can attach an ISO to a VMware virtual machine using following steps:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Under **CD/DVD Drives**, click **Connect**.
5. Select the **Host File** from the list to attach and click **Confirm Connect**.
6. Click **Submit**.

The ISO is now attached to the VMware virtual machine.

You can remove an ISO on VMware virtual machine using following steps:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine.
3. Click **Configuration**, and then (**Reconfigure this VM**).
4. Under **CD/DVD Drives**, locate the **Host File** and click **Disconnect**.
5. Click **Confirm**.
6. Click **Submit**.

The ISO is removed from the VMware virtual machine.

Viewing Event Logs for a Virtual Machine or Template

Using an **Analysis Profile**, collect event log information from your virtual machines.

Note:

This feature is only available for Windows.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click the accordion for the item to view event logs.
3. Click on the item to view its **Summary**.
4. From **Diagnostics** click **Event Logs**.

The collected event log entries are displayed. Sort this list by clicking on the column headers.

Remote Consoles

A console is a graphical window that allows you to view the start up screen, shut down screen, and desktop of a virtual machine, and to interact with that virtual machine in a similar way to a physical machine.

Infrastructure Automation offers the following support for HTML5-based VNC, SPICE, and WebMKS consoles:

- VNC and SPICE consoles for Red Hat Virtualization Manager with websocket proxy.
- VMware's WebMKS HTML5-based console type.
- VNC consoles for OpenStack using OpenStack-supplied websocket proxy.

Note:

- VMware no longer supports the MKS console type. Also, VMRC is no longer a browser plugin but a native desktop application. As a result, the VMware MKS and the VMRC plugin options have been disabled in Infrastructure Automation.
- Currently, attempting to connect to the VMware WebMKS console for a virtual machine fails when the server security type is set to **2** for that virtual machine.
- Due to VMware licensing restrictions, Red Hat cannot ship the WebMKS SDK in Infrastructure Automation. For information on how to configure WebMKS support in Infrastructure Automation, see Configuring WebMKS Support in Infrastructure Automation.

All of the above make use of the websocket protocol supported by all recent versions of browsers, and can use SSL to encrypt the websocket connection.

- OpenStack Infrastructure Automation only makes an API call to get the URL for the console and open that console in a web browser; see Directly Connect to a VNC Console in the Red Hat OpenStack Platform *Instances and Images Guide* for more details.
- Red Hat Enterprise Virtualization Manager and VMware By default, the websocket connection runs over HTTPS or HTTP based on how the application was accessed. Under an appliance, you will most likely use HTTPS, and, therefore, the websocket connection will be wss:// (websocket with SSL).

  When configuring Red Hat Virtualization Manager for virtual machine console access, set the display type for each virtual machine to **noVNC** or **SPICE HTML5**. For more information on configuring console options, see Configuring Console Options in the Red Hat Virtualization *Virtual Machine Management Guide*.

Configuring Console Access to VMware ESXi Hosts At A Network Layer

When configuring access to the VNC or HTML5 console, make sure that at a network layer:

- All VNC ports (5900-6000) are opened from the machine on which you access the Infrastructure Automation Console to the Infrastructure Automation.
- All VNC ports (5900-6000) are opened from the Infrastructure Automation to each VMware ESXi host running virtual machines that you want to access.
- The firewall on VMware ESXi hosts is enabled and that the VMware ESXi host firewall ports are opened.
- The VNC service (**gdbserver**) is running and that the **gdbserver** service has an association with ports 5900-6000 usually defined with a **/etc/vmware/firewall/service.xml** firewall rules configuration.

  The **gdbserver** ruleset must be enabled on each ESXi host running virtual machines that will be accessed through the HTML5 console or VNC console on the Infrastructure Automation. The ruleset can be configured on the host itself, or using the VMware vCenter web user interface.

The following procedures apply to VMware vCenter 5.0 and later.

Using SSH to Configure VMware ESXi Hosts to Enable Console Access

Configure the **gdbserver** ruleset on the host using SSH.

1. Access the host:

   ```
   # ssh host@example.com
   ```

2. Set the **gdbserver** parameter:

   ```
   # esxcli network firewall ruleset set --ruleset-id gdbserver --enabled true
   ```

3. Confirm that the ruleset is active:

   ```
   # esxcli network firewall ruleset list
   ```

Using the VMware vCenter Web Interface to Configure ESXi Hosts to Enable Console Access

Configure the **gdbserver** ruleset on the host using the VMware vCenter web user interface.

1. Select the ESXi host in the VMware vCenter web interface.
2. Click the **Manage** tab.
3. Click the **Settings** sub tab.
4. Click **System > Security Profile** from the list on the left.
5. Click **Edit**.
6. Select the **gdbserver** ruleset, and then click **OK**.

Configuring the VMware ESXi Host Firewall Ports for Console Access

Follow these steps to configure the VMware ESXi host firewall ports for HTML5 or VNC console access to guest virtual machine consoles. The firewall ports must be enabled on each VMware ESXi host running virtual machines that will be accessed through the HTML5 or VNC console on the Infrastructure Automation.

1. Log in to your vSphere Client and select **Home > Inventory > Hosts and Clusters**.
2. In the **Hosts/Clusters** tree view, select the VMware ESXi host you want to configure for HTML5 or VNC console access.

3. Select the **Configuration** tab and open the **Software** box.
4. Select **Security Profile**.
5. Browse to the **Firewall Properties** dialog window by selecting the **Properties** link from the **Firewall** section.
6. In the **Firewall Properties**, scroll down to **GDB Server** and select it.
7. Click **OK**.

Configuring WebMKS Support in Infrastructure Automation

Complete the following steps to enable WebMKS support in Infrastructure Automation.

1. Log in to the Infrastructure Automation user interface appliance console as the root user.
2. On the Infrastructure Automation user interface appliances, create a folder titled **`webmks`** in the **`/var/www/miq/vmdb/public/`** directory.

   `/var/www/miq/vmdb/public/webmks`
3. Download and extract the contents of VMware WebMKS SDK into the **`webmks`** folder.

Opening a Console for a Virtual Machine

Open a web-based VNC or SPICE console for a virtual machine.

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click on the virtual machine that you want to access.
3. Click (**Access**) and select **VM Console** or **Web Console**.

The virtual machine console opens in a new tab in your browser.

Managing Virtual Machine Snapshots

A snapshot is a view of a virtual machine's operating system and applications on any or all available disks at a given point in time. Take a snapshot of a virtual machine before you make a change to it that may have unintended consequences. You can use a snapshot to return a virtual machine to a previous state.

View virtual machines by infrastructure providers by navigating to **Compute > Infrastructure > Providers**.

Creating a Virtual Machine Snapshot

To create a snapshot of a virtual machine:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine, and open it to view its details.
3. Under **Properties**, click **Snapshots** to show information about the virtual machine's snapshots.
4. Click (**Create a new snapshot for this VM**).
5. Enter snapshot details in **Description**.
6. (Optional) Select **Snapshot VM memory** for the snapshot to preserve the virtual machine's current runtime state. This option will appear only if the VM **Power State** is (**On**).
7. Click **Create**.

Deleting a Virtual Machine Snapshot

To delete a virtual machine snapshot:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the target virtual machine, and open it to view its details.
3. Under **Properties**, click **Snapshots** to show information about the virtual machine's snapshots.
4. Select the snapshot to delete under **Available Snapshots**.

5. Click and select **Delete Selected Snapshot**.
6. Click **OK**.

Reverting a Virtual Machine to a Snapshot

A virtual machine can be reverted to a previous state using a snapshot.

The virtual machine state must be powered `off` for the **Revert to selected snapshot** option to be available.
To revert a virtual machine to a snapshot:

1. Power the virtual machine off.
2. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
3. Select the target virtual machine, and open it to view its details.
4. Under **Properties**, click **Snapshots** to show information about the virtual machine's snapshots.
5. Select the snapshot to revert to under **Available Snapshots**.
6. Click (**Revert to selected snapshot**).
7. Click **OK**.

Creating a Template Based on a Virtual Machine

A template is a copy of a virtual machine (VM) that you can use to simplify the subsequent, repeated creation of similar VMs. Templates capture the configuration of software, configuration of hardware, and the software installed on the VM on which the template is based.

Note:

Virtual machines must meet the following criteria to publish as templates:

- The VM must not be blank, archived or orphaned.
- The VM must be associated with a oVirt / RHV provider.
- oVirt / RHV provider 4.0 or higher, with `use_ovirt_engine_sdk` set to `true` in `/var/www/miq/vmdb/config/settings.yml` on your Infrastructure Automation appliance.
- The VM power state is `OFF`.
- Sealing a template is not supported for Windows OS VMs.

To create a template based on an existing VM:

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click on a oVirt / RHV provider.
3. Select a VM.
4. Click (**Lifecycle**), then click (**Publish selected VM to a Template**).
5. Under the **Request** tab, provide **Request Information** and identify a **Manager**:
    1. Enter an **E-Mail** address.
    2. Provide a **First** and **Last** name in the individual fields.
    3. Enter a **Manager** name.
6. Assign tags to the new template in the **Purpose** tab.
    1. Click on a tag category and check applicable tags.
7. Use fields in the **Catalog** tab to identify the source VM and provide details for the template:
    1. Select the source VM from the **Name** field.
    2. (Optional) Check **Seal Template**.

        Note:

        Sealing, which uses the `virt-sysprep` command, removes system-specific details from a virtual machine before creating a template based on that VM. This prevents the original VM's details from appearing in subsequent VMs that are created using the same template. It also ensures the functionality of other features, such as predictable vNIC order.

3. The **Number of VMs** indicates the output will be a single template.
4. Provide a **Name** and **Description** for the created template.

8. Under the **Environment** tab, specify information for the storage domain in which to create the template's disks.
    1. Check **Choose Automatically** to allow Infrastructure Automation to determine the destination cluster and datastore.

       Or, manually enter cluster and datastore information using the following steps:
    2. Select the cluster with which to associate the template from the **Cluster** list. By default, this is identical to the cluster of the source VM.
    3. Choose a **Datastore** for the destination to create the template's disks.
9. Provide provisioning and retirement information for VMs based on the template in the **Schedule** tab.
    1. Select **Schedule** or **Immediately on Approval** for **Schedule Info** to determine when to provision the VM.
    2. Set the **Time to Retirement** under **Lifespan** using the drop-down menu.
10. Click **Submit**.

Retiring Virtual Machines

Specify a retirement date and time for virtual machines or instances using Infrastructure Automation.

To set retirement date and time:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the virtual machines or instances for retirement.
3. Click (**Lifecycle**), then select **Set Retirement Date**.
4. Under **Enter Retirement Date as**, select **Specific Date and Time**.
5. Select a date using the calendar control, then click (**Select Time**).
6. Specify the hour and minute for retirement.
7. Select a **Retirement Warning** from the list.
8. Click **Save**.

Migrating Virtual Machines

Infrastructure Automation supports migrating virtual machines between datacenters, clusters, resource pools, folders and hosts. Create and submit a request with detailed environment information to which to migrate the VM.

To create a request to migrate a virtual machine:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Click **VMs & Templates** on the accordion menu.
3. Select the VM to migrate.
4. Click **Configuration**, and then **Lifecycle**.
5. Click **Migrate selected items**. The Migrate Virtual Machine form will appear.
6. Under the **Request** tab:
    1. Provide an **E-mail** address, **First Name** and **Last Name**.
    2. Include any **Notes** and a Manager's **Name**.
7. Under the **Environment** tab:
    1. Select a **Datacenter** to migrate the VM to.
    2. Select a **Cluster** from the list.
    3. Choose a **Resource Pool**.
    4. Select a **Folder** to migrate the VM to.

5. Under **Host** choose a **Filter** and click on a **Name** from the list that appears.
6. **Filter** the **Datastore** and click a **Name** from the results.

8. Under the **Schedule** tab:
    1. Select **Schedule Info** for migrating the virtual machine. Provide a **Provision on** date if selecting **Schedule**.
9. Click **Submit**.

Resource Pools

Resource pools are used to allocate CPU and memory across a group of virtual machines.

Removing a Resource Pool

If a resource pool is decommissioned or requires troubleshooting, it might require removal from the VMDB.

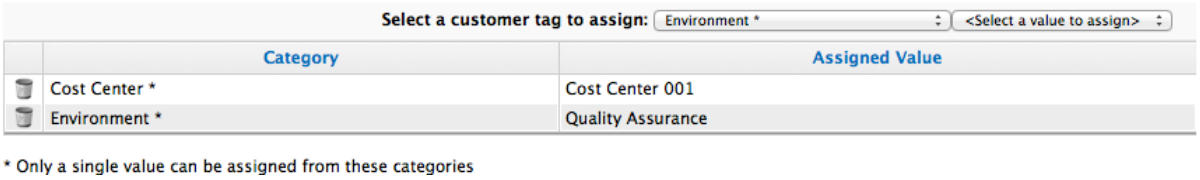
1. Browse to menu: **Compute > Infrastructure > Resource Pools**.
2. Click on the resource pool to remove.
3. Click **Configuration**, and then (**Remove from the VMDB**).
4. Click **OK**.

The resource pool is removed. The virtual machines remain in the VMDB, but are no longer associated with this resource pool.

Tagging a Resource Pool

Use tags to categorize a resource pool.

1. Browse to menu: **Compute > Infrastructure > Resource Pools**.
2. Click the resource pool to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a customer tag from the first dropdown, and then a value for the tag.



Viewing the Resource Pool Summary

Use the Resource Pool Summary to see the number of discovered virtual machines, the parent host, and the parent cluster. It is the default view when you click on one resource pool.

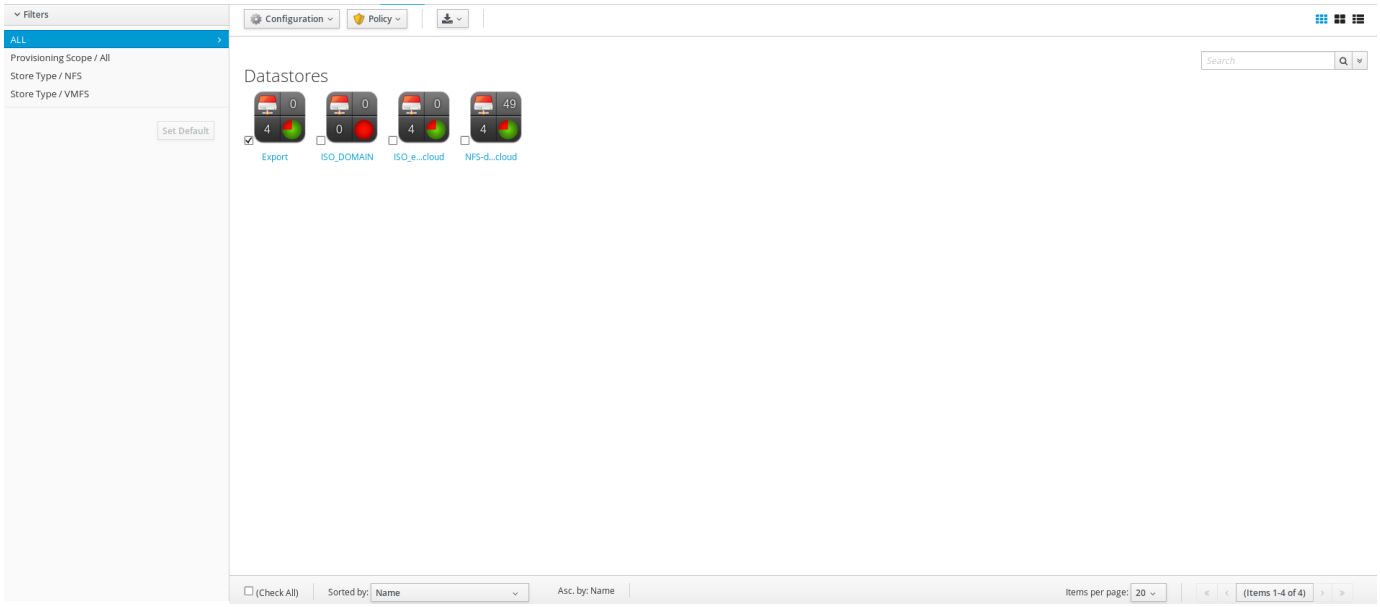
Resource Pools Accordion

Use the **Resource Pools** accordion to access the properties and objects associated with the resource pool.

- Click **Properties** to view the Resource Pools summary screen.
- Click **Relationships** to see the clusters, virtual machines, and hosts related to this resource pool.

Datastores

A storage location is considered a device where digital information resides and is connected to a resource. Infrastructure Automation detects, analyzes, and collects capacity and utilization data for both VMFS and NFS datastores. Datastores connected to a provider are automatically created on discovery. On creation of a repository, a datastore is automatically created.



After detecting datastores, you might want to examine them more closely to see virtual machines, hosts, and available space.



1. File system type
2. Number of hosts
3. Number of virtual machines
4. Available space

Performing SmartState Analysis on Datastores

Analyze a datastore to collect information on the types of files on a datastore, and to see the number of managed/registered, managed/unregistered, and unmanaged virtual machines. To perform a SmartState analysis, the datastore is accessible from a running host and valid security credentials are supplied for that host.

Note:

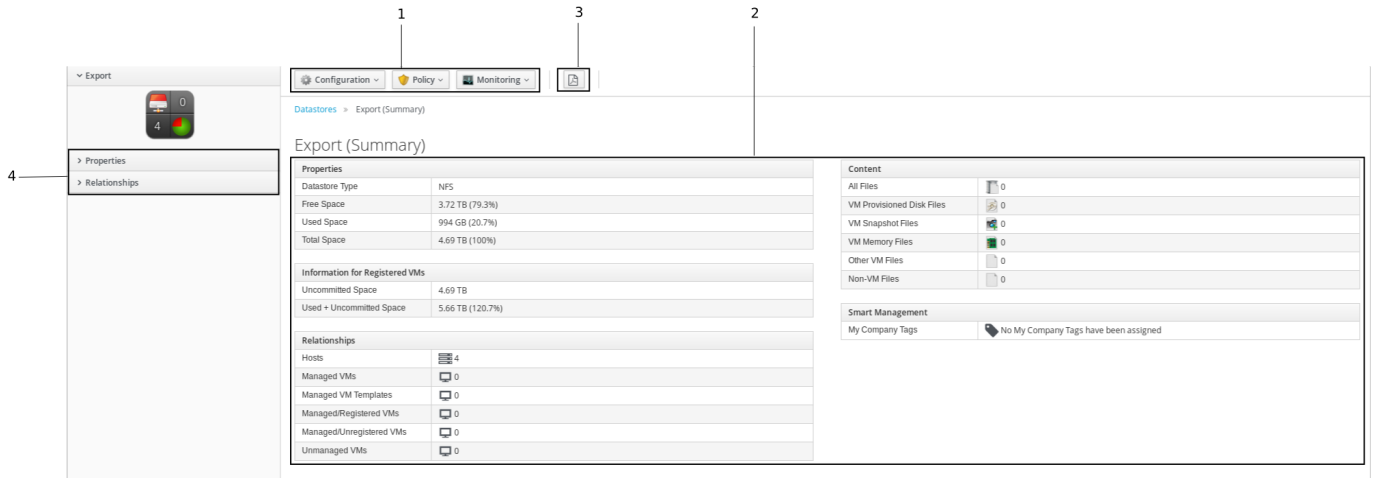
- SmartState analysis on datastores is processed by the **Provider Operations** role. It is enabled by default.
- Be aware that executing a SmartState analysis on a datastore from the console takes a while to return data on the content. If capacity and utilization roles are enabled, Infrastructure Automation performs the analysis automatically on a scheduled basis approximately every 24 hours.

1. Browse to menu: **Compute > Infrastructure > Datastores**.
2. Select the datastores to analyze.
3. Click **Configuration**, and then (**Perform SmartState Analysis**).
4. Click **OK**.

Viewing a Datastore

You can click on a specific datastore to view its details. The screen provides you with a datastore taskbar, virtual thumbnail, accordion, and summary.

Datastore Management Screen.



1. **Datastore Taskbar**: Choose between **Configuration**, **Policy** and **Monitoring** options for the selected datastore.
2. **Datastore Summary**: See summary such as datastore properties, storage, VM information.
3. **Datastore PDF**: Generates datastore summary in PDF format.
4. **Datastore Accordion**: See details about **Properties**, **Relationships**, **Storage Relationships** and **Content** for the chosen datastore.

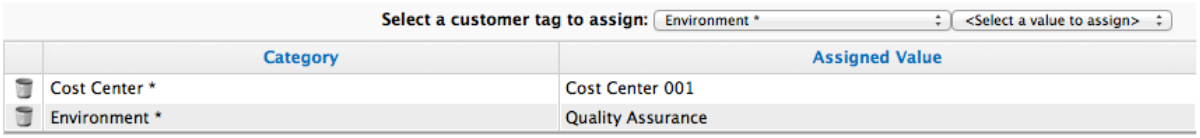
Note:

To view **Content** section details, run a SmartState Analysis on the datastore. For information on how to perform SmartState Analysis, see Performing SmartState Analysis on Datastores.

Tagging a Datastore

Use tags to categorize a datastore.

1. Browse to menu: **Compute > Infrastructure > Datastores**.
2. Click the datastore to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a customer tag from the first list, and then a value for the tag from the second list.



5. Select more tags as required.
6. Click **Save**.

Viewing Capacity and Utilization Charts for a Datastore

You can view capacity and utilization data for a datastore.

Note:

Infrastructure Automation requires network visibility to your provider assigned the server role of *Capacity & Utilization Collector* to enable this feature.

1. Browse to menu: **Compute > Infrastructure > Datastores**, then click the Datastore for which to view Capacity and Utilization data.
2. Click (**Monitoring**), and then (**Utilization**).
3. From **Interval**, select to view hourly or daily data points and the dates to view data.
4. Use **Show VM Types** to include only managed/registered, managed/unregistered, or unmanaged virtual machines.

- **Managed/Registered VM** - A virtual machine connected to a host and exists in the VMDB. Also, a template connected to a management system and exists in the VMDB.

  Note:

  Templates cannot be connected to a host.

- **Managed/Unregistered VM** - A virtual machine or template that resides on a repository that is no longer connected to a management system or host, but exists in the VMDB. A virtual machine previously considered registered might become unregistered if the virtual machine is removed from management system inventory.
- **Not Managed** - Files discovered on a datastore that do not have a virtual machine associated with them in the VMDB. These files might be registered to a management system that Infrastructure Automation does not have configuration information. Possible causes might be the management system has not been discovered or the management system has been discovered but no security credentials are provided.

5. Use **Time Profiles** to select a time range for the data.

Note:

Daily charts only include full days of data. If a day does not include all the 24 data points for a day, the data does not show for that day.

For information about data optimization including utilization trend reports, see Data Optimization.

Removing a Datastore

If a datastore no longer contains any files associated with the virtual environment, remove it from the VMDB. This button is enabled only if a datastore is completely empty.

1. Browse to menu: **Compute > Infrastructure > Datastores**.
2. Click on the Datastore to remove.
3. Click **Configuration**, and then (**Remove Datastore from Inventory**).
4. Click **OK**.

Data Optimization

Infrastructure Automation optimization functions allow you to view utilization trends in your environment. In addition, you can predict where you have capacity for more virtual machines, see Planning Where to Put a New Virtual Machine in the *Deployment Planning Guide*.

Note: To access the utilization report features, you will first need to enable data collection in Infrastructure Automation; see the following sections in the *Deployment Planning Guide*:

- Assigning the Capacity and Utilization Server Roles
- Capacity and Utilization Data Collected

Utilization Trends

Infrastructure Automation allows you to view the resource utilization of your clusters, providers, and datastores. You can choose from summary, details, or report view.

Viewing a Utilization Trend Summary

This procedure shows you how to view a utilization trend summary.

1. Browse to menu: **Overview > Utilization**.
2. Click **Summary** if it is not already selected.
3. Expand the tree on the left side until you can see the wanted providers, clusters, or datastores.
4. Click the item.
5. Use the **Options** section in the **Summary** tab to change the characteristics of the data.
   - Use **Trends** to select how far back you want to calculate the trend.
   - Use **Selected Day** for the date you want to see percent utilization for in the chart on the **Summary** tab.
   - Use **Classification** to see trends only for a specific applied tag.
   - Use **Time Profile** to select an existing time profile. If the logged on user does not have any time profiles available, this option will not show.
   - Select a **Time Zone**.

Viewing Detail Lines of a Utilization Trend

This procedure shows you how to view detail lines of a utilization trend.

1. Browse to menu: **Overview > Utilization**.
2. Expand the tree on the left side until you can see the wanted providers, clusters, or datastores.
3. Click the item.
4. Click **Details** if it is not already selected.
5. From the **Options** area, select how far back you want to view the trends for and any classifications you want to use.

Viewing a Report of a Utilization Trend

To find out more about resource utilization, view utilization trend reports.

1. Browse to menu: **Overview < Utilization**.
2. Expand the tree on the left side until you can see the wanted providers, clusters, or datastores.
3. Click the item.
4. Click **Report** if it is not already selected.
5. From the **Options** area, select how far back you want to view the trends for and any classifications you want to use.

PXE Servers

PXE servers are used by Infrastructure Automation to bootstrap virtual machines for the purpose of provisioning. They include images for different operating systems that can be customized using customization templates and are used in conjunction with IPMI Servers.

Availability Zones

An availability zone is a provider-specific method of grouping cloud instances and services. Infrastructure Automation uses Amazon EC2 regions and OpenStack Nova zones as availability zones.

Viewing an Availability Zone

You can click on an availability zone to view its details. The screen provides you with an availability zone accordion and an availability zone summary page.

- You can choose between graphical or text view of the datastore summary.
- Use the availability zone accordion to view the **Properties** of the zone and its **Relationships** to other cloud resources.
- Use the availability zone summary to see details on **Relationships** (**Cloud Provider**, **Instances**) and **Smart Management** (**Company Tags**).

Viewing Availability Zone Relationships

Use the availability zone accordion's **Relationship** section to see items related to an availability zone.

1. Browse to menu: **Compute > Clouds > Availability Zones**.
2. Click the availability zone to view the configuration.
3. From the availability zone accordion, click **Relationships**.
4. Click the type of resource relationship to view as a list.

Cloud Tenants

A tenant is an OpenStack term for an organizational unit or project. Infrastructure Automation creates cloud tenants to match existing OpenStack tenants for managing resources and controlling visibility of objects.

Note:

OpenStack tenant mapping must be enabled for cloud tenants to be created. See Tenant Mapping for details.

OpenStack uses tenants for the following reasons:

- Assigning users to a project
- Defining quotas for a project
- Applying access and security rules for a project
- Managing resources and instances for a project

This helps administrators and users organize their OpenStack environment and define limits for different groups of people. For example, one project might require higher quotas and another project might require restricted access to certain ports. OpenStack allows you to define these limits and apply them to a project.

Infrastructure Automation can abstract information from tenants including quotas and relationships to other OpenStack objects.

To see multiple tenants in Infrastructure Automation, the user authenticating to your OpenStack environment from Infrastructure Automation must be configured to have visibility into these tenants.

Note:

This section describes OpenStack cloud tenant usage. For information about tenants created in Infrastructure Automation for access and resource control, see Access Control in *General Configuration*.

Tenant Mapping

When adding an OpenStack cloud or infrastructure provider, enable *tenant mapping* in Infrastructure Automation to map any existing tenants from that provider.

This means Infrastructure Automation will create new cloud tenants to match each existing OpenStack tenant; each new cloud tenant and its corresponding OpenStack tenant will have identical resource assignments (including user and role synchronization) with the exception of quotas. Tenant quotas are not synchronized between Infrastructure Automation and OpenStack, and are available for reporting purposes only. You can manage quotas in Infrastructure Automation but this will not affect the quotas created in OpenStack.

During a provider refresh, Infrastructure Automation will also check for any changes to the tenant list in OpenStack. Infrastructure Automation will create new cloud tenants to match any new tenants, and delete any cloud tenants whose corresponding OpenStack tenants no longer exist. Infrastructure Automation will also replicate any changes to OpenStack tenants to their corresponding cloud tenants.

If you leave tenant mapping disabled, Infrastructure Automation will not create cloud tenants or tenant object hierarchy from OpenStack.

Viewing Cloud Tenant Relationships

From **Compute > Clouds > Tenants**, click on a specific cloud tenant to view its details.

The screen provides you with a cloud tenant accordion and a cloud tenant summary.

- Use cloud tenant summary views to change how you are looking at the **Summary**.
- Use the cloud tenant accordion to view the **Properties** of the tenant and its **Relationships**.
- Use the cloud tenant summary to see details on **Relationships** (**Cloud Provider**, **Security Groups**, **Instances**, and **Images**), **Quotas** (including all OpenStack **Compute**, **Network**, and **Volume** quotas) and **Smart Management** (**Company Tags**).

Volumes

A volume is a block storage device that you can attach to or detach from an instance to manage the storage available to that instance. Volumes are managed through storage managers, which are added automatically to Infrastructure Automation when the corresponding provider is added.

Amazon Elastic Block Store Manager Volumes

This section outlines the actions that you can perform on Amazon Elastic Block Store manager volumes.

Creating Volumes

You can create and attach volumes to your storage manager.

To create a volume:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Click **Configuration**, then click (**Add a new Cloud Volume**).
3. Select the Amazon Elastic Block Store manager from the **Storage Manager** list.

4. Select an availability zone from the **Availability Zone** list.
5. Enter a **Volume Name**.
6. Select the type of the volume from the **Cloud Volume Type** list.

   Note:

   See Amazon EBS Volume Types for more information on volume types.
7. Enter the size of the volume in gigabytes (GB).
8. Select whether the volume should be encrypted using the **Encryption** toggle.
9. Click **Add**.

The volume appears in the list of volumes after it has been provisioned.

Creating a Snapshot of a Volume

You can create a snapshot of a volume to preserve the state of the volume at a specific point in time. The snapshot can be used to create a duplicate of the volume.

To create a snapshot of a volume:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Click the volume to snapshot to open the volume's summary page.
3. Click **Configuration**, then click **Create a Snapshot of this Cloud Volume**.
4. Enter a name for the snapshot in **Snapshot Name**.
5. Click **Save**.

Click **Cloud Volume Snapshots** on the summary page of a volume to view the snapshots for that volume.

Attaching a Volume to an Instance

To attach a volume to an instance:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to attach.
3. Click **Configuration**, then click **Attach selected Cloud Volume to an Instance** to open the **Attach Cloud Volume** screen.
4. Select an instance from the list.
5. Optionally, enter a **Device Mountpoint**.
6. Click **Attach**.

Detaching a Volume from an Instance

To detach a volume from an instance:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to detach.
3. Click **Configuration**, then click **Detach selected Cloud Volume from an Instance** to open the **Detach Cloud Volume** screen.
4. Select an instance from the list.
5. Click **Detach**.

Editing a Volume

You can edit several properties of existing volumes.

To edit a volume:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to edit to open its summary page.
3. Click **Configuration**, then click (**Edit this Cloud Volume**).
4. Enter a new **Volume Name**.
5. Select a new volume type from the **Cloud Volume Type** list.

   Note:

   See Amazon EBS Volume Types for more information on each of the volume types.
6. Enter a new size in gigabytes.
7. Click **Save**.

Deleting a Volume

To delete a volume from a storage manager:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to delete.
3. Click **Configuration**, then click (**Delete selected Cloud Volumes**).

OpenStack Block Storage Manager Volumes

This section outlines the actions that you can perform on OpenStack Block Storage manager volumes.

Creating Volumes

You can create and attach volumes to your storage manager.

To create a volume:

After creating a volume, only the volume name can be edited.

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Click **Configuration**, then click (**Add a new Cloud Volume**).
3. Select the OpenStack Block Storage manager from the **Storage Manager** list.
4. Enter a **Volume Name**.
5. Enter the size of the volume in gigabytes (GB).
6. Under **Placement**, select the cloud tenant to attach it to.
7. Click **Add**.

The volume appears in the list of volumes after it has been provisioned.

Creating a Backup of a Volume

You can create a backup of a volume to protect against data loss, and restore it in the future.

For OpenStack Block Storage managers, the `openstack-cinder-backup` service must be enabled on the OpenStack Block Storage manager to create a volume backup.
To create a backup of a volume:

1. Browse to menu: **Storage < Block Storage < Volumes**.
2. Click the volume you want to back up to open the volume's summary page.
3. Click **Configuration**, then click (**Create a Backup of this Cloud Volume**).
4. Enter a name for the backup in **Backup Name**.
5. (Optional) Select **Incremental?** to take an incremental backup of the volume instead of a full backup.

Note:

You can take an incremental backup of a volume if you have at least one existing full backup of the volume. An incremental volume saves resources by capturing only changes made to the volume since its last backup. See Create an Incremental Volume Backup in the *Storage Guide* for more information.

6. (Optional) Select **Force?** to allow backup of a volume attached to an instance.

   Note:

   Selecting the **Force** option will back up the volume whether its status is *available* or *in-use*. Backing up an *in-use* volume ensures data is crash-consistent.

7. Click **Save**.

View a volume's backups by clicking **Cloud Volume Backups** on the volume's summary page.

Note:

See Back Up and Restore a Volume in the *Storage Guide* for more information about backups.

Restoring a Volume from a Backup

In case of data loss, you can restore a volume from a backup with the following steps:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Click the volume whose backup you want to restore. This will open the volume's summary page.
3. Click **Configuration**, then click (**Restore from a Backup of this Cloud Volume**).
4. Select the volume to restore from in the **Cloud Volume Backup** list.
5. Click **Save**.

Restoring a Cloud Volume from a Backup

In case of data loss, you can restore from a cloud volume backup with the following steps:

1. Browse to menu: **Storage > Block Storage > Volume Backups**.
2. Select a **Cloud Volume Backup** to restore.
3. Click **Configuration**, then click (**Restore backup to Cloud Volume**).
4. Select the **Volume** to restore from the backup.
5. Click **Save**.

Deleting a Cloud Volume Backup

Delete unnecessary cloud volume backups through the following steps:

1. Browse to menu: **Storage > Block Storage > Volume Backups**.
2. Select the **Cloud Volume Backups** to delete.
3. Click **Configuration**, then click (**Delete selected Backups**).
4. Click **OK** to confirm your choice.

Creating a Snapshot of a Volume

You can create a snapshot of a volume to preserve the state of the volume at a specific point in time. The snapshot can be used to create a duplicate of the volume.

To create a snapshot of a volume:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Click the volume to snapshot to open the volume's summary page.
3. Click **Configuration**, then click (**Create a Snapshot of this Cloud Volume**).

4. Enter a name for the snapshot in **Snapshot Name**.
5. Click **Save**.

Click **Cloud Volume Snapshots** on the summary page of a volume to view the snapshots for that volume.

Note:

See Create, Use, or Delete Volume Snapshots in the *Storage Guide* for more information about snapshots.

Attaching a Volume to an Instance

To attach a volume to an instance:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to attach.
3. Click **Configuration**, then click (**Attach selected Cloud Volume to an Instance**) to open the **Attach Cloud Volume** screen.
4. Select an instance from the list.
5. Optionally, enter a **Device Mountpoint**.
6. Click **Attach**.

Detaching a Volume from an Instance

To detach a volume from an instance:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to detach.
3. Click **Configuration**, then click **Detach selected Cloud Volume from an Instance** to open the **Detach Cloud Volume** screen.
4. Select an instance from the list.
5. Click **Detach**.

Editing a Volume

Only the volume name can be edited on an existing volume.

To edit a volume's name:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to edit to open its summary page.
3. Click **Configuration**, then click (**Edit this Cloud Volume**).
4. Enter the new **Volume Name**.
5. Click **Save**.

Deleting a Volume

To delete a volume from a storage manager:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to delete.
3. Click **Configuration**, then click (**Delete selected Cloud Volumes**).

IBM Power Systems Virtual Servers Block Storage Manager Volumes

This section outlines the actions that you can perform on IBM Power Systems Virtual Servers Block Storage manager volumes.

Creating Volumes

You can create and attach volumes to your storage manager.

To create a volume:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Click **Configuration**, then click **Add a new Cloud Volume**.
3. Select the desired IBM Power Systems Virtual Servers instance from the **Storage Manager** list.
4. Enter a **Volume Name**.
5. Select the type of the volume from the **Cloud Volume Type** list.

   Note:

   See Power Systems Virtual Servers documentation for more information on volume types.
6. Enter the size of the volume in gigabytes (GB).
7. Click **Add**.

The volume appears in the list of volumes after it has been provisioned.

Attaching a Volume to an Instance

To attach a volume to an instance:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to attach.
3. Click **Configuration**, then click (**Attach selected Cloud Volume to an Instance**) to open the **Attach Cloud Volume** screen.
4. Select an instance from the list.

   Note:

   Note: A VM cannot have disks from different storage types. Only VMs matching storage type will be available for selection.
5. The **Device Mountpoint** field is not used.
6. Click **Attach**.

Detaching a Volume from an Instance

To detach a volume from an instance:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to detach.
3. Click **Configuration**, then click **Detach selected Cloud Volume from an Instance** to open the **Detach Cloud Volume** screen.
4. Select an instance from the list.
5. Click **Detach**.

Deleting a Volume

To delete a volume from a storage manager:

1. Browse to menu: **Storage > Block Storage > Volumes**.
2. Select the volume to delete.
3. Click **Configuration**, then click (**Delete selected Cloud Volumes**).

Flavors

Flavors indicate the resource profiles available for instances. Each flavor contains a value set for CPUs, CPU cores and memory. Flavors allow you to pre-configure resource settings, which you can then apply during instance provisioning. You can also change the flavor of a provisioned instance; see Resizing an Instance for instructions.

Infrastructure Automation provides the ability to view individual flavor information and instances currently using the flavor.

Creating a Flavor

You can create a new flavor for the provider.

1. Browse to menu: **Compute > Clouds > Flavors**.
2. Click **Configuration**, then click (**Add a new Flavor**).
3. Select the provider from the **Provider** list.
4. Enter a **Name** for the flavor.
5. Enter **RAM size in MB**.
6. Enter **VCPUs**.
7. Enter **Disk size in GB**.
8. Enter **Swap size in MB**.
9. Enter **RXTX factor**. This is an optional property allows servers with a different bandwidth to be created with the RXTX factor. The default value is 1. That is, the new bandwidth will be the same as that of the attached network.
10. Click **Public** to set `True` or `False`. The default is `True`. If you set it to false, select cloud tenants from the **Cloud Tenant** list.
11. Click **Add**.

Viewing a Flavor

You can click on a specific flavor to view its details. The screen provides you with a flavor accordion and a flavor summary.

- Use flavor summary views to change how you are looking at the summary.
- Use the flavor accordion to view the **Properties** of the flavor and its **Relationships**.
- Use the flavor summary to see details on **Properties** (**CPUs**, **CPU Cores**, **Memory**), **Relationships** (**Cloud Provider**, **Instances**), and **Smart Management** (**Company Tags**).

Viewing Flavor Relationships

Use the **Relationship** section in the flavor accordion to see items related to the flavor.

1. Browse to menu: **Compute > Clouds > Flavors**.
2. Click a flavor to view the configuration.
3. From the flavor accordion, click **Relationships**.
4. Click the type of resource to see the flavor's relationships.

Deleting a Flavor

1. Browse to menu: **Compute > Clouds > Flavors**.
2. Select the flavors you want to remove from the list.
3. Click **Configuration**, then click **Remove selected Flavors**.
4. Cick **OK** on the warning window to remove selected flavors permanently.

Cloud Networks

Infrastructure Automation enables configuration and administration for the software-defined networking component of Red Hat OpenStack Platform. The virtual network infrastructure enables connectivity between instances and the physical external network.

This section describes common cloud network administration tasks, such as adding and removing subnets and routers to suit your Red Hat OpenStack Platform providers.

Creating and Administering Cloud Networks

Create a network to provide instances a place to communicate internally and receive IP addresses using Dynamic Host Configuration Protocol (DHCP). A network can also be integrated with external networks in your Red Hat OpenStack Platform deployment or elsewhere, such as the physical network.

Note:

- Keystone API v3 is required to create cloud tenants on Red Hat OpenStack Platform cloud providers. For more information, see OpenStack Identity (keystone) in the Red Hat OpenStack Platform *Architecture Guide*.
- For further details on cloud network objects and administration, see the *Networking Guide* in the Red Hat OpenStack Platform documentation..

Adding a Cloud Network

Add a new cloud network following the steps in this procedure:

1. Browse to menu: **Network > Networks**.
2. Click **Configuration** and then click **Add a new Cloud Network**.
3. In the **Network Providers** area, select a **Network Manager** from the drop-down menu.
4. Under **Placement**, select a **Cloud Tenant**.
5. In the **Network Provider Information** section, choose a **Provider Network Type**.
    - If **Local** is selected, provide a **Physical Network** name.
    - If **GRE** is selected, provide a **Physical Network** name and **Segmentation ID**.
6. In the **Network Information** area:
    1. Provide a descriptive **Network Name** based on the role the network will perform.
    2. Enable an **External Router**
    3. Set the **Administrative State** to control whether the network is immediately available.
    4. Establish **Shared** status of the network.
7. Click **Add**.

Editing Cloud Network Details

To edit network information details of a cloud network:

1. Browse to menu: **Networks > Networks**.
2. Select a network from the list view.
3. Click **Configuration**, then  (**Edit selected Cloud Network**).
4. Edit **Network Information** fields.
5. Click **Save**.

Deleting a Cloud Network

To delete a cloud network:

1. Browse to menu: **Networks > Networks**.
2. Select a cloud network from the list view.
3. Click **Configuration**, then click **Delete selected Cloud Networks**.

Creating and Administering Subnets

Infrastructure Automation enables creation and administration of subnets for your cloud networks. Create subnets in pre-existing networks as means to grant network connectivity to instances. Subnets are the means by which instances are granted network connectivity. Each instance is assigned to a subnet as part of the instance creation process.

Consider proper placement of instances to best accommodate their connectivity requirements:

- Tenant networks in OpenStack Networking can host multiple subnets.
- Subnets are isolated from one another.
- Host distinctly different systems on different subnets within the same network.
- Instances on one subnet that wish to communicate with another subnet must have traffic directed by a router.
- Place systems requiring a high volume of traffic amongst themselves in the same subnet, avoiding routing and subsequent latency and load issues.

Adding a Subnet

Add a subnet to an existing cloud network following the procedure below.

1. Browse to menu: **Networks > Subnets**.
2. Click **Configuration**, then click **Add a new Cloud Subnet**.
3. Select a **Network Manager**.
4. Under **Placement**, select a **Cloud Tenant**.
5. Provide the following **Cloud Subnet** details:
    1. A descriptive **Subnet Name**.
    2. The **Gateway** IP address of the router interface for the default gateway.
    3. **Enable DHCP** services for the subnet. DHCP allows automated distribution of IP settings to instances.
    4. Select the **IP Version**. The IP address range in the Network Address field must match whichever version you select.
    5. Input the **Subnet CIDR** address in CIDR format, which contains the IP address range and subnet mask in one value.

       Note:

       Determine the address by calculating the number of bits masked in the subnet mask and append that value to the IP address range. For example, the subnet mask 255.255.255.0 has 24 masked bits. To use this mask with the IPv4 address range 192.168.122.0, specify the address 192.168.122.0/24.

Editing a Cloud Subnet

To edit the details of a cloud subnet:

1. Browse to menu: **Networks > Subnets**.
2. Click on a subnet from the list view.
3. Click **Configuration**, then  **Edit this Cloud Subnet**.
4. Edit **Cloud Subnet details** fields.
5. Click **Save**.

Configuring Network Routers

Infrastructure Automation enables configuration for Red Hat OpenStack Platform provider cloud network routing services using an SDN-based virtual router.

- Routers are a requirement for your instances to communicate with external subnets, including those out in the physical network.
- Routers and subnets connect using interfaces, with each subnet requiring its own interface to the router.
- A router's default gateway defines the next hop for any traffic received by the router.
- A router's network is typically configured to route traffic to the external physical network using a virtual bridge.

Adding a Network Router

Add a network router to an existing cloud network by following the procedure below.

1. Browse to menu: **Networks > Network Routers**.
2. Click **Configuration** and click **Add a new Router**.
3. In the **Network Provider** area, select a **Network Manager**.
4. Provide a **Router Name**.
5. Under **External Gateway**:
    1. If **Yes** is selected, provide the following:

1. Choose to **Enable Source NAT**. In Source Network Address Translation (SNAT), he NAT router modifies the IP address of the sender in IP packets. SNAT is commonly used to enable hosts with private addresses to communicate with servers on the public Internet.
2. Select a **Network**.
3. Select a **Subnet**.

6. Select a **Cloud Tenant**.
7. Click **Add**.

Editing Network Router Details

To edit the details of a network router:

1. Browse to menu: **Networks > Network Routers**.
2. Select a network router from the list view.
3. Click **Configuration**, then **Edit selected Router**.
4. Edit required fields.
5. Click **Save**.

Adding an Interface to a Network Router

Interfaces allow you to interconnect routers with subnets. As a result, the router can direct any traffic that instances send to destinations outside of their intermediate subnet.

To add an interface to a network router:

1. Browse to menu: **Networks > Network Routers**.
2. Select a network router from the list view.
3. Click **Configuration**, then (**Add Interface to this Router**).
4. Select a **Subnet** from the list.
5. Click **Add**.

Removing a Network Router Interface

You can remove an interface to a subnet if you no longer require the router to direct its traffic.

To remove an interface:

1. Browse to menu: **Networks > Network Routers**.
2. Select a network router in the list view.
3. Click **Configuration**, then (**Remove Interface from this Router**).
4. Confirm your choice.

Deleting a Network Router

To delete a network router:

1. Browse to menu: **Networks > Network Routers**.
2. Select a network router from the list view.
3. Click **Configuration**, then click **Delete selected Routers**.

Creating Floating IPs

Floating IP addresses allow you to direct ingress network traffic to your cloud network instances. Define a pool of validly routable external IP addresses, which can then be dynamically assigned to an instance. All incoming traffic destined for that floating IP is routed to the instance to which it has been assigned.

Note:

Red Hat OpenStack Networking allocates floating IP addresses to all projects (tenants) from the same IP ranges/CIDRs. As a result, every subnet of floating IPs is consumable by any and all projects. Manage this behavior using quotas for specific projects.

Adding Floating IPs.

Floating IP addresses allow you to direct ingress network traffic to your instances.

Add floating IP addresses to an existing cloud network by following the procedure below.

1. Browse to menu: **Networks > Floating IPs**.

2. Click **Configuration** and click Add a new Floating IP.
3. Select a **Network Manager**.
4. Choose an **External Network**
5. Under **Association Information** provide the following:
    1. (Optional) An **Associated Port** for the floating IP.
    2. (Optional) The **Floating IP Address**.
6. Select a **Cloud Tenant**.
7. Click **Add**.

Managing Port Association of a Floating IP

To manage the port associations of a floating IP:

1. Browse to menu: **Networks > Floating IPs**.
2. Click on a floating IP from the list view.
3. Click **Configuration**, then **Manage the port association of this Floating IP**.
4. To associate a port, add a new **Port id**
5. To disassociate a port, delete the **Port id** field information.
6. Click **Save**.

Deleting a Floating IP

To delete a floating IP

1. Browse to menu: **Networks > Floating IPs**.
2. SClick on a floating IP from the list view to view its summary page.
3. Click **Configuration**, then click **Delete this Floating IP**.

Security Groups

You can group instances using security groups to restrict port or IP address accessibility. Security groups can be created and assigned to instances using Infrastructure Automation instance provisioning.

Cloud providers that currently support this function include: Amazon EC2, OpenStack, and Red Hat Enterprise Virtualization.

Editing Security Group Details

Editing security group information allows users to make changes to existing security group details and firewall rules, in additional to adding new firewall rules.

To edit details of a security group:

1. Browse to menu: **Networks > Security Groups**.
2. Click on a security group to view the summary page.
3. Click **Configuration**, then **Edit this Security Group**.
4. Under **Security Group Information**, edit the **Security Group Name** and the **Security Group Description**.
5. Edit existing **Firewall Rules** or add new firewall rules by clicking **Add a Firewall Rule**.
6. Click **Save**.

Viewing Security Groups

This procedure describes how to view security groups.

1. Browse to menu: **Networks > Security Groups**.
2. Click the desired security groups for viewing the details.
    - In **Properties**, you can view the basic information of the security group.
    - In **Relationships**, you can view the cloud provider and the instances associated with the security group.
    - In **Firewall Rules**, you can view a list of ports and IP ranges that are accessible.

      Note:

      This box is not available if you have not set any rules for your security group.

Tagging Security Groups

Apply tags to security groups to categorize them.

1. Browse to menu: **Networks > Security Groups**.
2. Select the security group to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a customer tag to assign from the dropdown menu.
5. Select a value to assign.
6. Click **Save**.

Deleting a Security Group

To delete a security group:

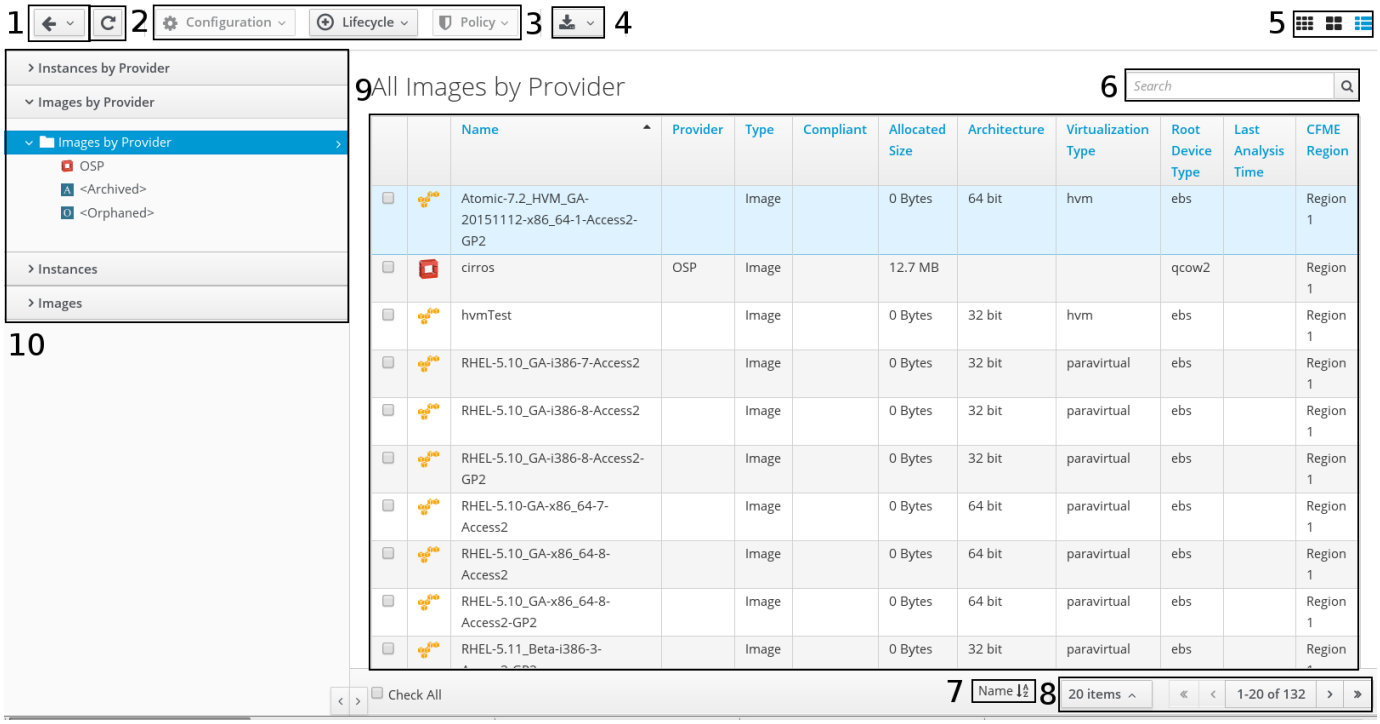
1. Browse to menu: **Networks > Security Groups**.
2. Click on a security group in the list view to view its summary page.
3. Click on **Configuration**, then click **Delete this Security Group**.

Instances and Images

Images are the static templates containing the software configuration from which you provision a running **Instance** - a virtual machine, with which you can interact - on your cloud provider.

The **Instance** and **Images** containers, combined with the ability to analyze information inside each instance or image, provides in-depth information across the cloud environment. This rich set of information enables Infrastructure Automation users to improve problem resolution times and effectively manage instances and images in their cloud environment.

The **Instances** and **Images** pages display all instances and images the server discovered from your cloud providers. The taskbar on each page is a menu driven set of buttons that provide access to functions related to instances and images.



1. History button
2. Refresh screen button
3. Taskbar
4. Download buttons
5. View buttons
6. Name search bar/Advanced Search button

7. Sort dropdown
8. Navigation bar
9. Main area in List View
10. Cloud/Filter Navigation

Console uses Virtual Thumbnails to describe instances and images. Each thumbnail contains four quadrants by default. This allows you to glance at an instance or image for a quick view of its contents.



1. Top left quadrant: Operating system of the Instance
2. Bottom left quadrant: Instance Cloud Provider
3. Top right quadrant: Power state of Instance or Status icon
4. Bottom right quadrant: Number of Snapshots for this Instance

Icon	Description
	Template: Cloud Image
	Retired: Instance has been retired
	Archived: Instance has no provider or availability zone associated with it.
	Orphaned: Instance has no availability zone but does have a provider associated with it.
	Disconnected: Instance is disconnected.
	On: Instance is powered on.
	Off: Instance is powered off.
	Suspended: Instance has been suspended.

The **Instances** page has four accordions organizing your instances and images in different ways. All of these accordions share a set of common controls:

- Use **Instances by Provider** and **Images by Provider** to view your instances and images organized by provider. In addition, you can see archived and orphaned items here.
- Use the **Instances** to view, apply filters, and collect information about all of your instances.
- Use **Images** to view, apply filters, and collect information about all of your images.

Through the console, you can view your instances and images in multiple ways:

- Filter instances
- Change views
- Sort
- Create a report
- Search by Tags
- Search by collected data

Filtering Instances and Images

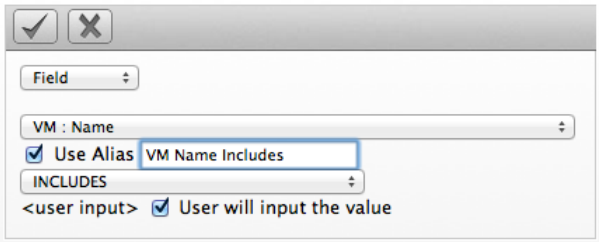
The **Instance Filter** accordion is provided so that you can easily navigate through groups of instances. You can use the ones provided or create your own through **Advanced Filtering** capabilities.

Using an Instance or Image Filter

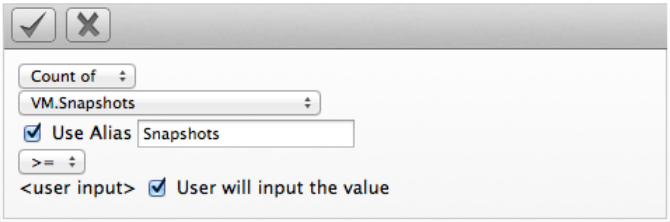
1. Browse to menu: **Compute > Clouds > Instances**.
2. Click on the **Instances** or **Images** accordion.
3. Click on the desired filter from the left pane.

Creating an Instance or Image Filter

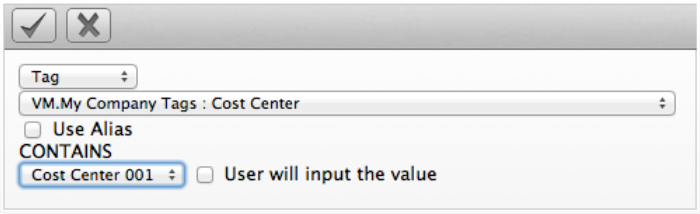
1. Browse to menu: **Compute > Clouds > Instances**.
2. Go to the **Instances** or **Images** accordion.
3. Click **All Instances** or **All Images**, then click (**Advanced Search**) to open the expression editor.
4. Use the expression editor to choose the appropriate options for your criteria. Based on what you choose, different options will show.
   - For all of the types of searches, you have the options of creating an alias and requested user input. Select **Use Alias** to create a user friendly name for the search. If you are requested user input for the search, this text will show in the dialog box where the input is requested.
   - Click **Field** to create criteria based on field values.

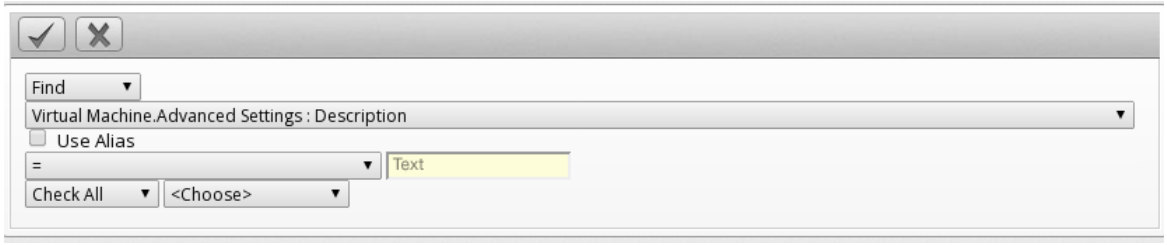
   - Click **Count of** to create criteria based on the count of something, such as the number of snapshots for an instance, or the number of instances on a host.

   - Click **Tag** to create criteria based on tags assigned to your virtual infrastructure, such as for power states or production tagging.

   - Click **Find** to seek a particular value, and then check a property.

5. Click (**Commit Expression Element Changes**) to add the expression.
6. Click **Save**.
7. Type in a name for the search expression in **Save this Instance search as**. To set the filter to show globally, check **Global Search**.
8. Click **Save**.

The filter is saved and shows in the **My Filters** area of the **Filter** accordion. If you checked **Global Search**, the filter shows there.

Loading a Report Filter or Search Expression

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to search (either **Instances** or **Images**).

3. Click (**Advanced Search**) to open the expression editor.
4. Click **Load**.
5. Select either a saved instance search or an instance report filter.

   Note:

   The set of items to select will depend on the type of resource you are searching.
6. Click **Load** to load the search expression.
7. If you want to edit the expression, click on it and make any edits for the current expression.
   - Click (**Commit expression element changes**) to add the changes.
   - Click (**Undo the previous change**) to remove the change you just made.
   - Click (**Redo the previous change**) to put the change that you just made back.
   - Click AND (**AND with a new expression element**) to create a logical AND with a new expression element.
   - Click OR (**OR with a new expression element**) to create a logical OR with a new expression element.
   - Click NOT (**Wrap this expression element with a NOT**) to create a logical NOT on an expression element or to exclude all the items that match the expression.
   - Click (**Remove this expression element**) to take out the current expression element.
8. Click **Load**.
9. Click **Apply**.

Changing Views for Instances and Images

While you can set the default view for different pages from the settings menu, then menu: **Configuration > My Settings > Default Views**, the current view can also be controlled from the Instances pages.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to view.
3. Click the appropriate button for the desired view.
   - Click for **Grid View**.
   - Click for **Tile View**.
   - Click for **List View**.

Sorting Instances and Images

Virtual machines and images can be sorted by Name, Availability Zone, Flavor, Cloud Provider, Compliant, Last Analysis Time, and Region.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the desired items to sort.
3. To sort instances or images when in grid or tile view:
   - From the **Sort by dropdown**, click the attribute to sort.
4. To sort instances or images when in list view:
   - Select the **List View**.
   - Click on the **Column Name** to sort. For example, click on **Availability Zone** to sort by the name of the availability zone.

Creating an Instance or Image Report

For a listing of instances and images, you can create a quick report in CSV, TXT, or PDF formats.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the desired items for report creation.
3. Click (**Download**).
   - Click for a **TXT** file.
   - Click for a **CSV** file.
   - Click for a **PDF** file.

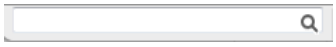
Searching for Instances or Images

To the right of the taskbar on the **Instances** page, you can enter names or parts of names for searching. You can search in the following ways.

- Type characters that are **included** in the name. For example, if you type **`sp1`**, all Instances whose names include **`sp1`** appear, such as **`Windows2003`** and **`Sp1clone`**.
- Use **`*`** at the end of a term to search for names that begin with specific characters. For example, type **`v*`** to find all instances whose names begin with the letter **`v`**.
- Use **`*`** at the beginning of a term to search for names that end with specific characters. For example, type **`*\sp2`** to find all instances whose names end with **`sp2`**.
- Erase all characters from the search box to go back to viewing all instances.

Search for instances and images:

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the desired items to search.

   
3. In the **Name Filter** bar in the upper right corner of the window, type your criteria.
4. Click (**Search by Name within results**) or press **Enter**.
5. Type in other criteria to filter on what is currently displayed.
6. Click (**Search by Name within results**) or press **Enter**.

Analyzing Instances and Images with SmartState Analysis

Analyze an instance to collect metadata such as user accounts, applications, software patches, and other internal information. If Infrastructure Automation is not set up for automatic analysis, perform a manual analysis of an instance. To perform a SmartState Analysis, Infrastructure Automation requires a running SmartProxy with visibility to the instance's storage location so that a snapshot can be created.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to analyze.
3. Check the instances and images to analyze.
4. Click **Configuration**, and then **Perform SmartState Analysis** on the taskbar.
5. Click **OK**.

Note:

Restrictions on Displaying Files Collected

- File size bigger than 20k characters
- File with missing name
- Non MIME .conf file, with non ascii characters
- Non MIME .conf file, without content
- MIME .exe binary file
- MIME .txt non binary file

- Non MIME .conf ascii file

SmartState Analysis for instances runs as a process independent from providers. For example, a successful SmartState Analysis of a host does not mean SmartState Analysis for instances will be successful. Ensure to enter credentials for the provider that contains the instance for the SmartState Analysis to work.

Comparing Instances and Images

You can compare multiple instances in Infrastructure Automation server. This allows you to see how different instances are from their original image. This helps detect missing patches, unmanaged user accounts, or unauthorized services.

Use the comparison feature to:

- Compare multiple instances from different hosts
- Compare multiple instances side-by-side
- Quickly see similarities and differences among multiple instances and a base
- Narrow the comparison display to categories of properties
- Print or export in the comparison results to a PDF or CSV file

Compare instances and images:

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to analyze.
3. Click the checkboxes for the items to compare.
4. Click **Configuration**, and then (**Compare Selected items**). The comparison displays in a compressed view with a limited set of properties listed.
5. To delete an item from the comparison, click (**Remove this VM from the comparison**) at the bottom of the items column.
6. To view many items on one screen, go to a compressed view by clicking (**Compressed View**). To return to an expanded view, click (**Expanded View**).
7. To limit the mode of the view, there are two buttons in the task bar.
   - Click (**Details Mode**) to see all details for an attribute.
   - Click (**Exists Mode**) to limit the view to if an attribute exists compared to the base or not. This only applies to attributes that can have a boolean property. For example, a user account exists or does not exist, or a piece of hardware that does or does not exist.
8. To change the base instance that all the others are compared to, click its label at the top of its column.
9. To go to the summary screen for an instance, click its **Virtual Thumbnail** or icon.

Creating an Instance Comparison Report

Output a the data from a comparison report in TXT, CSV or PDF formats.

1. Create the comparison for the report.
2. Click (**Download**).
   - Click for a **TXT** file.
   - Click for a **CSV** file.
   - Click for a **PDF** file.

Refreshing Instances and Images

Refresh your instances to get the latest data the provider can access. This includes information such as the power state, container, and hardware devices attached to the instance.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the desired items to analyze.
3. Click the checkboxes for the items to refresh.
4. Click **Configuration**, and then (**Refresh Relationships and Power States**) on the **Instance Taskbar**.

Extracting Running Processes from Instances and Images

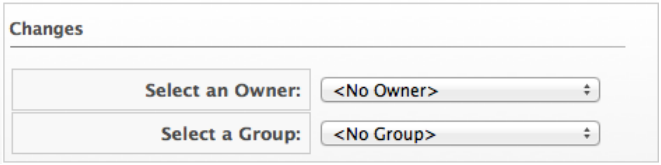
Infrastructure Automation can collect processes running on Windows instances. To do this, enter domain credentials for the zone where the instance is located. The instance must be running and must have an IP address in the VMDB, usually obtained from a SmartState Analysis.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the checkboxes for the instances to collect processes.
3. Click **Configuration**, and then (**Extract Running Processes**) on the taskbar.
4. Click **OK**.

Setting Ownership for Instances and Images

You can set the owner of a group of instances and images by either individual user or group. This allows you an additional way to filter and can be used to enforce quotas.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to change.
3. Click the checkboxes for the items to set ownership.
4. Click **Configuration**, and then (**Set Ownership**) on the **Instance Taskbar**.
5. From the **Select an Owner** dropdown, select a user.



6. From the **Select a Group** dropdown, select a group
7. Click **Save**.

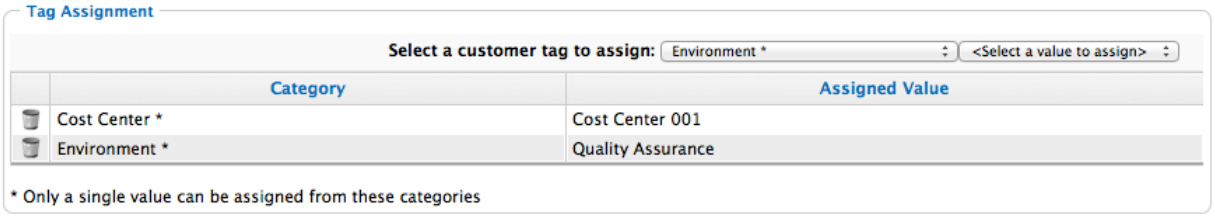
Removing Instances and Images from the VMDB

If an instance has been decommissioned or you need to perform some troubleshooting, you might need to remove a specific instance from the VMDB. This does not however remove the instance or image from its provider.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to remove.
3. Click the checkboxes for the items to remove.
4. Click **Configuration**, and then (**Remove from the VMDB**) button.
5. Click **OK**.

Tagging Instances and Images

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the items to tag.
3. Click the checkboxes for the items to tag.
4. Click (**Policy**), and then (**Edit Tags**).
5. Select a customer tag from the first dropdown, and then a value for the tag.



6. Click **Save**.

Reviewing an Instance or Image

After viewing your list of instances and images, click on a specific item to review a **Summary** screen of it. The **Summary** screen provides you with a **Virtual Thumbnail** and a **Taskbar**.

- Use the **Taskbar** to perform actions on the selected item.
- Use **Summary Views** to change the view type of the summary screen.
- Use **Virtual Thumbnails** for a quick glance at the item.
- Use the **Summary** screen to see a quick summary of the attributes of the item.

Viewing Running Processes after Collection

1. Click an instance with collected processes.
2. From the **Diagnostics** area, click **Running Processes**.

The most recent collection of running processes is displayed. Sort this list by clicking on the column headers.

Managing Security Groups for Cloud Instances

Manage security groups associated with cloud provider instances using Infrastructure Automation.

To add a security group to an instance:

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click on **Instances by Provider**, and select an instance.
3. Click **Configuration**, then click (**Add a Security Group**).
4. Select a **Security Group** from the drop-down menu to add to the instance.
5. Click **Save**.

To remove a security group from an instance:

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click on **Instances by Provider**, and select an instance.
3. Click **Configuration**, then click (**Remove a Security Group**).
4. Select a **Security Group** from the drop-down menu to remove from the instance.
5. Click **Save**.

Editing Instance or Image Properties

Edit the properties of an instance or image to set parent and child instances. SmartState Analysis also can detect this.

1. From **Compute > Clouds > Instances**.
2. Click the accordion for the items to edit.
3. Click the item to edit properties.

4. Click (**Configuration**), and then (**Edit this Instance or Edit this Image**) on the Taskbar.
5. From the **Parent Instance** dropdown, select the parent instance.
6. From **Child Instance** selection, select instances that are based on the current instance from the list of **Available Instances**.
7. Click **Save**.

Controlling the Power State of an Instance

Follow this procedure to control the power states of an instance through the Infrastructure Automation console.

1. Navigate to **Compute > Clouds > Instances**.
2. Click the instance to change the power state.
3. Click **Power Operations**, then click the button for the desired power operation.
   - Click (**Start**) to start the selected instances.
   - Click (**Terminate**) to terminate the selected instances.
   - Click (**Suspend**) to suspend the selected instances.
   - Click (**Reset**) to reset the selected instances.
   - Click (**Stop Guest**) to stop the guest operating system.
   - Click (**Restart Guest**) to restart the guest operating system.
4. Click **OK**.

Right Sizing an Instance

Infrastructure Automation uses collected statistics to recommend the best size for an instance. Infrastructure Automation uses the information from the **Normal Operating Range** to calculate the recommendations.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click an instance for right-sizing.
3. Click **Configuration**, and then (**Right-Size Recommendations**) button.

A new page appears with three levels of Memory and CPU recommendations, Conservative, Moderate, and Aggressive, next to the Normal Operating Range statistics.

Resizing an Instance

Infrastructure Automation allows you to resize an existing, active instance by changing its flavor. This is only possible if your OpenStack deployment has:

- At least two Compute nodes, or with resizing to the same host enabled
- Enough capacity to support the needs of the new flavor

Note:

Keep in mind that the instance will undergo a controlled shutdown when you change its flavor.

For more information about the requirements and underlying OpenStack process involved, see Resize an Instance in the Red Hat OpenStack Platform *Instances and Images Guide*.

To resize an instance through Infrastructure Automation:

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the instance whose flavor you want to change.
3. Click **Configuration**, and then (**Reconfigure this Instance**).
4. In the **Reconfigure Instance** section, select the new flavor you want from the **Choose Flavor** dropdown.
5. Click **Submit**. Doing so will initiate the flavor change, and it might take several minutes before Infrastructure Automation verifies whether the change was successful.

See Flavors and Manage Flavors in the Red Hat OpenStack Platform *Instances and Images Guide* for more information.

Migrating a Live Instance

Live migration involves moving a live instance between Compute nodes. Live migration is useful for avoiding instance downtime during cloud maintenance or load management. See How to Migrate a Live Instance in the Red Hat OpenStack Platform *Migrating Instances* guide for details on the underlying OpenStack process.

To migrate a live instance:

1. Browse to menu: **Compute > Clouds > Instances**.
2. On the right pane, click the instance to be migrated. Use the **Instances by Provider** accordion to filter instances by provider and/or availability zone.
3. Click (**Lifecycle**), then (**Migrate selected Instance**).
4. On the **Migrate Instance** section, select your preferred migration options:
   - **Auto-select Host?**: let the OpenStack provider automatically choose a destination Compute node. If you prefer to choose a specific node, uncheck this option and choose from the **Destination Host** dropdown.
   - **Block Migration**: check this option to perform a *block-based* migration. With this migration, the entire virtual machine image is moved from the source node to the destination node. If your OpenStack provider uses *shared storage*, leave this option unchecked. See Prerequisites in the Red Hat OpenStack Platform *Migrating Instances* guide for related information.
   - **Disk Over Commit**: check this option to prevent the OpenStack provider from verifying first whether the destination host has available disk space to host the instance.
5. Click **Submit**.

Once the migration initiates, the instance list will reload with a message indicating that the selected instance is being migrated. Upon completion, the instance list will reload and the evacuated instance will be displayed as (**On**).

Evacuating an Instance

If a Compute node is shut down, you can *evacuate* instances hosted on it. This is only useful if the instances use shared storage or block storage volumes. See Evacuate Instances in the Red Hat OpenStack Platform *Instances and Images Guide* for details on the underlying OpenStack process.

To evacuate an instance:

1. Browse to menu: **Compute > Clouds > Instances**.
2. On the right pane, click the instance to be evacuated. Use the **Instances by Provider** accordion to filter instances by provider and/or availability zone.
3. Click (**Lifecycle**), then (**Evacuate selected Instance**).
4. On the **Evacuate Host** section, select your preferred evacuation options:
   - **Auto-select Host?**: let the OpenStack provider automatically choose a destination Compute node. If you prefer to choose a specific node, uncheck this option and choose from the **Destination Host** dropdown.
   - **On Shared Storage**: leave this checked to indicate that all instance files are on shared storage.
5. Click **Submit**.

Once the evacuation initiates, the instance list will reload with a message indicating that the selected instance is being evacuated. Upon completion, the instance list will reload and the evacuated instance will be displayed as (**On**).

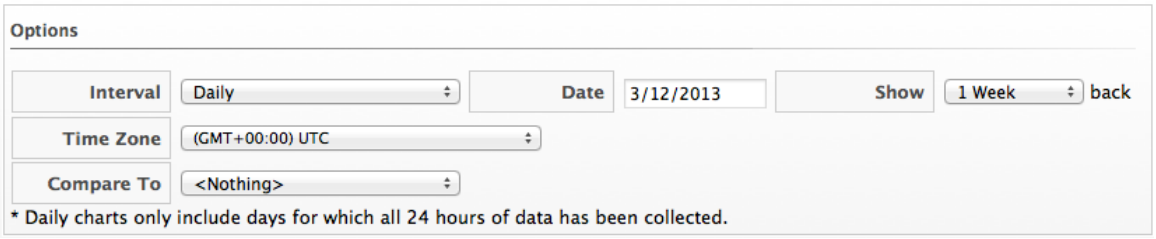
Viewing Capacity and Utilization Charts for an Instance

View capacity and utilization data for instances that are part of a cluster.

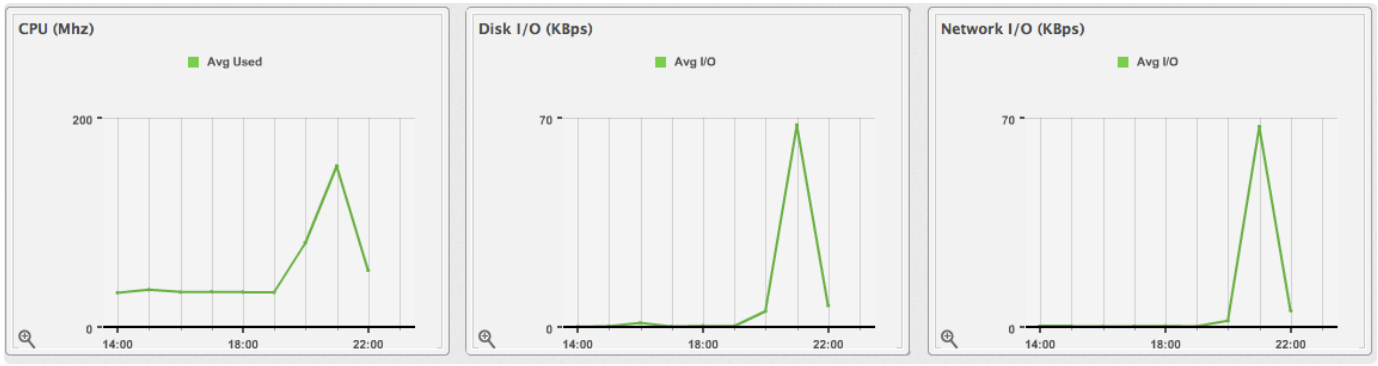
Note:

You must have a server with network visibility to your provider assigned the server role of **Capacity & Utilization Collector** to use this feature.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion to view capacity data.
3. Click the item to view.
4. Click (**Monitoring**), and then (**Utilization**) on the taskbar.
5. Select to view hourly, most recent hour, or daily data points for the dates to view data.



6. Select a **Time Profile**.



Note:

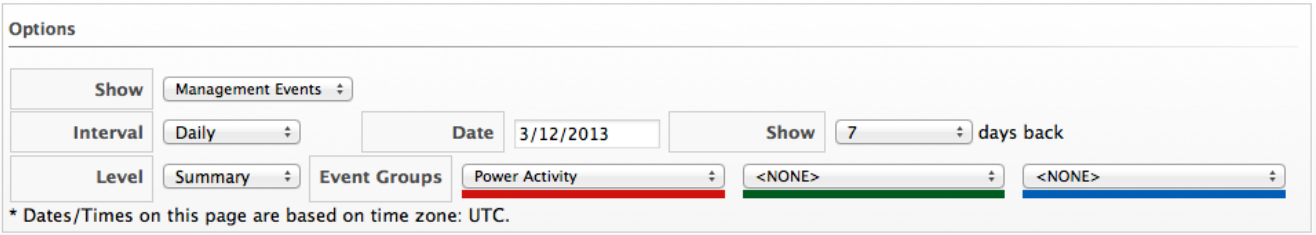
Daily charts only include full days of data. This means Infrastructure Automation does not show daily data for a day without a complete 24 data point range for a day.

For information about data optimization including utilization trend reports, see Data Optimization.

Viewing the Instance or Image Timeline

View the timeline of events for an instance or image if registered to a host.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the instance to view the timeline.
3. Click (**Monitoring**), and then (**Timelines**) on the taskbar.
4. From **Options**, customize the period of time to display, and the types of events to view.



- Use the **Interval** dropdown to select hourly or daily data points.
- Use **Date** to type the date of the timeline to display.
- If viewing a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days.
- The three **Event Group** dropdowns allow selection of different event groups to display. Each has its own color.
- From the **Level** dropdown, select either a **Summary** event or a **Detail** list of events. For example, the detail level of a **Power On** event might include the power on request, the starting event, and the actual **Power On** event. If you select **Summary**, you only see the **Power On** event in the timeline.

5. To see more detail on an item in the timeline, click on it. A balloon appears with a clickable link to the resource.

Viewing the Instance or Image Summary

When you click on a specific instance or image, you will see the **Virtual Thumbnail**, and an operating system-specific summary screen of the item. Where applicable, click on a subcategory of the summary to see more detail on that section.

The summary page contains the following categories:

- **Properties** include information such as the base operating system, hostname, IP addresses, instance vendor, cloud resources, and snapshots. This includes the ability to analyze multiple partitions, multiple disks, Linux logical volumes, extended partitions, and Windows drives. Some categories can be clicked on for additional detail. For example, click **Container** to view notes associated with an instance.
- **Lifecycle** shows the date of discovery and the last analysis. If a retirement date and time or owner has been set, these display as well.
- **Relationships** include information on the instance's cloud provider, genealogy such as parent and child instances, and drift.
- **VMsafe** shows properties of the VMsafe agent if it is enabled.
- **Compliance** shows the status of system compliance checks and history of past checks.
- **Power Management** displays the current power state, last boot time, and last power state change. **State Changed On** is the date that the instance last changed its power state. This is a container view of the power state, therefore a restart of the operating system does not cause the container power state to change and does not update this value.
- **Security** includes information on users and groups.
- **Configuration** includes information on applications, services, packages, init processes, and files. This section changes depending on the base operating system.
- **Diagnostics** provides a link to viewing running processes and the information from the latest collected event logs.
- **Smart Management** shows all tags assigned to this instance.

Performing a SmartState Analysis on an instance or image provides more detailed information in these categories.

Viewing User Information for an Instance or Image

Infrastructure Automation's **SmartState Analysis** feature returns user information. Explore the user to get details on the user's account, including group memberships.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click on the instance or image to open its summary page.
3. From the **Security** section of the summary page, click **Users**.
4. Click the user to view details.

Viewing Group Information for an Instance or Image

Infrastructure Automation's **SmartState Analysis** feature returns group information. Explore the group to get a list of its users.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the item to view user information.
3. Click on the item to view its **Summary**.
4. From the **Security** section of the **Instance Summary**, click **Groups**.
5. Click the group to view users.

Viewing Genealogy of an Instance or Image

Infrastructure Automation detects the lineage of an instance. View an instance's lineage and compare the instances that are part of its tree. This also allows tagging of instances that share genealogy.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the item to view genealogy.
3. Click on the item to view its **Summary**.
4. From the **Relationships** area in the **Summary**, click **Genealogy**.

Detecting Drift on Instances or Images

The configuration of an instance might change over time. **Drift** is the comparison of an instance to itself at different points in time. The instance needs to be analyzed at least twice to collect this information. Detecting drift provides you the following benefits:

- See the difference between the last known state of a machine and its current state.
- Review the configuration changes that happen to a particular instance between multiple points in time.
- Review the association changes that happen to a particular instance between multiple points in time.
- Review the classification changes that happen to an instance between two time checks.
- Capture the configuration drifts for a single instance across a time period.

Detect drift on instances or images:

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the item to view drift.
3. Click on the item to view its **Summary**.
4. From the **Relationships** area in the **Summary**, click **Drift History**.
5. Click the checkboxes for the analyses to compare.
6. Click  (**Select up to 10 timestamps for Drift Analysis**)] at the top of the screen. The results display.
7. Check the **Drift** sections on the left to view in your comparison.
8. Click **Apply**.
9. The following descriptions pertain to the **Expanded View**  . Whether you see the value of a property or an icon representing the property depends on the properties type.
    - A property displayed in the same color as the base means the compared analysis matches the base for that property.
    - A property displayed in a different color from the base means the compared analysis does not match the base for that property.
10. If you are in the **Compressed View**  , the values of the properties are not displayed. All items are described by the icons shown below.
    - A  (**checkmark**) means that the compared analysis matches the base for that property. If you hover over it, the value of the property will display.
    - A  (**triangle**) means the compared analysis does not match the base for that property. If you hover over it, the value of the property displays. Click the minus sign next to the sections name to collapse it.
11. To limit the scope of the view, you have three buttons in the **Resource** button area.
    - Click  (**All attributes**) to see all attributes of the sections you selected.
    - Click  (**Attributes with different values**) to see only the attributes that are different across the drifts.
    - Click  (**Attributes with the same values**) to see only the attributes that are the same across drifts.
12. To limit the mode of the view, there are two buttons in the **Resource** button area.
    - Click  (**Details Mode**) to see all details for an attribute.
    - Click  (**Exists Mode**) to only see if an attribute exists compared to the base or not. This only applies to attributes that can have a Boolean property. For example, a user account exists or does not exist, or a piece of hardware that does or does not exist.

This creates a drift analysis. Download the data or create a report from your drift for analysis using external tools.

Creating a Drift Report for an Instance or Image

1. Create the comparison to analyze.
2. Click  (**Download**).
3. Click the output button for the type of report you want.
    - Click  (**Download drift report in text format**) for a text file.
    - Click  (**Download drift report in CSV format**) for a csv file.
    - Click  (**Download drift report in PDF format**) for a PDF file.

Viewing Analysis History for an Instance or Image

Each time a SmartState Analysis is performed on an instance, a record is created of the task. This information is accessed either from the instance accordion or the instance summary. Use this detail to find when the last analysis was completed and if it completed successfully. If the analysis resulted in an error, the error is shown here.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the desired item to view analysis history.
3. Click on the item to view its summary.
4. From the **Relationships** area in the summary, click **Analysis History**. A history of up to the last 10 analyses is displayed.

Descending by: Started Per page: 20 (It

	Started ▼	Finished	Status	Message
🔍	2 Days Ago	2 Days Ago	OK	OK
🔍	3 Days Ago	3 Days Ago	OK	OK
🔍	4 Days Ago	4 Days Ago	OK	OK
🔍	5 Days Ago	5 Days Ago	OK	OK
🔍	6 Days Ago	6 Days Ago	OK	OK
🔍	6 Days Ago	6 Days Ago	OK	OK

5. Click on a specific analysis to see its details.

Viewing Event Logs for an Instance or Image

Using an **Analysis Profile**, collect event log information from your instances.

Note:

This feature is only available for Windows.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click the accordion for the item to view event logs.
3. Click on the item to view its **Summary**.
4. From **Diagnostics** click **Event Logs**.

The collected event log entries are displayed. Sort this list by clicking on the column headers.

Orchestration Stacks

Browse to menu: **Compute > Clouds > Stacks** to see a list of orchestration stacks along with information such as *Name*, *Provider*, *Type*, *Status*, *Instances*, *Security Groups*, and *Cloud Networks*. Click on a stack to see more information about it, including properties, retirement date and time, and relationships with the cloud provider and instances. You can click on instances to see details of all instances the stack relates to.

Tagging Orchestration Stacks

Apply tags to orchestration stacks to categorize them together at the same time. . Browse to menu: **Compute > Clouds > Stacks**. . Select the orchestration stacks to tag. . Click (**Policy**), and then (**Edit Tags**). . **Select a customer tag to assign** from the first list. . Select a value to assign from the second list. . Click **Save**.

Retiring Orchestration Stacks

You can either retire orchestration stacks on a set date and time, or retire them immediately.

To set a retirement date:

1. Browse to menu: **Compute > Clouds > Stacks**.

2. Select the orchestration stacks to retire on a set date and time.
3. Click (**Lifecycle**), and then (**Set Retirement Dates**).
4. On the **Retire Orchestration Stacks** page, set **Retirement Date and Time**.
5. Select **Retirement Warning** from the list. The default is *None*.
6. Click **Save**.

Note:

Saving a blank date will remove all retirement dates.

To retire selected stacks immediately:

1. Browse to menu: **Compute > Clouds > Stacks**.
2. Select the orchestration stacks to retire now.
3. Click (**Lifecycle**), and then (**Retire selected Orchestration Stacks**). A pop-up window appears to confirm the action.
4. Click **OK**.

Removing Orchestration Stacks

1. Browse to menu: **Compute > Clouds > Stacks**.
2. Select the orchestration stacks to remove from the VMDB. A warning pop-up window appears to confirm the action.
3. Click **Configuration**, and then (**Remove Stacks from the VMDB**).
4. Click **OK**.

Key Pairs

This tab lists key pairs and fingerprints for all cloud providers. Click on a key pair to see a summary and its relationship with instances. On this screen, click on instances to see details of all instances the key pair relates to. You can use the key pairs added during provisioning instances.

Note:

Adding a new key pair is currently only supported for OpenStack.

Adding a New Key Pair

1. Browse to menu: **Compute > Clouds > Key Pairs**.
2. Click **Configuration**, then click (**Add a new Key Pair**).
3. In **Basic Information**, enter a **Name** and the **Public Key (optional)** generated using **ssh-keygen** command.
4. Select your OpenStack provider from the **Provider** list.
5. Click **Add**.

Removing a Key Pair

1. Browse to menu: **Compute > Clouds > Key Pairs**.
2. Select the key pair you want to remove from the key pairs list. Or, click on the key pair to see the instances it relates to.
3. Click **Configuration**, then click (**Remove selected Key Pairs from Inventory**). A warning appears to confirm the action.
4. Click **OK**.

Downloading Key Pairs

1. Browse to menu: **Compute > Clouds > Key Pairs**. You will see a list of existing key pairs.
2. Click the  button, then select the option to download key pairs data in your preferred format:
    1. Download as Text
    2. Download as CSV
    3. Print or export as PDF

Object Stores

Browse to menu: **Storage > Object Stores** to see a list of all cloud object stores along with information including *Key*, *Size (bytes)*, *Object Count*, *Cloud Tenant*, and *Cloud Provider*. Click on an object store to see its properties and relationships with the cloud provider, tenant, and object store on the summary page.

Tagging Object Stores

Apply tags to object stores to categorize them together at the same time.

1. Browse to menu: **Storage > Object Stores**.
2. Select the object stores to tag.
3. Click **Policy**, and then **Edit Tags**.
4. **Select a customer tag to assign** from the first list.
5. Select a value to assign from the second list.
6. Click **Save**.

VMware Networking Switches

After adding a VMware provider, Infrastructure Automation automatically discovers all vSphere distributed switches (vDS) on that provider and collects the information in the Infrastructure Automation inventory.

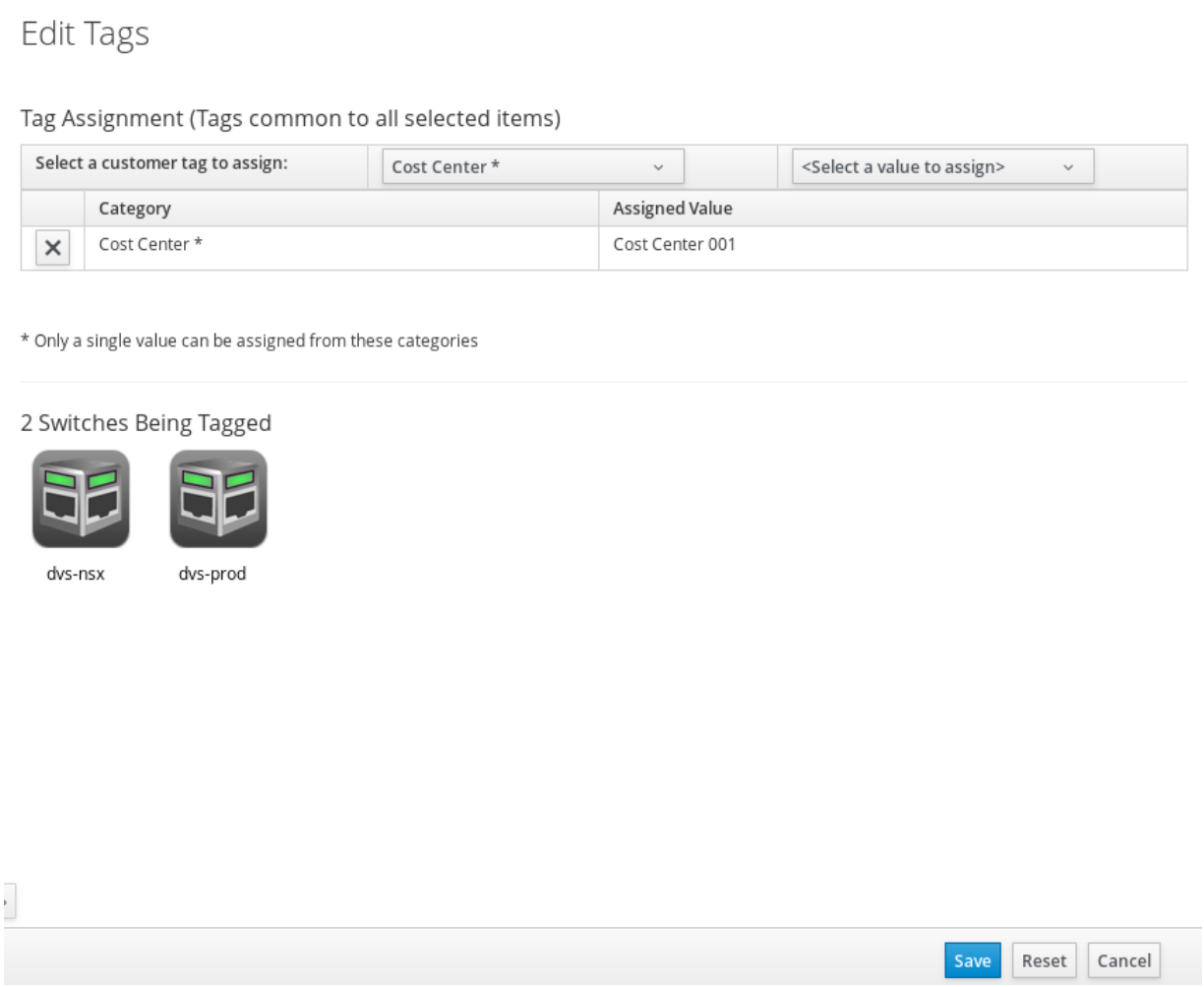
Browse to menu: **Compute > Infrastructure > Networking** to see a list of all VMware switches, along with information including *Name*, *Ports*, and *UUID*. Switches and port groups are listed by provider, then cluster on the sidebar.

Click on a switch to view its summary page, which displays relationships with hosts, and any tags.

Tagging VMware Networking Switches

Use tags to categorize vSphere distributed switches.

1. Browse to menu: **Compute > Infrastructure > Networking**.
2. Select the switches to tag.
3. Click (**Policy**), and then (**Edit Tags**).



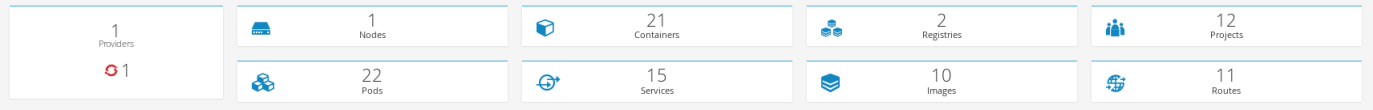
4. Select a customer tag from the first dropdown, and then a value for the tag.
5. Select more tags or click **Save** to apply the tags.

Container Entities

This chapter provides information on managing resources on your containers providers.

The Containers Overview Page

The containers overview page shows information on all containers providers and entities known to Infrastructure Automation. The **Overview** page provides links to other summary pages which contain further information on the containers providers and entities. The **Overview** page also provides metrics for **Aggregated Node Utilization**, **Network Utilization Trend**, **New Image Usage Trend**, **Node Utilization**, and **Pod Creation and Deletion Trends**.



1. Browse to menu: **Compute > Containers > Overview**.
2. Click the desired containers entity, or provider, if applicable, for viewing the summary with further information.

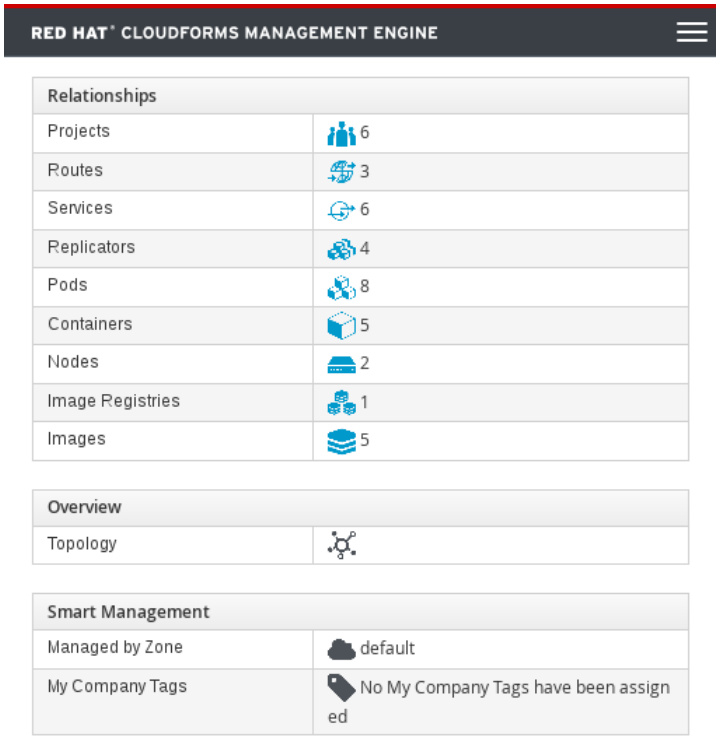
Note:

To reliably associate pods and images, Infrastructure Automation requires information from the `docker-pullable` field, added in OpenShift 3.3.1.2. This can affect the results of the **Chargeback by Image** report for older OpenShift providers, and potentially cause image inspection (done as part of Smart State Analysis) to fail due to associating a container to the wrong image. Consequently, Infrastructure Automation may not report accurate information about pods and images in OpenShift providers before version 3.3.1.2.

Viewing a Container Entity Summary

Container entity (object) summaries are found at menu: **Compute > Containers > Entity**, where you can view information about container entities and their components.

- Viewing a Containers Provider Summary Browse to menu: **Compute > Containers > Providers** to view information on different aspects of a containers provider. The summary includes:
    - The status of the provider and its components.
    - The relationships between different entities of the containers provider. These relationships are summarized in the **Relationships** box on the right-hand side of the summary page.



    - Additional information on aggregated capacity of all CPU cores of all nodes, and aggregated capacity of all memory of all nodes.
- Viewing a Container Nodes Summary Browse to menu: **Compute > Containers > Container Nodes** to view information on different aspects of a container node. The summary includes:
    - The number of entities on a node.
    - A node's capacity and utilization.
    - The version of the underlying operating system and software.

To view the timeline of events for a node from a container nodes summary page, click (**Monitoring**), and then (**Timelines**).

- Viewing a Containers Summary Browse to menu: **Compute > Containers > Containers** to view information on different aspects of a container. The summary includes:
    - The relationships of the container to a related node, pod, or image.
    - The node the container runs on.
    - The container ID.
    - Properties of the container image, such as name, tag, etc.
- Viewing a Container Images Summary Browse to menu: **Compute > Containers > Container Images** to view information on different aspects of a container image. The summary includes:
    - The containers currently using the images.
    - The image registry the image is from.
- Viewing an Image Registries Summary Browse to menu: **Compute > Containers > Image Registries** to view information on different aspects of an image registry. The summary includes:
    - Which images are from the registry.
    - The number of images that come from that registry.
    - Which containers use images from that registry.

    - The host and port of the registry.
- Viewing a Pods Summary Browse to menu: **Compute > Containers > Pods** to view information on different aspects of a pod. The summary includes:
    - The containers that are part of the pod.
    - The services that reference the pod.
    - The node the pod runs on.
    - If the pod controlled by a replicator.
    - The IP address of the pod.
- Viewing a Replicators Summary Browse to menu: **Compute > Containers > Replicators** to view information on different aspects of a replicator. The summary includes:
    - The number of requested pods.
    - The number of current pods.
    - The labels and selector for the replicator.
- Viewing a Container Services Summary Browse to menu: **Compute > Containers > Container Services** to view information on different aspects of a container service. The summary includes:
    - The pods that the container service provide traffic to.
    - The port configurations for the container service.
    - The labels and selector for the container service.
- Viewing a Volumes Summary Browse to menu: **Compute > Containers > Volumes** to view information on the persistent volumes of a container provider. The summary includes:
    - The pods the volume is connected to.
    - The volume's connection parameters.
    - The volume's storage capacity.
    - The volume's iSCSI target details (if applicable).
- Viewing a Container Builds Summary Browse to menu: **Compute > Containers > Container Builds** to view different aspects of a container build. The summary includes:
    - The build configuration the container build is based on.
    - Which build instances have been created.
    - Which phase in the build process the instance has completed.
    - Which pod a build instance reside in.
- Viewing a Container Templates Summary Browse to menu: **Compute > Containers > Container Templates** to view different aspects of a container template. The summary includes:
    - The project the template is associated with.
    - The objects the template contains.
    - The parameters that can be used with the template's objects.
    - The template's version number.

Using the Topology Widget

The **Topology** widget is an interactive topology graph, showing the status and relationships between the different entities of the containers providers and projects to which Infrastructure Automation has access.

- The topology graph includes pods, containers, services, nodes, virtual machines, hosts, routes, and replicators within the overall containers provider environment. - Each entity in the graph displays a color indication of its status.
- Hovering over any individual graph element will display a summary of details for the individual element.
- Double-click the entities in the graph to browse to their summary pages.
- It is possible to drag elements to reposition the graph.
- Click the legend at the top of the graph to show or hide entities.
- Click **Display Names** on the right-hand side of the page to show or hide entity names.

Viewing the Topology for Container Providers

1. Browse to menu: **Compute > Containers > Providers**.
2. Click the desired containers provider for viewing the provider summary.
3. On the provider summary page, click **Topology** in the **Overview** box on the right side of the page.

Viewing the Topology for Container Provider Projects

The project topology page displays the project as the center node, surrounded by its related entities.

1. Browse to menu: **Compute > Containers > Projects**.
2. Click on a project.
3. On the project summary page, click  (**Topology View**) on the top right side of the page.

Limiting the Number of Containers Shown in the Topology View

1. Navigate to the settings menu, then **My Settings**, and click on the **Visual** tab.
2. Select the number of container items from the drop-down under **Topology Default Items in View**.
3. Click **Save**.

Analyzing Container Images with SmartState Analysis

Perform a SmartState Analysis of a container image to inspect the packages included in an image.

1. Browse to menu: **Compute > Containers > Container Images**.
2. Check the container image to analyze. You can check multiple images.
3. Click **Configuration**, and then  **Perform SmartState Analysis**.

The container image is scanned. The process will copy over any required files for the image. After reloading the image page, all new or updated packages are listed.

To monitor the status of container image SmartState Analysis tasks, navigate to the settings menu, then **Tasks**. The status of each task is displayed including time started, time ended, what part of the task is currently running, and any errors encountered.

Note:

See *Scanning Container Images in with OpenSCAP* for details on scanning container images using OpenSCAP policies.

Configuring Automatic Tagging for Container Entities

Container object labels in OpenShift can be used to automatically create tags and tag categories in Infrastructure Automation. This is done by mapping Infrastructure Automation tags to existing OpenShift or Kubernetes labels.

Labels from OpenShift can be mapped to Infrastructure Automation tags for the following container entities:

- Projects
- Nodes
- Routes
- Replicators
- Container services
- Pods
- Container builds

Note:

Tags automatically created from OpenShift labels are completely managed by the Infrastructure Automation system and cannot be manually assigned or unassigned. Deleting a mapping rule from Infrastructure Automation immediately deletes the resulting tags.

You can view a container entity's OpenShift labels on the entity's details page under **Labels**.

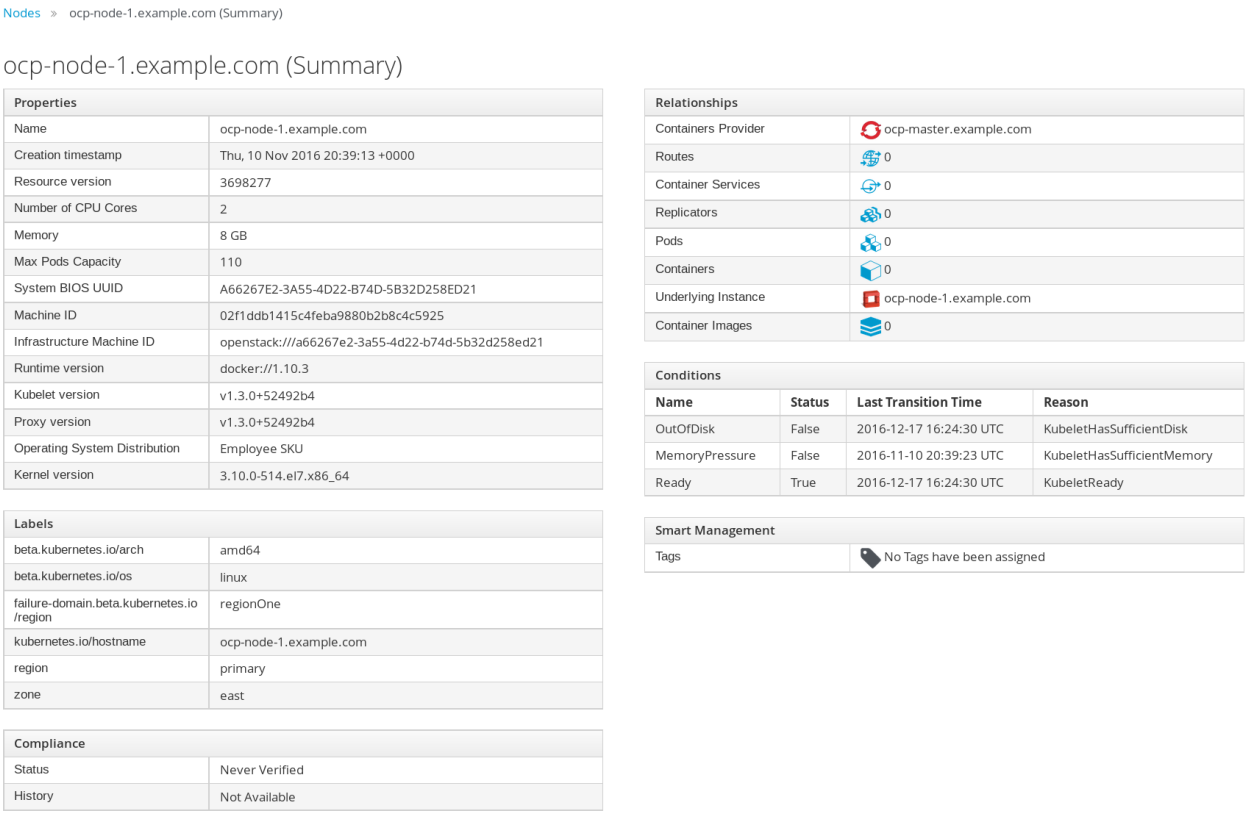
The following example shows how to configure tagging for a node, but the same steps can be used for mapping labels to tags on other container entities.

To configure automatic tagging on container entities using labels:

1. Note the *key* of the OpenShift label you want to map to a Infrastructure Automation tag. OpenShift labels consist of two parts: a *key* and a *value*.
    1. Browse to menu: **Compute > Containers > Container Nodes**.
    2. Select a node to open its summary page.

3. Under **Labels**, note the label(s) to map to Infrastructure Automation tag(s). Any OpenShift labels will list the *key* in the left column of the **Labels** table, and the *value* in the right column of the **Labels** table.

   This node has six labels (key/value pairs) that were created in OpenShift and collected in the Infrastructure Automation inventory:



Nodes » ocp-node-1.example.com (Summary)

ocp-node-1.example.com (Summary)

Properties	
Name	ocp-node-1.example.com
Creation timestamp	Thu, 10 Nov 2016 20:39:13 +0000
Resource version	3698277
Number of CPU Cores	2
Memory	8 GB
Max Pods Capacity	110
System BIOS UUID	A66267E2-3A55-4D22-B74D-5B32D258ED21
Machine ID	02f1ddb1415c4feba9880b2b8c4c5925
Infrastructure Machine ID	openstack:///a66267e2-3a55-4d22-b74d-5b32d258ed21
Runtime version	docker://1.10.3
Kubelet version	v1.3.0+52492b4
Proxy version	v1.3.0+52492b4
Operating System Distribution	Employee SKU
Kernel version	3.10.0-514.el7.x86_64

Relationships	
Containers Provider	ocp-master.example.com
Routes	0
Container Services	0
Replicators	0
Pods	0
Containers	0
Underlying Instance	ocp-node-1.example.com
Container Images	0

Conditions			
Name	**Status**	**Last Transition Time**	**Reason**
OutOfDisk	False	2016-12-17 16:24:30 UTC	KubeletHasSufficientDisk
MemoryPressure	False	2016-11-10 20:39:23 UTC	KubeletHasSufficientMemory
Ready	True	2016-12-17 16:24:30 UTC	KubeletReady

Labels	
beta.kubernetes.io/arch	amd64
beta.kubernetes.io/os	linux
failure-domain.beta.kubernetes.io /region	regionOne
kubernetes.io/hostname	ocp-node-1.example.com
region	primary
zone	east

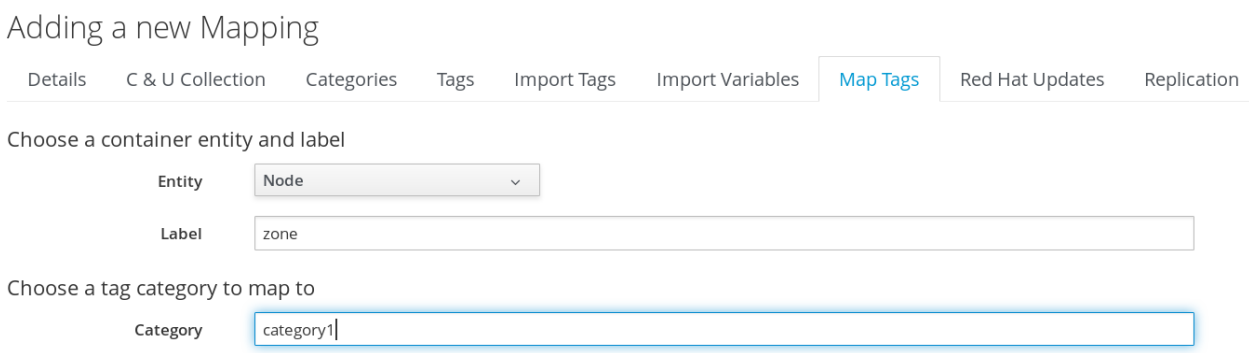
Smart Management	
Tags	No Tags have been assigned

Compliance	
Status	Never Verified
History	Not Available

Note:

To create an OpenShift label, see Developer CLI Operations in the OpenShift Container Platform *CLI Reference* guide. A new label added in OpenShift will only show up in Infrastructure Automation after the next OpenShift provider refresh.

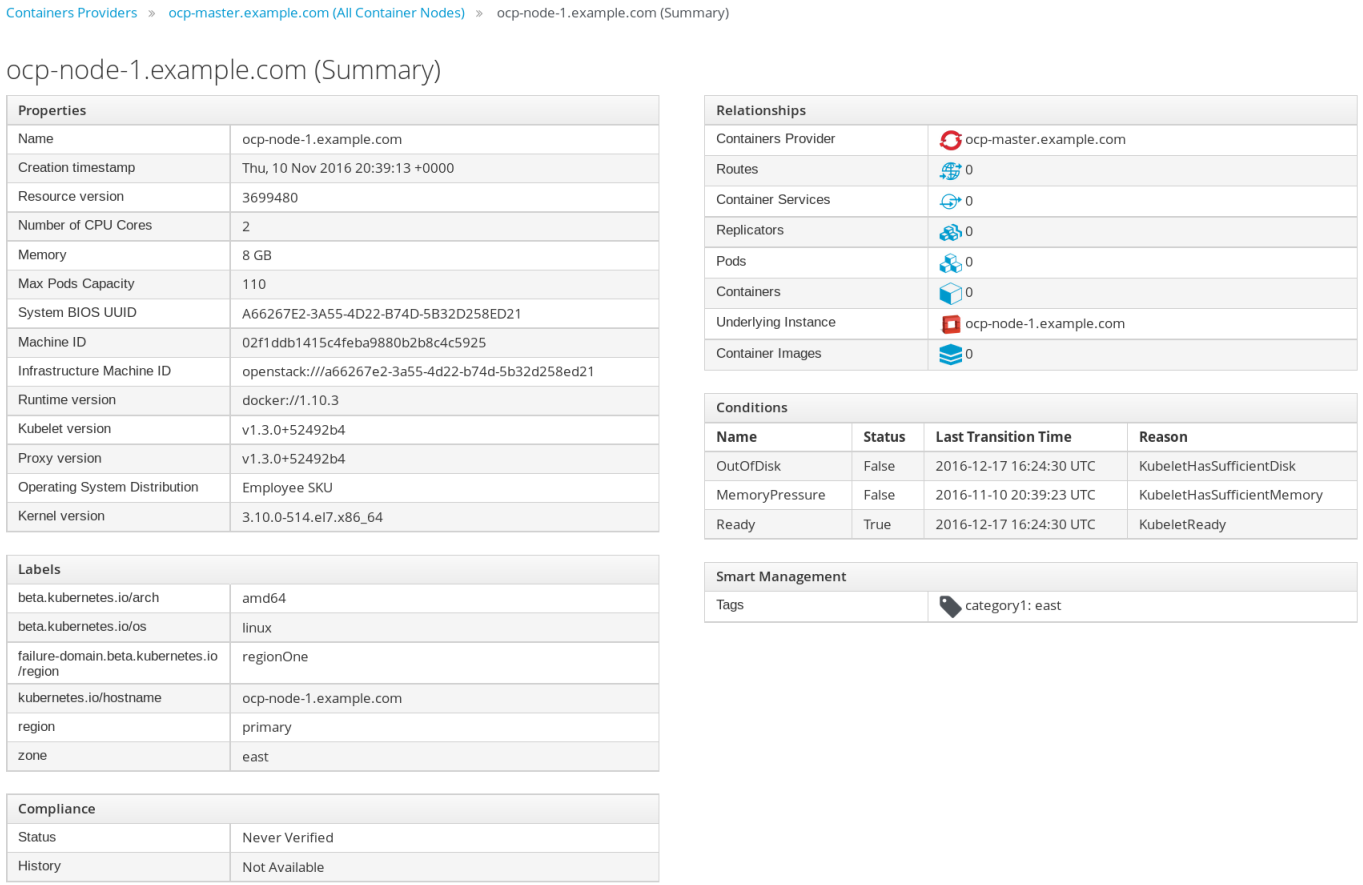
2. Navigate to **Configuration** and select the region.
3. Click the **Map Tags** tab.
4. Click **Add** to create a new mapping rule.
   1. Select a container entity to tag from the **Entity** list, or select **All** to tag all entities.
   2. Specify the *key* from the OpenShift label you noted earlier in the **Label** field.
   3. Specify a Infrastructure Automation tag category in **Category** to map the label to. If the tag category does not exist yet in Infrastructure Automation, it will be created automatically.



   4. Click **Add**. The mapping will show in the table on the **Map Tags** tab.
5. Refresh the provider to complete the mapping:

1. Browse to menu: **Compute > Containers > Providers**.
2. Select the provider to refresh.
3. Click **Configuration**, and then **Refresh Items and Relationships**.

The label will display on the entity's summary page under **Smart Management** under **Company Tags** as `<Category> : <value>`.



Any container entity with the OpenShift `zone` label will be tagged automatically as `category1` in Infrastructure Automation. If the *value* for `zone` is `south`, for example, the entity will be tagged as `category1
south`.

You can use these tags to create reports. See Monitoring, Alerts, and Reporting for details on creating reports.

- Managing Providers
- Infrastructure Providers
- Refreshing Providers
- Tagging Multiple Providers
- Viewing a Provider
- Removing a Provider
- Viewing the Provider Timeline
- Viewing Hosts and Clusters
- Viewing Virtual Machines and Templates
- Configuration Management Providers
- Automation Management Providers
  - Embedded Workflows
    - Embedded Workflows UI
    - Adding an Embedded Workflows Repository
    - Viewing the List of Workflows in a Repository
    - Refreshing Repositories
    - Adding Credentials
      - Credential Types
        - SCM
    - Authoring a workflow
      - Unsupported Amazon States Language features
      - Running an Embedded Workflow on Appliances
      - Running an Embedded Workflow on the OpenShift Container Platform (OCP)

            - Example: Provisioning Workflow
            - Credentials
        - Viewing workflow details
        - Creating a Service Dialog using an Embedded Workflow
        - Creating Service Catalog Items using an Embedded Workflow
    - Ansible
        - Enabling the Embedded Ansible Server Role
        - Verifying the Embedded Ansible Worker State
        - Adding a Playbook Repository
        - Refreshing Repositories
- Credentials
    - Adding Credentials
    - Credential Types
        - Machine
        - Network
        - SCM
        - Amazon
        - Azure
        - OpenStack
        - Red Hat Virtualization
        - VMware
- Tagging Ansible Playbooks, Repositories, and Credentials
    - Adding Tags to Ansible Playbooks
    - Adding Tags to Ansible Repositories
    - Adding Tags to Ansible Credentials
- Optimizing Ansible Playbooks for Infrastructure Automation
    - Installing Roles on an Embedded Ansible Appliance
    - Ansible Service Linking
        - Example: Linking a virtual machine to a service
        - Modifying the Automate Workspace Using the manageiq-automate Role.
            - Role Variables
            - Example Playbook
    - Callbacks in Multiple Appliance Environments
        - Ansible Tower
            - Working with an Ansible Tower Provider
            - Adding an Ansible Tower Provider
            - Refreshing an Ansible Tower Provider
            - Viewing Ansible Tower Providers and Inventory
            - Viewing Ansible Tower Configured Systems
            - Executing an Ansible Tower Job or Workflow Template from a Service Catalog
            - Executing an Ansible Tower Job Using a Custom Automate Button
    - Embedded Terraform (OpenTofu)
    - Importing OpenTofu image on an appliance
    - Usage of Embedded Terraform (OpenTofu)
        - Enabling the Embedded Terraform Server Role
        - Verifying the Embedded Terraform worker state
        - Adding a Template Repository
        - Refreshing Repositories
        - Credentials
            - Adding Credentials
            - Credential Types
                - SCM
                - Amazon
                - Azure
                - Google Compute Engine
                - IBM Cloud Classic Infrastructure
                - OpenStack
                - VMware
        - Running a Terraform Template from a Service Catalog
            - Create a catalog
            - Create a Terraform Service Catalog item
            - Run the Terraform Template:
            - Retire the Terraform services
- Cloud Providers
- Refreshing Cloud Providers
- Tagging Cloud Providers
- Removing Cloud Providers
- Editing a Cloud Provider
- Viewing a Cloud Provider’s Timeline
- Physical Infrastructure Providers

- Image Import Workflows
  - IBM PowerVC to IBM Power Systems Virtual Server
    - Preparing Infrastructure Automation hosting server (required once):
    - Preparing IBM PowerVC Server (required once):
    - Add a Cloud Object Storage provider in Infrastructure Automation:
    - Add an IBM Power Systems Virtual Server provider in Infrastructure Automation:
    - Add an IBM PowerVC provider in Infrastructure Automation:
    - Grant Image Import Permissions in Infrastructure Automation:
    - Enable Embedded Ansible in Infrastructure Automation:
    - Start the workflow
    - Troubleshooting
- Network Managers
- Adding or Viewing Network Providers
- Refreshing Network Providers
- Tagging Network Providers
- Removing Network Providers
- Viewing a Network Provider's Timeline
- Using the Topology Widget for Network Providers
- Containers Providers
- Tagging Containers Providers
- Removing Containers Providers
- Editing a Containers Provider
- Hiding Environment Variables for Containers Providers
- Viewing a Containers Provider's Timeline
- Storage Managers
  - Appendix
  - Using a Self-Signed CA Certificate

Managing Providers

Infrastructure Automation can manage a variety of external environments, which are known as providers and managers. A provider or manager is any system that Infrastructure Automation integrates with for the purpose of collecting data and performing operations.

In Infrastructure Automation, a *provider* is an external virtualization, cloud, or containers environment that manages multiple virtual machines or instances residing on multiple hosts. One example is Red Hat Virtualization, a platform that manages multiple hosts and virtual machines.

In Infrastructure Automation, a *manager* is an external management environment that manages more than one type of resource. One example of a manager is OpenStack, which manages infrastructure, cloud, network, and storage resources.

This guide covers working with providers and managers in Infrastructure Automation, which include:

- Infrastructure providers
- Configuration management providers
- Automation management providers
- Cloud providers
- Physical infrastructure providers
- Networking management providers
- Container providers
- Storage managers

For information on working with the resources that are contained by a provider or manager, see Managing Infrastructure and Inventory.

Infrastructure Providers

In Infrastructure Automation, an infrastructure provider is a virtual infrastructure environment that you can add to a Infrastructure Automation appliance to manage and interact with the resources in that environment. This chapter describes the different types of infrastructure providers that you can add to Infrastructure Automation, and how to manage them. Infrastructure providers can be either discovered automatically by Infrastructure Automation, or added individually.

The web interface uses virtual thumbnails to represent infrastructure providers. Each thumbnail contains four quadrants by default, which display basic information about each provider:



1. Number of hosts
2. Management system software
3. Currently unused
4. Authentication status

Icon	Description
	Validated: Valid authentication credentials have been added.
	Invalid: Authentication credentials are invalid.
	Unknown: Authentication status is unknown or no credentials have been entered.

For information on managing specific infrastructure providers see the following.

- Red Hat Virtualization Providers
- OpenStack Infrastructure Providers
- VMware vCenter Providers
- IBM Power HMC
- OpenShift Virtualization (KubeVirt)

Refreshing Providers

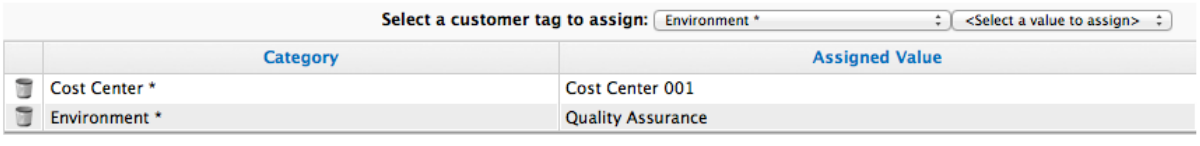
Refresh a provider to find other resources related to it. Use **Refresh** after initial discovery to get the latest data about the provider and the virtual machines it can access. Ensure the provider has credentials to do this. If the providers were added using **Discovery**, add credentials using (**Edit Selected Infrastructure Provider**).

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Select the providers to refresh.
3. Click **Configuration**, and then (**Refresh Relationships and Power States**).
4. Click **OK**.

Tagging Multiple Providers

Apply tags to all providers to categorize them together at the same time.

1. Navigate to **Infrastructure > Providers**.
2. Check the providers to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. In the **Tag Assignment** area, select a customer tag to assign from the first list, then select a value to assign from the second list.



5. Select more tags as required; click (**Save**).

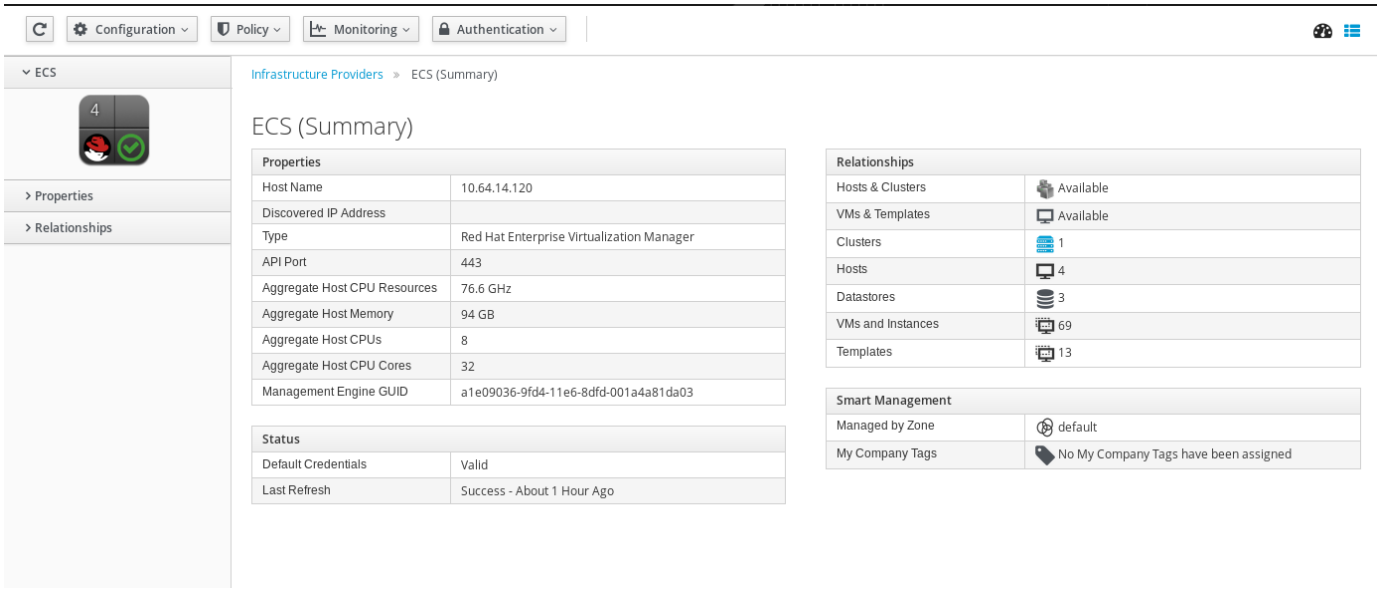
Viewing a Provider

From a list of providers, you can review a specific provider by clicking on it. This displays various options to access provider information.

There are two methods of viewing an infrastructure provider's details: the summary screen (default) and the dashboard screen. Use the summary and dashboard buttons to toggle between views.

Both the summary and dashboard screens contain a taskbar with **Reload**, **Configuration**, **Policy**, **Monitoring**, and **Authentication** buttons to manage the selected provider.

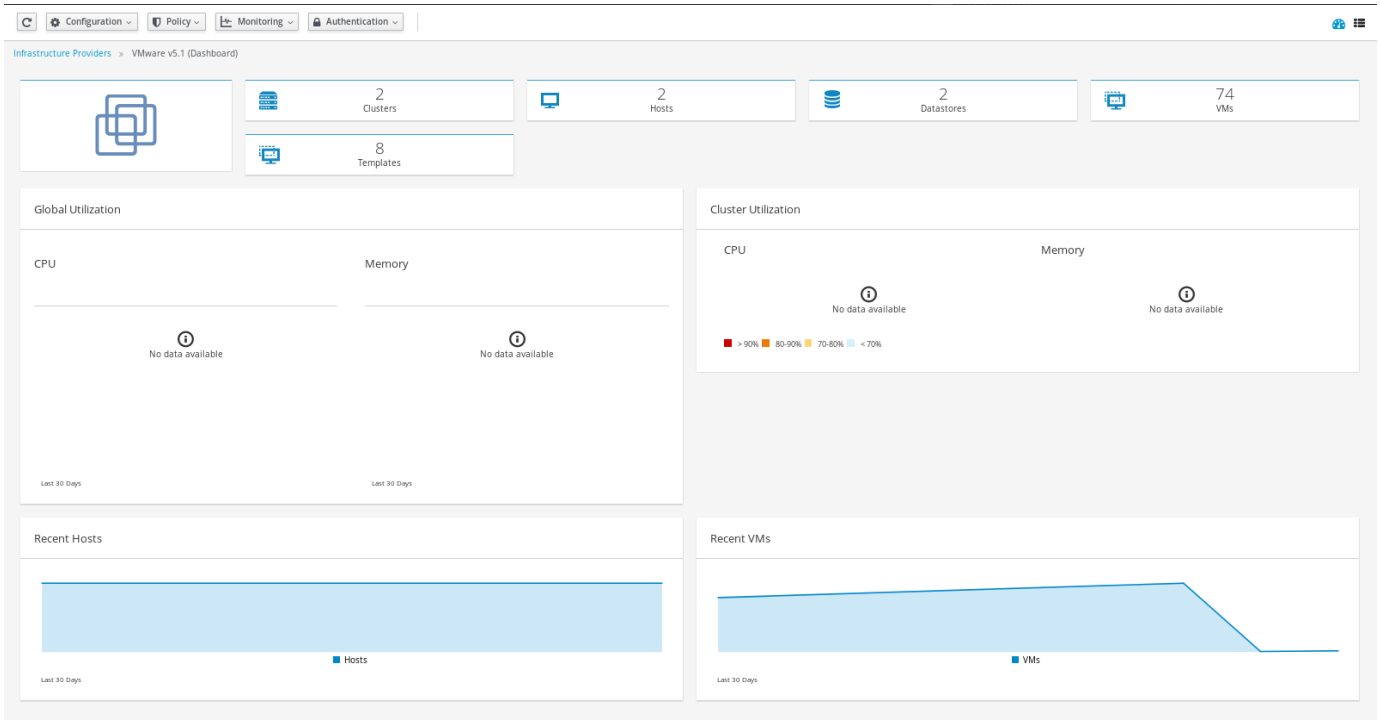
Provider Summary Screen.



The provider summary screen displays information about the provider in table format.

- Provider accordion: Displays details about the provider's **Properties** and **Relationships** on the sidebar. Click to expand these lists.
- Provider summary: Displays a provider's **Properties**, **Status**, **Relationships**, and **Smart Management**. Click on an item in the **Relationships** table to see more information about that entity.

Provider Dashboard Screen.



From the dashboard, you can view:

- Number of clusters, hosts, virtual machines, templates, datastores, resource pools, and other entities on the provider. Click on an entity to see more information about that item.
- Aggregate utilization for CPU, memory, and storage

- Network I/O statistics
- Trends for hosts and virtual machines discovered

To view the dashboard:

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click the infrastructure provider to view.
3. To access the dashboard view, click (**Dashboard view**).

To return to the summary view, click (**Summary view**).

Removing a Provider

If a provider has been decommissioned or requires some troubleshooting, it might require deletion from the VMDB.

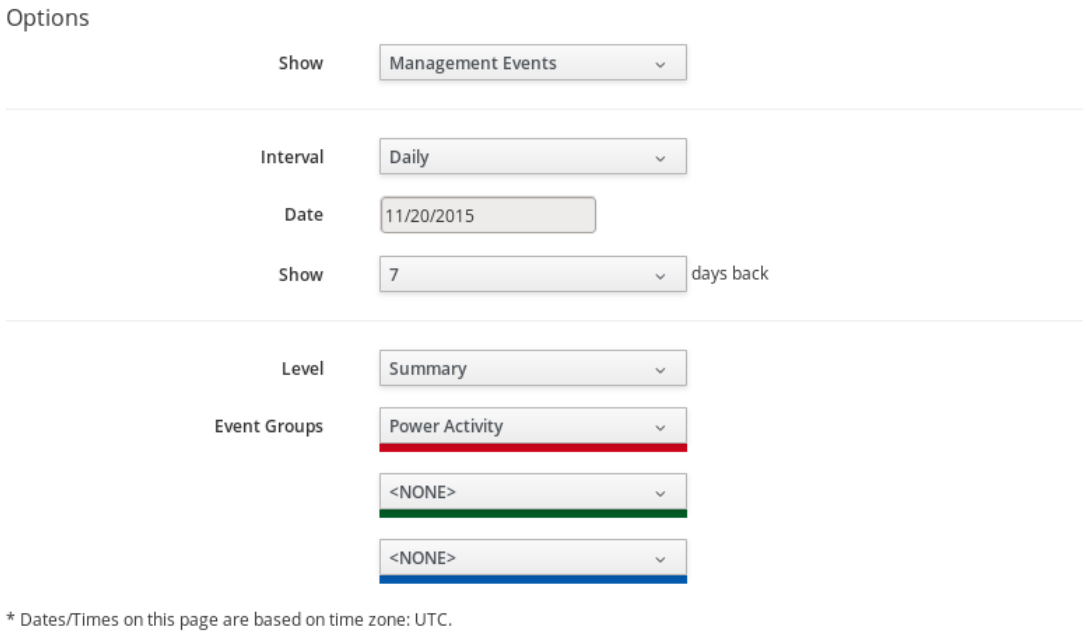
Deleting a provider removes the account information from Infrastructure Automation console. You will no longer be able to view any associated history including chargeback reports generated for the deleted provider. Additionally, if Infrastructure Automation is the database of record, deleting providers would become a major problem for the other systems relying on it for accurate and consistent billing information. Review all the dependencies carefully before deleting a provider.

1. Navigate to **Compute > Infrastructure > Providers**.
2. Select the check box for the provider to delete.
3. Click (**Configuration**), then (**Remove Infrastructure Providers from the VMDB**).
4. Click (**OK**).

Viewing the Provider Timeline

View the timeline of events for the virtual machines registered to a provider.

1. Navigate to **Compute > Infrastructure > Providers**.
2. Click a provider.
3. Click (**Monitoring**), and then (**Timelines**)from the taskbar, or from the provider accordion, click **Properties > Timeline**.
4. From **Options**, customize the period of time to display and the types of events to see.



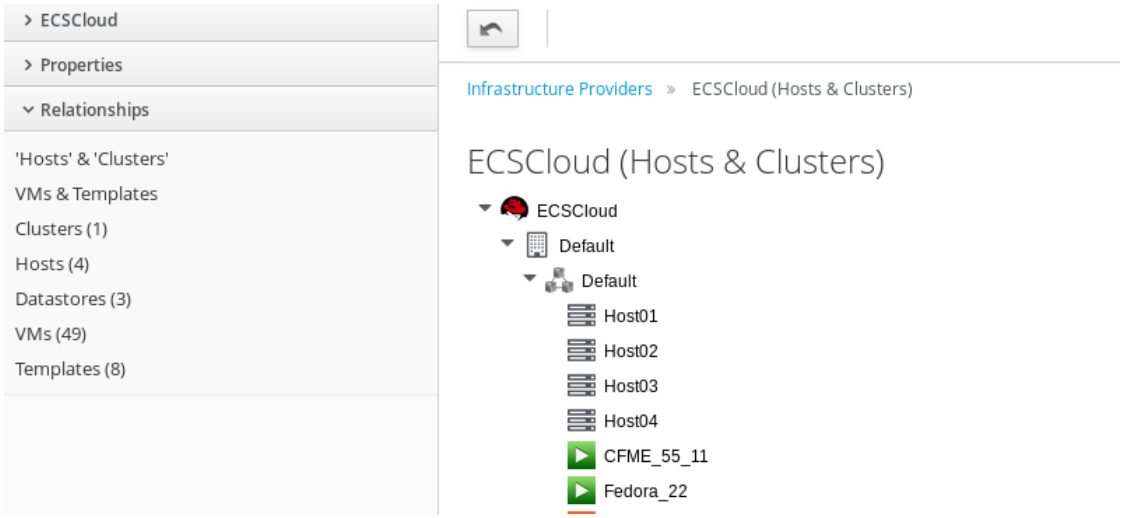
- Use **Show** to select regular Management Events or Policy Events.
- Use the **Interval** dropdown to select hourly or daily data points.
- Use **Date** to type the date for the timeline to display.

- If you select to view a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days.
- The three **Event Groups** lists allow you to select different groups of events to display. Each has its own color.
- From the **Level** list, select a **Summary** event, or a **Detail** list of events. For example, the detail level of a **Power On** event might include the power on request, the starting event, and the actual **Power On** event. If you select **Summary**, only the Power On event displays in the timeline.

Viewing Hosts and Clusters

Access a tree view of the hosts and clusters for a provider from the **Provider Summary**.

1. Navigate to **Compute > Infrastructure > Providers**.
2. Click the provider to view the hosts and clusters.
3. Click on the **Relationships** accordion, then click **Hosts & Clusters**.



Viewing Virtual Machines and Templates

Access a tree view of the virtual machines and templates for a provider from the **Provider Summary**.

1. Navigate to **Compute > Infrastructure > Providers**.
2. Click the provider to view the virtual machines and templates.
3. From accordion menu, click **Relationships**, then click **VMs & Templates**.

Configuration Management Providers

In Infrastructure Automation, a configuration management provider is a systems management product that you can add to a Infrastructure Automation appliance to manage the lifecycle of your resources. Configuration management providers are useful for uniformly applying changes and updates across providers, and for recording and reporting status and change activity. They can also help eliminate the confusion and error brought about by the existence of different providers.

This chapter describes the different types of configuration management providers available to Infrastructure Automation, and how to manage them. Configuration management providers must be added individually to Infrastructure Automation.

- IBM Terraform provider
- Red Hat Satellite 6

Automation Management Providers

In Infrastructure Automation, an automation management provider is a management tool that integrates with Infrastructure Automation to simplify automation operations for your resources. The following chapter describes the automation management providers that you can use with Infrastructure Automation, and how to work with them.

Infrastructure Automation provides automation management through the following features:

Automate enables real-time, bidirectional process integration. This embedded automate feature provides you with a method to implement adaptive automation for management events and administrative or operational activities.

Embedded Workflows provide the capability to simplify service catalog creation and improve the overall service catalog experience through a step-based automation system. Workflows can run docker containers to accomplish a task that the user requires as part of the state machine. Workflows are imported from Git repository content into Infrastructure Automation where they are made available for users to run. *Embedded Workflows* are an alternative to the *Embedded Automate* capability for authoring and running automation. With *Embedded Workflows*, you can continue to use the *Embedded Automate* feature for service catalog creation and management.

Ansible integration delivers out-of-the-box support for backing service, alert, and policy actions by using Ansible playbooks. Sync your existing playbook repositories with Infrastructure Automation, add credentials to access providers, and create service catalog items for actions ranging from creating and retiring VMs, updating security software, or adding more disks when space runs low.

Ansible Tower is a management tool that is integrated with Infrastructure Automation, designed to help automate infrastructure operations utlizizing existing Ansible Tower providers in your inventory. Infrastructure Automation allows you to execute Ansible Tower jobs by using service catalogs and Automate. Using Ansible Tower, you can schedule Ansible playbook runs and monitor current and historical results, allowing for troubleshooting or identification of issues before they occur.

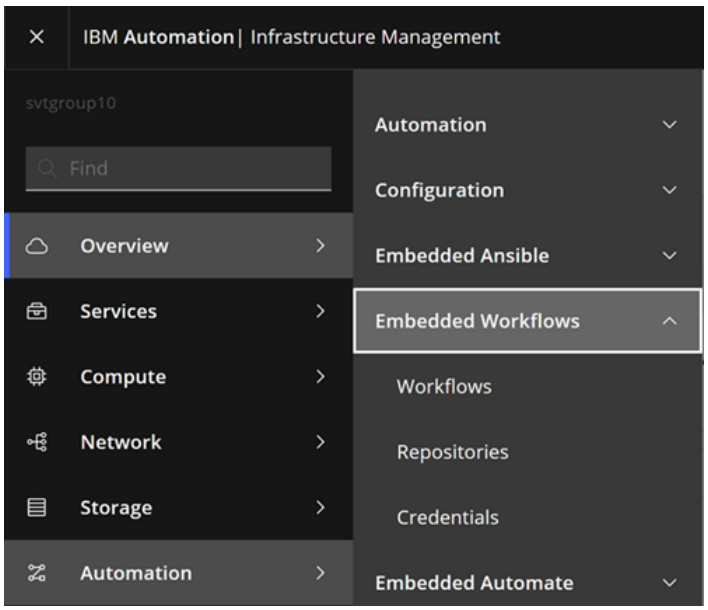
Embedded Workflows

Embedded Workflows provide the capability to simplify service catalog creation and improve the overall service catalog experience through a step-based automation system. Workflows can run docker containers to accomplish a task that the user requires as part of the state machine. Workflows are imported from Git repository content into Infrastructure Automation where they are made available for users to run.

Embedded Workflows are an alternative to the *Embedded Automate* capability for authoring and running automation. With *Embedded Workflows*, you can continue to use the *Embedded Automate* feature for service catalog creation and management.

Embedded Workflows UI

For the *Embedded Workflows* feature, the following pages are available in the UI, which you can use to access and work with workflows. To view these pages, go to **Automation** > **Embedded Workflows** and select the page.



- **Workflows**

  This UI page includes a summary of the available workflows. You can use different views to review the workflows:

  - Text view, which is the raw representation (JSON or YAML) of the Amazon State Languages (ASL) code.
  - Graph view, which shows a visual representation of the workflow structure.
- **Repositories**

  This page lists, and provides a summary of, the added embedded workflow repositories. From this page, you can add, update, and remove repositories. When a repository is added and synchronized, any workflows that are found in the repository can be viewed on the **Workflows** UI page.

- **Credentials**

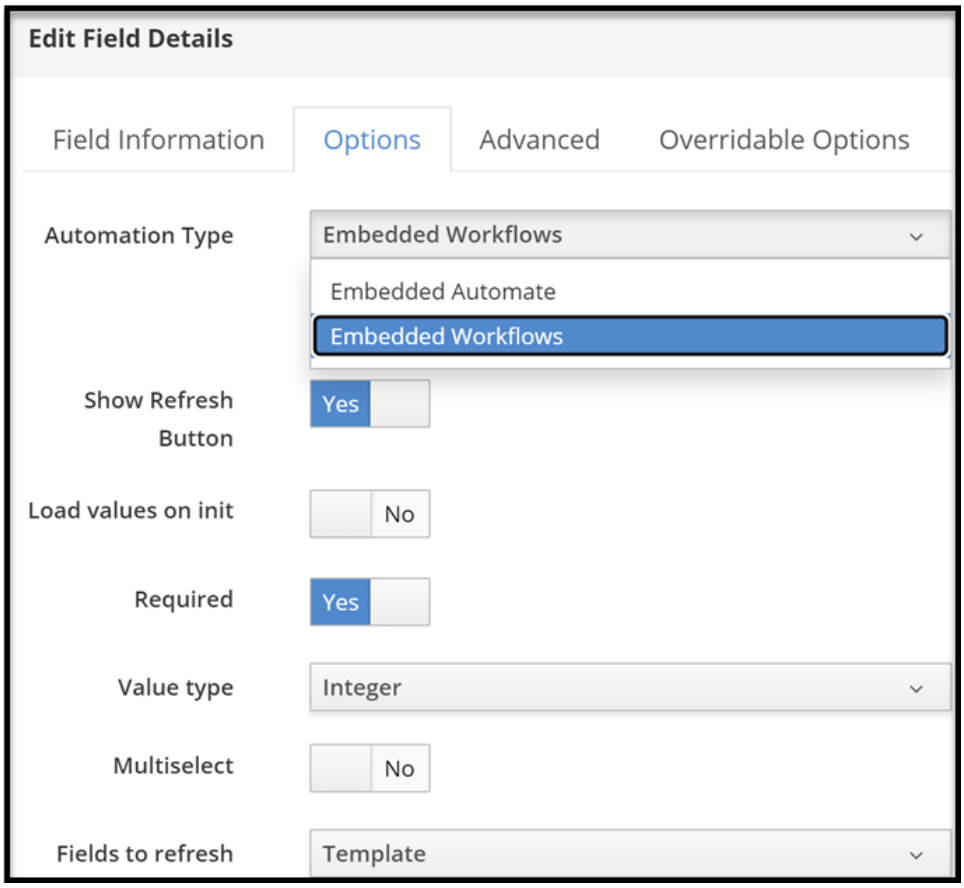
  Credentials are used by Infrastructure Automation for any authentication that is required when running embedded workflows, and when importing workflows from a version control system.

  This page lists, and provides a summary of, the credentials for accessing and using embedded workflows. From this page, you can add, update, and delete credentials.

In addition to the *Embedded Workflows* UI, **Embedded Workflows** can also be used in Service Dialogs for dynamic dialog fields and with Service Catalog Items for provisioning, reconfiguration, and retirement entrypoints.

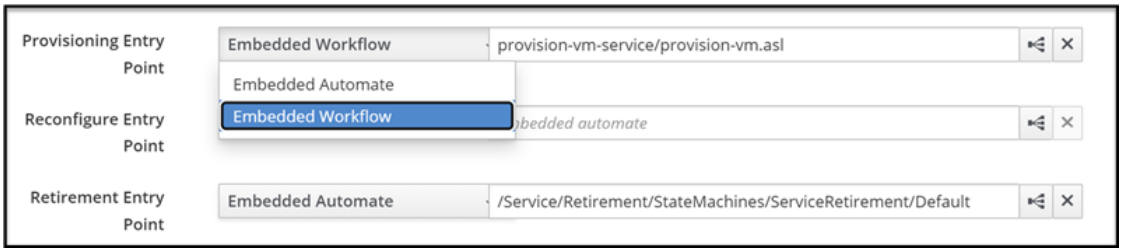
- Service Dialog

  When creating a service dialog, fields that are dynamic include the option for using **Embedded Workflows** or **Embedded Automate**.



- Service Catalog Items

  The provisioning, reconfigure, and retirement entry points include the option for using **Embedded Workflows** or **Embedded Automate**. You can set the point field to the workflow that you want to use for any, or all, of the three service entry points.

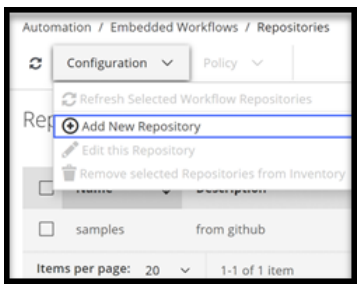


Adding an Embedded Workflows Repository

As a starting point to begin using workflows, you must first add a repository that includes your workflows.

To add a repository, complete the following steps:

1. Click **Automation** > **Embedded Workflows** > **Repositories** > **Configuration** > **Add New Repository**.

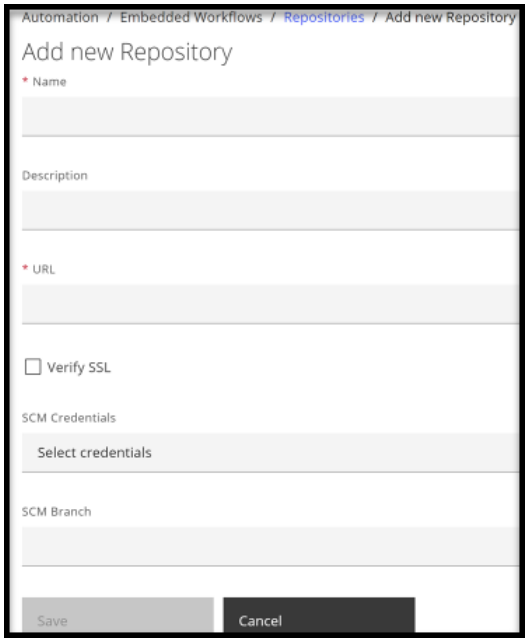


2. Enter the repository name in the **Name** field.
3. Add a description for the repository in the **Description** field.

4. Enter a Git repository URL for the repository in the **URL** field. The Git repository URL does not need to be a GitHub repository.
5. If you are adding a repository that requires authentication then add an SCM (Source Control Management) credential for accessing the repository. Select that credential for this repository from the **SCM Credentials** dropdown menu. These credentials are used in different ways within embedded workflows.

   Source Control credentials have several attributes that can be configured:

   - **Username**: The username to use in conjunction with the source control system.
   - **Password**: The password to use in conjunction with the source control system.
   - **Private key passphrase**: If the SSH private key used is protected by a passphrase, you can configure a key passphrase for the private key.
   - **Private Key**: Copy or drag and drop the actual SSH Private Key to be used to authenticate the user to the source control system by using SSH.
6. Provide a branch name in the **SCM Branch** field for the branch that you want to pull.
7. Click **Save**. Upon saving, the repository is imported and scanned for any included workflows (any Amazon States Language `.asl` files).



8. Confirm that the synchronization with your repository is successful by viewing the list of workflows that are detected for the repository.

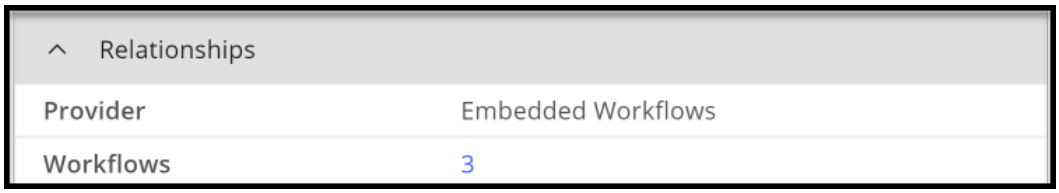
To view the added repositories, return to the **Automation** > **Embedded Workflows** > **Repositories** page. The list of repositories is displayed. To view a summary of details for a repository, click the repository to open the *Summary* page for that repository.

If you need to edit a repository, select the repository from the list of repositories. Then, click **Configuration** > **Edit this Repository**. After you complete your edits, click **Save**.

If you need to delete a repository, select the repository from the list of repositories. Then, click **Configuration** > **Remove this Repository from Inventory**.

Viewing the List of Workflows in a Repository

When your repository is added, the repository shows in the *Repositories* list. You can now click the entry for the repository to open it so that you can view the imported workflows. If any workflows are found, click the corresponding number to view the list of workflows.



Alternatively, click **Automation** > **Embedded Workflows**. Then, click **Workflows** to see the list of workflows across all repositories.

Refreshing Repositories

Infrastructure Automation allows you to refresh a targeted repository or all repositories in your inventory to ensure that your workflows are current.

Refresh a targeted repository:

1. Browse to the menu: **Automation** > **Embedded Workflows** > **Repositories**.
2. Click a repository.
3. Click **Configuration**, then (**Refresh this Repository**).

Alternately, you can refresh some or all repositories from the list view:

1. Browse to the menu: **Automation** > **Embedded Workflows** > **Repositories**.
2. Check those repositories to refresh. Click **Check All** to select all repositories.
3. Click **Configuration**, then (**Refresh Selected Workflow Repositories**).

Adding Credentials

Credentials are used by Infrastructure Automation for any authentication that is required when running embedded workflows, or when importing workflows from a version control system.

Infrastructure Automation can store credentials that are used by embedded workflows. Credentials that are saved in Infrastructure Automation are mapped to values in the `Credentials` field of your Workflow and are made available to the container with a workflow when the workflow runs.

1. Browse to the menu: **Automation** > **Embedded Workflows** > **Credentials**.
2. Click **Configuration**, then (**Add New Credential**).
3. Provide a **Name** for the credential.
4. Select the **Credential Type**. More fields appear depending on the type chosen.
5. Click **Add**.

To view the added credentials, return to the **Automation** > **Embedded Workflows** > **Credentials** page. The list of credentials is displayed. To view a summary of details for a credential, click the credential to open the *Summary* page for that credential.

If you need to edit a credential, select the credential from the list of credentials. Then, click **Configuration** > **Edit this Credential**. After you complete your edits, click **Save**.

If you need to delete a credential, select the credential from the list of credentials. Then, click **Configuration** > **Remove selected Credentials from Inventory**.

Credential Types

Each credential type that is used by Infrastructure Automation for embedded workflows is detailed in the following sections.

SCM

SCM (source control Management) credentials are used with Projects to clone and update local source code Git repositories. Source Control credentials have several attributes that can be configured:

- **Username**: The username to use in conjunction with the source control system.
- **Password**: The password to use in conjunction with the source control system.
- **Private key passphrase**: If the SSH private key used is protected by a passphrase, you can configure a key passphrase for the private key.
- **Private Key**: Copy or drag-and-drop the actual SSH Private Key to be used to authenticate the user to the source control system by using SSH.

Authoring a workflow

You can configure workflows for use in the following areas during service catalog creation:

- **Dynamic dialog fields** - You (or your users) can attach workflows to a field so that the field becomes populated or refreshed with the results from the running of the workflow.
- **Service Entry Points** - You (or your users) can set and use workflows for provisioning, reconfiguring, and retiring services.

You can create and use embedded workflows as needed to not only change parts of the provisioning process, but also to automate other operational tasks.

Workflows must be authored in Amazon State Languages (ASL) format. As part of authoring a workflow, you (or your users) can build container images that are able to perform any tasks that are required in any language that you like. You can use these images during Task states in your workflows.

1. Define the code for the workflow. If your workflow requires the use of any credentials or parameters to be specified, ensure that they are passed in the code.

   Within the workflow code, you need to specify the states that your workflow requires, including any next steps. For `Task` type steps in the workflow, a docker container is called. The container defines what happens for that Task state. For example, a docker container can run to clone a template. If your states require parameters or credentials, you can specify them in your state definitions.

   The workflow code must be in the Amazon States Language (ASL) format and follow its supported specifications. For more information about Amazon States Language and its specification, see Amazon States Language Guide.

   Note: The current implementation of the Amazon States Language does not support certain **Map** states. For more information, see Unsupported Amazon States Language features.

   #### Unsupported Amazon States Language features

   The following features of Amazon States Language are not supported by Floe:

   - Map State Fields:
     - ItemReader
     - ItemSelector/ItemBatcher
     - ResultWriter

2. Build the docker containers that are required for the workflow.

   When you have the code for your task resource written, you need to bundle it into a docker image. You can bundle the code by creating a standard Dockerfile and building the image (https://docs.docker.com/engine/reference/commandline/build/). Then, you can push the image to a registry, which makes the image available to be used by Infrastructure Automation. When you have pushed your images to an image registry, you can add the registry to Infrastructure Automation.

Pull secrets for containers are used differently between appliances and the OpenShift Container Platform (OCP). These differences are outlined in the following sections.

Running an Embedded Workflow on Appliances

- On appliances, **podman** is used to execute the container so use podman login as the **manageiq** user.

  ```
  # su manageiq
  $ podman login docker.io
  Username:
  Password:
  Login Succeeded!
  ```

  If you use the –root flag in the podman pull, images are pulled to a local directory '/var/www/miq/vmdb/data/containers/storage' as in the example

  ```
    podman pull <repository>/<image>:<tag>  --root /var/www/miq/vmdb/data/containers/storage
  ```

  It is worth noting that the default /home/manageiq partition has insufficient space to store large images.

  You can use any repository to store your images, for example you can use docker.io access token so that the token does not expire.

- Provide an image pull secret to a podified Kubernetes container, and then add it to a service account

  In order to pull an image from a private registry you have to provide an **ImagePullSecret** to your containers, see Pull an Image from a Private Registry. Infrastructure Automation uses a service account called ibm-infra-management-application-default to run containers for your workflows. You can add an **ImagePullSecret** to this service account by following Add Image Pull Secrets to a service account.

Running an Embedded Workflow on the OpenShift Container Platform (OCP)

If the user is running an embedded workflow on OCP, and is using a docker repository requiring a custom pull secret for their container images, the following steps need to be carried out.

1. Create a pull secret for the repository in question in the same namespace as Infrastructure Automation, for example:

   ```
   oc create secret docker-registry <pull-secret-name> --docker-server=<docker-server> --docker-username=<docker-username> --docker-password=<docker-password>
   ```

2. Add this pull secret to the ibm-infra-management-application-default, for example:

   ```
   oc secret link ibm-infra-management-application-default <pull-secret-name> --for=pull
   ```

Example: Provisioning Workflow

```
{
  "Comment": "Provision a VMware VM.",
  "StartAt": "CloneTemplate",
  "States": {
    "CloneTemplate": {
      "Type": "Task",
      "Resource": "docker://docker.io/manageiq/workflows-examples-clone-template:latest",
      "Next": "CheckTaskComplete",
      "Credentials": {
        "api_user.$": "$.api_user",
        "api_password.$": "$.api_password",
        "vcenter_user.$": "$.vcenter_user",
        "vcenter_password.$": "$.vcenter_password"
      },
      "Parameters": {
        "PROVIDER_ID.$": "$.dialog_provider",
        "VERIFY_SSL": false,
        "TEMPLATE.$": "$.dialog_source_template",
        "NAME.$": "$.dialog_vm_name"
      }
    },

    "CheckTaskComplete": {
      "Type": "Task",
      "Resource": "docker://docker.io/manageiq/workflows-examples-check-task-complete:latest",
      "Next": "PollTaskComplete",
      "Credentials": {
        "vcenter_user.$": "$.vcenter_user",
        "vcenter_password.$": "$.vcenter_password"
      },
      "Parameters": {
        "VCENTER_HOST.$": "$.vcenter_host",
        "TASK.$": "$.task"
      }
    },

    "PollTaskComplete": {
```

```
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.state",
          "StringEquals": "success",
          "Next": "PowerOnVM"
        },
        {
          "Variable": "$.state",
          "StringEquals": "running",
          "Next": "RetryState"
        },
        {
          "Variable": "$.state",
          "StringEquals": "error",
          "Next": "FailState"
        }
      ],
      "Default": "FailState"
    },

    "RetryState": {
      "Type": "Wait",
      "Seconds": 5,
      "Next": "CheckTaskComplete"
    },

    "PowerOnVM": {
      "Type": "Task",
      "Resource": "docker://docker.io/manageiq/workflows-examples-power-on-vm:latest",
      "Next": "SuccessState",
      "Credentials": {
        "vcenter_user.$": "$.vcenter_user",
        "vcenter_password.$": "$.vcenter_password"
      },
      "Parameters": {
        "VCENTER_HOST.$": "$.vcenter_host",
        "VM.$": "$.vm"
      }
    },

    "FailState": {
      "Type": "Fail",
      "Error": "FailStateError",
      "Cause": "No Matches!"
    },

    "SuccessState": {
      "Type": "Succeed"
    }
  }
}
```

Credentials

Infrastructure Automation provides a mechanism for securely passing credentials to your running workflows. Credentials should never be set statically in your workflow definition.

Long lived credentials like usernames and passwords should be defined as Mapped Credentials as described in **`Adding Credentials`**.

Short lived credentials such as bearer tokens which are obtained while the workflow is running can be set as state output and stored securely in the Credentials field for further states. This can be accomplished by using **`ResultPath`** with a path starting with **`$.Credentials`**. This will set the output of the state in the **`Credentials`** payload.

For an example lets say we have a State which takes a username and password and outputs a bearer token to be used later on:

```
"Login": {
  "Type": "Task",
  "Resource": "docker://login:latest",
  "Credentials": {
    "username.$": "$.username",
    "password.$": "$.password"
  },
  "ResultPath": "$.Credentials",
  "Next": "NextState"
}
```

If the output of the docker image is **`{"bearer_token":"abcd"}`** then we will be able to use this in the next state like so:

```
"NextState": {
  "Type": "Task",
  "Resource": "docker://do-something:latest",
  "Credentials": {
    "token.$": "$.bearer_token"
  }
}
```

All of the normal Input/Output processing still applies so if you need to manipulate the output you can use **`ResultSelector`**. Say for example our login docker image outputs the token as **`{"result":"abcd"}`** but we want to store it as **`"bearer_token"`**. We can use **`ResultSelector`** to change this:

```
"Login": {
  "Type": "Task",
  "Resource": "docker://login:latest",
  "Credentials": {
    "username.$": "$.username",
    "password.$": "$.password"
  },
  "ResultSelector": {
    "bearer_token.$": "$.result"
  },
  "ResultPath": "$.Credentials",
  "Next": "NextState"
}
```

We can also store the result in a parent node for organization:

```
"Login": {
  "Type": "Task",
  "Resource": "docker://login:latest",
  "Credentials": {
    "username.$": "$.username",
    "password.$": "$.password"
  },
  "ResultPath": "$.Credentials.VMware",
  "Next": "NextState"
}
```

And then access it like:

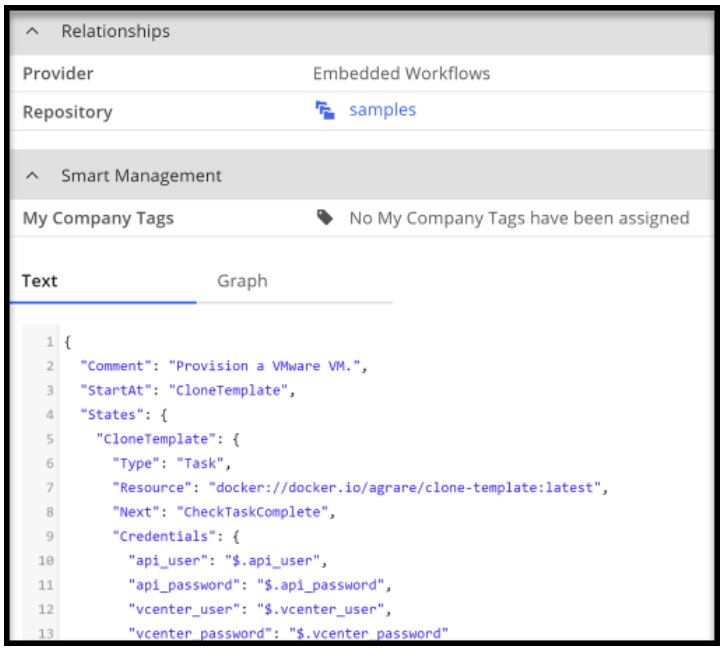
```
"NextState": {
  "Type": "Task",
  "Resource": "docker://do-something:latest",
  "Credentials": {
    "token.$": "$.VMware.bearer_token"
  }
}
```

If you have a mapped credential and your state overwrites its, a new non-mapped entry will be created and the original mapped credential will be left intact.

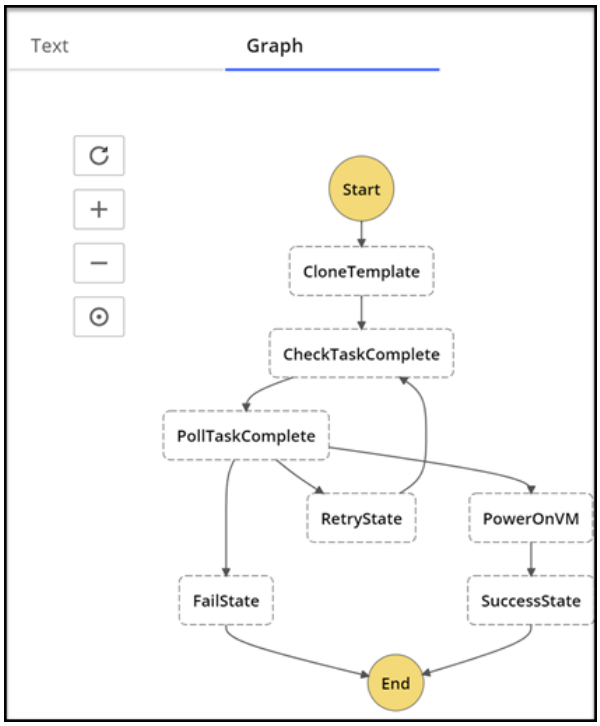
Viewing workflow details

Once a workflow is imported, you can view the details, such as to verify the payload. To view the details for a workflow:

1. Go to the menu: **Automation** > **Embedded Workflows**. Then, click **Workflows** to see the workflows list.
2. Click a workflow. The default *Text* view of the workflow opens where you see the code content.



   Alternatively, click the **Graph** tab to see a graph of the same data.



3. If your workflow has credential fields, create a workflow credential for each of the fields. For more information about creating a credential, see Adding Credentials.

4. Map any workflow credentials to the workflow.

    1. Go to the menu: **Automation** > **Embedded Workflows**. Then, click **Workflows** to see the workflows list.

    2. Select the workflow that includes the credentials that you need to map.

    3. Click **Configuration** > **Map Credentials to this Workflow**.

        The table that displays on the *Map Credentials* page displays all the current mappings for the selected workflow. Verify that the list of credential identifiers includes all of the identifiers that are in the workflow payload.

    4. Use the drop-down fields to map one or more credentials for the workflow. Verify that the table is updated with the correct information.

    5. Click **Save**.

Creating a Service Dialog using an Embedded Workflow

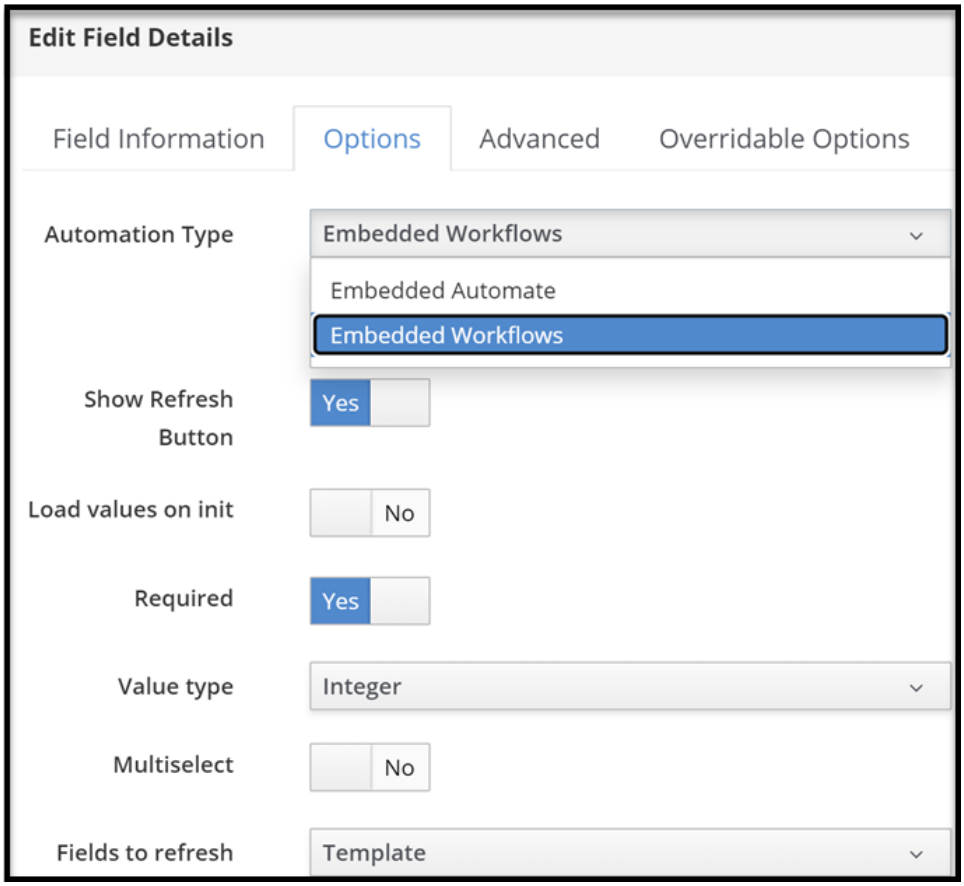
Fields that are *Dynamic* offer the choice of *Embedded Workflows* in addition to the original *Embedded Automate*.

Dialogs are the part of Infrastructure Automation that you interface with when you order services.

1. To open a service dialog, click **Automation** > **Embedded Automate** > **Customization**.
2. From the *Service Dialog* section, review the list of created dialogs. Click one of the dialogs to view the details.
3. Click **Edit the dialog**. You can edit the *Name*, *Provider*, and *Template*.

    When you select to edit any of the dropdown settings, you can see a *dynamic* option. Toggle to **On** to make sure it is enabled.

4. Click the **Options** tab. You can now select between *Embedded Automate* and *Embedded Workflows* in the Automation field.

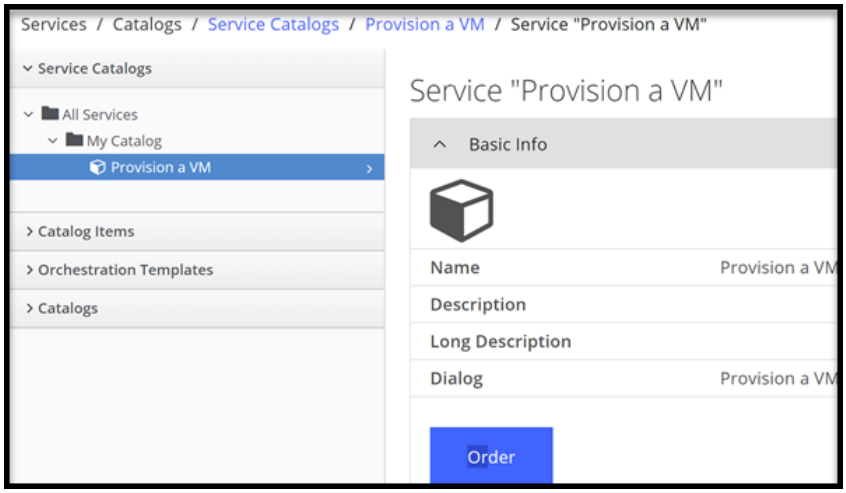


If you choose **Embedded Workflows**, you can select the workflow to run. Complete the remaining fields and click **Save** when you complete the form. You can chain dialog fields together by using **Fields to refresh**, which lets one dialog field trigger the run of another when there are interdependencies. When you *order* a service catalog, any dynamic dialogs that are backed by embedded workflows run in the background to populate the dialogs.

Creating Service Catalog Items using an Embedded Workflow

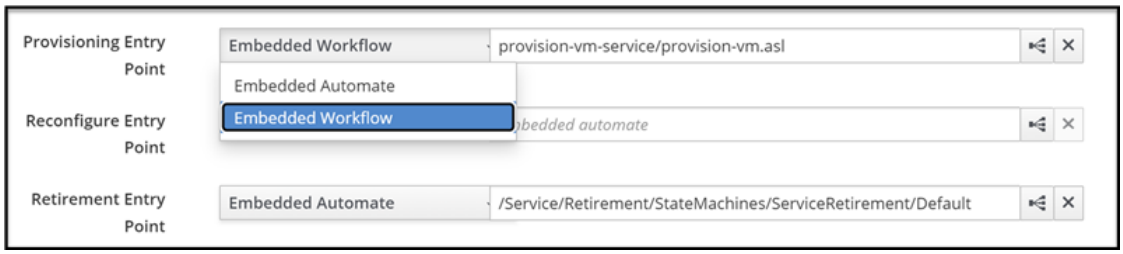
You can create a generic service catalog item that uses an embedded workflow. To use previously set up workflows and to view a service catalog item, you need to *order* the item. Complete the following steps to order a service catalog item that uses embedded workflows:

1. Click **Services** > **Catalogs**.
2. The *Service Catalogs* section opens and you see the existing Service Catalogs. If you want to order one, click it and then **Order**.



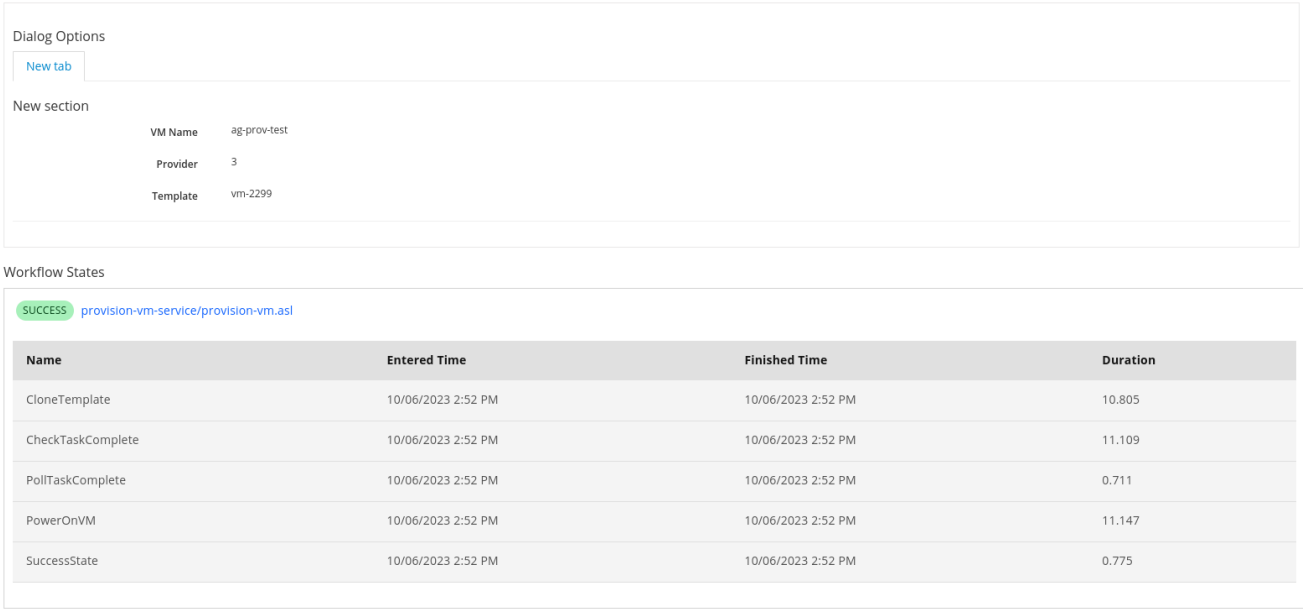
Complete the fields for your specified dialog. Then, click **Submit**.

3. If you want to know more about the item first or to edit it, click the **Catalogs** section, and then click that item that is created. Click **Configuration** > **Edit this item**.



4. When you complete all your edits, click **Save**.
5. Confirm that you can order the service catalog item and that it runs the workflows to drive the dynamic dialog dropdown.

   The list of services and requests is shown when the catalog item is submitted. Clicking the request shows the execution status, including any embedded workflows.



Ansible

Ansible integrates with Infrastructure Automation to provide automation solutions, using playbooks, for Service, Policy and Alert actions. Ansible playbooks consist of series of *plays* or tasks that define automation across a set of hosts, which is known as the inventory.

Ranging from simple to complex tasks, Ansible playbooks can support cloud management:

- **Services** - allow a playbook to back a Infrastructure Automation service catalog item.
- **Control Actions** - Infrastructure Automation policies can execute playbooks as actions based on events from providers.
- **Control Alerts** - set a playbook to launch prompted by a Infrastructure Automation alert.

Ansible is built into Infrastructure Automation so there is nothing to install. The basic workflow when using Ansible in Infrastructure Automation is as follows:

1. Enable the **Embedded Ansible** server role.
2. Add a source control repository that contains your playbooks.
3. Establish credentials with your inventory.
4. Back your services, alerts, and policies using available playbooks.

Enabling the Embedded Ansible Server Role

In Infrastructure Automation, the **Embedded Ansible** role is disabled by default. Enable this server role to utilize Ansible Automation Inside.

Note: Configure your Infrastructure Automation appliance network identity (hostname/IP address) before enabling the Embedded Ansible server role. Restart the **`evmserverd`** service on the appliance with the enabled Embedded Ansible server role after making any changes to the hostname or IP address.

1. Browse to the settings menu, then **Configuration > Settings**.
2. Select the desired server under **Zones**.
3. Set the **Server Role** for **Embedded Ansible** to **On**.

Verifying the Embedded Ansible Worker State

Verify that the Embedded Ansible worker has started to utilize its features.

1. Browse to the settings menu, then **Configuration > Diagnostics** and click on the desired server.
2. Click on the **Roles by Servers** tab.

A table of all workers and current status will appear from which you can confirm the state of your embedded Ansible worker.

Adding a Playbook Repository

Add a repository so that Infrastructure Automation can discover and make available your playbooks.

1. Browse to menu: **Automation > Ansible > Repositories**.
2. Click **Configuration**, then (**Add New Repository**).
3. Provide a Repository Name in the **Name** field.
4. Add a description for the repository in the **Description** field.
5. Add a **URL** or IP Address for the repository.
6. Select the appropriate **SCM Credentials** from the drop-down menu.
7. Provide a branch name in the **SCM Branch** field.
8. Click **Save**.

Once you have synced a repository, its playbooks will become available to Infrastructure Automation.

Refreshing Repositories

Infrastructure Automation allows you to refresh a targeted playbook repository or all repositories in your inventory to ensure your playbooks are current.

Refresh a targeted repository:

1. Browse to menu: **Automation > Ansible > Repositories**.
2. Click on a repository.
3. Click **Configuration**, then (**Refresh this Repository**).

Alternately, you can refresh some or all repositories from the list view:

1. Browse to menu: **Automation > Ansible > Repositories**.
2. Check those repositories to refresh. Click **Check All** to select all repositories.
3. Click **Configuration**, then (**Refresh Selected Ansible Repositories**).

Credentials

Credentials are utilized by Infrastructure Automation for authentication when running Ansible playbooks against machines, synchronizing with inventory sources, and importing project content from a version control system.

Adding Credentials

Infrastructure Automation can store credentials used by playbooks. Credentials saved in Infrastructure Automation are matched and executed with a playbook when run.

1. Browse to menu: **Automation > Ansible > Credentials**.
2. Click **Configuration**, then (**Add New Credential**).
3. Provide a **Name** for the credential.
4. Select the **Credential Type**. Additional fields will appear depending on the type chosen.
   - **Vault Password**: Ansible Vault credentials have only the **Vault Password** attribute that may be configured. For more information on Ansible Vault, see Using Vault in playbooks.
5. Click **Add**.

Credential Types

Each credential type used by Infrastructure Automation is detailed in the following sections.

Machine

Machine credentials enable Infrastructure Automation to invoke Ansible on hosts under your management. Just like using Ansible on the command line, you can specify the SSH username, optionally provide a password, an SSH key, or a key password. They define SSH and user-level privilege escalation access for playbooks, and are used when running playbooks on a remote host.

- **Username**: The username to be used for SSH authentication.
- **Password**: The actual password to be used for SSH authentication.
- **SSH Private Key**: Copy or drag-and-drop the SSH private key for the machine credential.
- **Private Key Passphrase**: If the SSH Private Key used is protected by a password, you can configure a Key Password for the private key.
- **Privilege Escalation**: Specifies the type of escalation privilege to assign to specific users. Options include **sudo**, **su**, **pbrun**, **pfexec**.
- **Privilege Escalation Username**: Enter the username to use with escalation privileges on the remote system.
- **Privilege Escalation Password**: Enter the actual password to be used to authenticate the user via the selected privilege escalation type on the remote system.

Network

Network credentials are used by Ansible networking modules to connect to and manage networking devices.

Network credentials have several attributes that may be configured:

- **Username**: The username to use in conjunction with the network device.
- **Password**: The password to use in conjunction with the network device.
- **Authorize**: Select this from the Options field to add an Authorize password which signs the RSA key with a password.
- **Authorize password**: If **Authorize** is checked, enter a password in the **Authorize Password** field.
- **SSH Key**: Copy or drag-and-drop the actual SSH Private Key to be used to authenticate the user to the network via SSH.
- **Private key passphrase**: The actual passphrase for the private key to be used to authenticate the user to the network via SSH.

SCM

SCM (source control) credentials are used with Projects to clone and update local source code repositories from a remote revision control system such as Git, Subversion, or Mercurial.

Source Control credentials have several attributes that may be configured:

- **Username**: The username to use in conjunction with the source control system.
- **Password**: The password to use in conjunction with the source control system.
- **Private key passphrase**: If the SSH private key used is protected by a passphrase, you may configure a key passphrase for the private key.
- **Private Key**: Copy or drag-and-drop the actual SSH Private Key to be used to authenticate the user to the source control system via SSH.

Amazon

Selecting this credential type enables synchronization of cloud inventory with Amazon Web Services.

- **Access Key**: User credentials that allow for programmatic calls to Amazon Web Services.
- **Secret Key**: The secret key that corresponds to the user access key.
- **STS Token**: Token generated by Amazon Web Services Security Token Service.

Azure

Selecting this credential type enables synchronization of cloud inventory with Microsoft Azure.

Microsoft Azure credentials have several attributes to configure:

- **Username**: The username to use to connect to the Microsoft Azure account.
- **Password**: The password to use to connect to the Microsoft Azure account.
- **Subscription ID**: The Subscription UUID for the Microsoft Azure account.
- **Tenant ID**: The Tenant ID for the Microsoft Azure account.
- **Client Secret**: The Client Secret for the Microsoft Azure account.
- **Client ID**: The Client ID for the Microsoft Azure account.

OpenStack

Selecting this credential type enables synchronization of cloud inventory with Red Hat OpenStack Platform.

OpenStack credentials have several attributes that may be configured:

- **Username**: The username to use to connect to OpenStack.
- **Password (API Key)**: The password or API key to use to connect to OpenStack.
- **Host (Authentication URL)**: The host to be used for authentication.
- **Project (Tenant Name)**: The Tenant name or Tenant ID used for OpenStack. This value is usually the same as the username.
- **Domain name**: The FQDN to be used to connect to OpenStack.

Red Hat Virtualization

Selecting this credential type enables synchronization of cloud inventory with Red Hat Virtualization.

Red Hat Virtualization credentials have several attributes that may be configured:

- **Username**: The username to use to connect to Red Hat Virtualization.
- **Password**: The password to use to connect to Red Hat Virtualization.
- **Host (Authentication URL)**: The host to be used for authentication.

  Important: Enter in **Host** the Red Hat Virtualization provider URL, followed by the path `/ovirt_engine/api`. Example: `https://your.rhv.com/ovirt_engine/api`

  For more information on the Ansible Roles that are available for Red Hat Virtualization, see Ansible Roles.

VMware

Selecting this credential type enables synchronization of inventory with VMware vCenter.

Important: If both Infrastructure Automation and a VMware provider are located in the same IPv6-only network, use a DNS-resolvable hostname for the VMware provider in the **vCenter Host** field when adding credentials.

VMware credentials have several attributes that may be configured:

- **Username**: The username to use to connect to vCenter.
- **Password**: The password to use to connect to vCenter.
- **vCenter Host**: The vCenter hostname or IP address to connect to.

Note: If the VMware guest tools are not running on the instance, VMware inventory sync may not return an IP address for that instance.

Tagging Ansible Playbooks, Repositories, and Credentials

Apply tags to Ansible playbooks, repositories, and credentials to categorize them. Tagging enables administrators to limit users to view those Ansible elements that have been enabled for that set of user permissions.

Adding Tags to Ansible Playbooks

1. Navigate to **Automate > Ansible > Playbooks**.
2. Select the checkboxes for the Ansible playbooks to tag.
3. Click **Policy**, and then (**Edit Tags**).
4. Select a customer tag to assign from the first list.
5. Select a value to assign from the value list.
6. Click **Save**.

Adding Tags to Ansible Repositories

1. Navigate to **Automate > Ansible > Repositories**.
2. Select the checkboxes for the Ansible repositories to tag.
3. Click **Policy**, and then (**Edit Tags**).
4. Select a customer tag to assign from the first list.
5. Select a value to assign from the value list.
6. Click **Save**.

Adding Tags to Ansible Credentials

1. Navigate to **Automate > Ansible > credentials**.
2. Select the checkboxes for the Ansible credentials to tag.
3. Click **Policy**, and then (**Edit Tags**).
4. Select a customer tag to assign from the first list.
5. Select a value to assign from the value list.
6. Click **Save**.

Optimizing Ansible Playbooks for Infrastructure Automation

Ansible is a simple model-driven configuration management, multi-node deployment, and remote-task execution system. When designing playbooks for use with Infrastructure Automation it is helpful to utilize solutions within the playbook itself to ensure optimal implementation of playbook-backed services or automated processes.

This section is intended to complement the existing documentation on Ansible playbooks and guide administrators through optimizing playbooks for use with Infrastructure Automation.

Installing Roles on an Embedded Ansible Appliance

Roles are ways of automatically loading certain variable files, tasks, and handlers based on a known file structure. Grouping content by roles also allows for easy sharing of roles with other users. Install roles on a Infrastructure Automation appliance with the Embedded Ansible server role activated to optimize playbooks.

When using this role in a playbook on a Infrastructure Automation appliance, add an empty **`roles`** directory at the root of the playbook. In the **`roles`** directory, include a **`requirements.yml`** file with the following contents:

```
---
- src: <ansible-galaxy-role>
```

Infrastructure Automation automatically installs the role once it sees the **`requirements.yml`** file in the playbook.

Ansible Service Linking

Infrastructure Automation provides a module allowing inventoried resources such as virtual machines created using Ansible playbooks to link back to the services used to generate them. During service ordering of a playbook the **`add_provider_vms`** module will allow the playbook to connect back to the worker appliance and identify the provider resources it was responsible for generating. Once linked, the newly generated resources are available to Infrastructure Automation's life cycle management features.

Linking VMs back to the service that created it requires implementing the following tasks in the playbook used for provisioning:

1. Create a resource and register it.
2. Link the service using the **`add_provider_vms`** method to the newly created resource.

Example: Linking a virtual machine to a service

In the following playbook task examples, a virtual machine is deployed to Amazon EC2 and linked back to the service. Examples are provided for linking the resource to its service by both an *href slug* and as an object.

Note:

- This example utilizes the `syncrou.manageiq-vmdb` role. This role allows Infrastructure Automation users to modify and/or change VMDB objects using an Ansible playbook. For information on implementing and utilizing roles when writing Ansible playbooks for Infrastructure Automation, see Installing Ansible Roles.
- For more information on Ansible Galaxy and roles, see the Ansible Galaxy documentation.
- Note the provider ID in order to successfully link to the service.

1. Create and register the resource.

```
- name: Create Ec2 Instance
  ec2:
    key_name: ""
    instance_tags: {Name: ""}
    group_id: ""
    instance_type: ""
    region: ""
    image: ""
    wait: yes
    count: 1
    vpc_subnet_id: ""
    assign_public_ip: yes
  register: ec2
```

2. Call the **`add_provider_vms`** method as an action to link to the service via an *href slug* or an object.

```
- name: Service Linking via an href slug
  manageiq_vmdb:
    href: "href_slug::services/80"
    action: add_provider_vms
    data:
      uid_ems:
        - ""
      provider:
        id: 24

- name: Service Linking via an object
  manageiq_vmdb:
    vmdb: ""
    action: add_provider_vms
    data:
      uid_ems:
        - ""
      provider:
        id: 24
```

Modifying the Automate Workspace Using the `manageiq-automate` Role.

The **manageiq-automate** role allows users of Infrastructure Automation Automate to modify and add to the automate workspace via an Ansible playbook.

Note: When using this role in a playbook on a Infrastructure Automation appliance with Embedded Ansible activated, add an empty **roles** directory at the root of the playbook. In the **roles** directory, include a **requirements.yml** file with the following contents:

```
---
- src: syncrou.manageiq-automate
```

Infrastructure Automation will automatically install the role once it sees the **requirements.yml** file in the playbook.

Role Variables

The **manageiq_automate** role employs the following variables when implemented in a playbook run on a Infrastructure Automation appliance. Variables are defined in **defaults/main.yml** and **vars/main.yml**.

auto_commit: By default is set to **True**. If set to False it does not auto commit back to Infrastructure Automation each call to a **set_** method in the **manageiq_automate** module.

manageiq_validate_certs: By default is set to **True**. If passed in with **extra_vars** or assigned in the playbook variables then the lookup allows self-signed certificates to be used when using SSL REST API connection URLs.

Example Playbook

The following example utilizes the **manageiq-automate** role. Using variable substitution, playbook tasks retrieve method parameters which are then used to modify object attributes. A final task uses the **set_retry** module to update the retry interval.

```
- name: Siphon Method Parameters into an object
  hosts: localhost
  connection: local
  vars:
  - auto_commit: True
  - object: root
  - interval: 600

  gather_facts: False
  roles:
  - syncrou.manageiq-automate

  tasks:
    - name: "Get the list of Method Parameters"
      manageiq_automate:
        workspace: ""
        get_method_parameters: yes
      register: method_params

    - name: "Set attributes"
      manageiq_automate:
        workspace: ""
        set_attributes:
          object: ""
          attributes: ""

    - name: Set Retry
      manageiq_automate:
        workspace: ""
        set_retry:
          interval: ""
```

Callbacks in Multiple Appliance Environments

In a Infrastructure Automation multiple appliance environment, enable the Embedded Ansible server role on a dedicated Infrastructure Automation appliance. Add **store_session:sql** to Ansible playbooks to ensure successful callbacks to Infrastructure Automation appliances in a multiple appliance environment.

Ansible Tower

Ansible Tower is a management tool integrated with Infrastructure Automation, designed to help automate infrastructure operations. Infrastructure Automation allows you to execute Ansible Tower jobs or workflows using service catalogs and Automate. No custom configuration or Ruby scripting is needed in Infrastructure Automation, as configuration is done in Ansible Tower using playbooks.

You can use the large library of existing Ansible playbooks as Infrastructure Automation state machines to automate tasks such as deployments, backups, package updates, and maintenance in your Infrastructure Automation environment. This can be particularly useful for quickly applying changes across large environments with many virtual machines or instances.

Using Ansible Tower, you can schedule Ansible playbook runs and monitor current and historical results, allowing for troubleshooting or identification of issues before they occur.

Infrastructure Automation supports Ansible Tower API v2 provider integration.

Working with an Ansible Tower Provider

The basic workflow when using Infrastructure Automation with an Ansible Tower provider is as follows:

1. Create an Ansible playbook which performs a specific task.
2. A new Ansible Tower job template is created from the playbook (or workflow template created from disparate jobs), which is then retrieved by Infrastructure Automation.
3. From the Ansible Tower job or workflow template, create a new catalog item in Infrastructure Automation, optionally with a service dialog that allows the user to enter parameters if needed.
4. The user orders the service from the Infrastructure Automation user interface, and fills out any additional arguments (for example, limiting the task to run on a specific set of virtual machines).
5. The job or workflow executes.

Notes:

- For more information about Ansible playbooks, see the Ansible playbook documentation.
- For more information about worklows, see Workflows in the Ansible Tower *User Guide*.

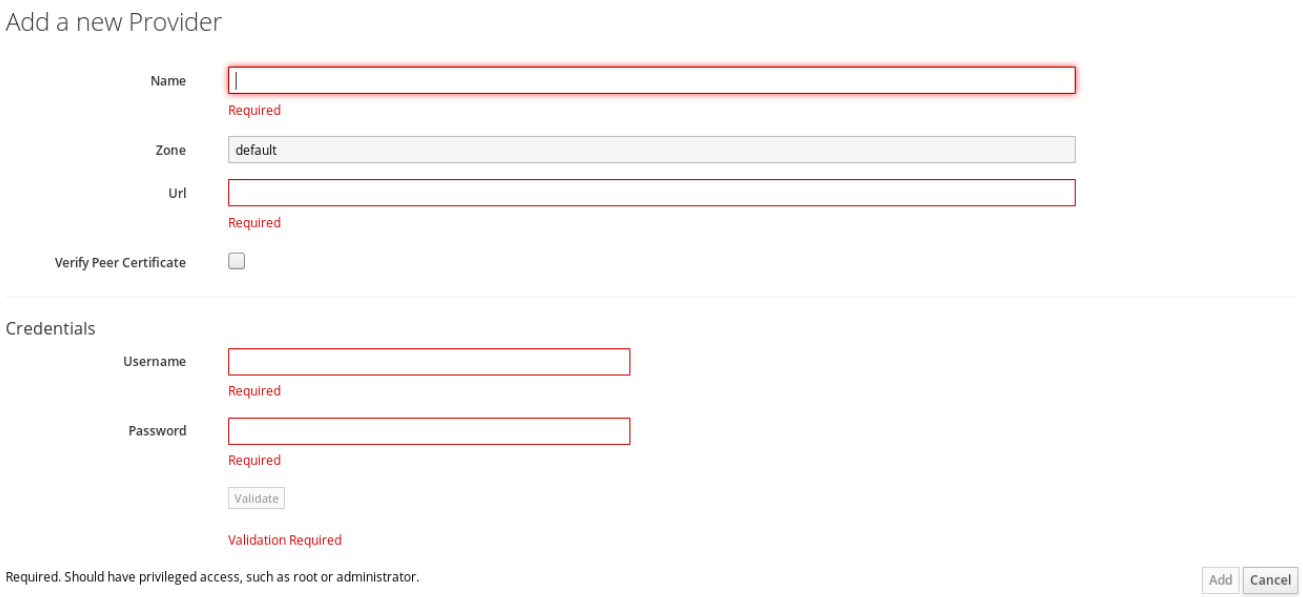
Adding an Ansible Tower Provider

To access your Ansible Tower inventory from Infrastructure Automation, you must add Ansible Tower as a provider.

Notes:

- Ensure **ENABLE HTTP BASIC AUTH** is set to **On** in the Ansible Tower configuration settings before adding the provider. See Tower Configuration in the Ansible Tower *Administration Guide*.
- A trailing slash is **not** required at the end of the Ansible Tower provider URL. Adding the trailing slash to the provider URL may result in a validation error.

1. Browse to menu: **Automation > Ansible Tower > Explorer** and click on the **Providers** accordion tab.
2. Under **Configuration**, click **Add a new Provider**.
3. In the **Add a new Provider** area:



1. Enter a **Name** for the new provider.

2. Add a **Zone** for the provider.
3. Enter the **URL** location or IP address to the Ansible Tower server. Add a trailing slash to the end of the Ansible Tower provider URL.

4. Select the **Verify Peer Certificate** checkbox if desired.
5. In the **Credentials** area, provide the **Username** and **Password**, and **Confirm Password**.
6. Click **Validate** to verify credentials.
7. Click **Add**.

After adding the Ansible Tower provider, refresh its relationships and power states in order to view the current inventory.

Refreshing an Ansible Tower Provider

Refresh relationships of all items related to an existing Ansible Tower configuration management provider including inventory, hosts, virtual machines, and clusters.

You can refresh inventory from Infrastructure Automation, or by enabling the **Update on Launch** option for inventory groups in Ansible Tower. The **Update on Launch** option allows Ansible Tower to automatically update inventory using a dynamic inventory script before launching an Ansible Tower job from a playbook. For more information, see the Ansible Tower documentation.

Important: It can take a long time to retrieve information from providers containing many virtual machines or instances. The Ansible Tower dynamic inventory script can be modified to limit updates to specific items and reduce refresh time.

To refresh an Ansible Tower provider's inventory in Infrastructure Automation:

1. Browse to menu: **Automation > Ansible Tower > Explorer** and click the **Providers** accordion tab.
2. Select the checkboxes for the Ansible Tower providers to refresh under **All Ansible Tower Providers**.
3. Click **Configuration**, and then (**Refresh Relationships and Power States**).
4. Click **OK**.

Infrastructure Automation then queries the Ansible Tower API and obtains an inventory of all available hosts, job, and workflow templates.

Viewing Ansible Tower Providers and Inventory

Infrastructure Automation automatically updates its inventory from Ansible Tower. This includes system groups (known as Inventories in Ansible Tower), basic information about individual systems, and available Ansible Tower job or workflow templates to be executed from the service catalog or Automate.

Note: To view and access Ansible Tower inventories and job or workflow templates in Infrastructure Automation, you must first create them in Ansible Tower.

To view a list of Ansible Tower providers and inventory:

1. Browse to menu: **Automation > Ansible Tower > Explorer**.
2. select the **Providers** accordion menu to display a list of **All Ansible Tower Providers**.
3. Select your Ansible Tower provider to expand and list the inventory groups on that Ansible Tower system. The inventory groups can be expanded to view the systems contained within each group, as well as configuration details for these systems.

Similarly, all discovered job and workflow templates are accessed under the provider by expanding the menu: **Automation > Ansible Tower > Explorer** and click the **Templates** accordion menu.

Viewing Ansible Tower Configured Systems

To view the systems in your Ansible Tower inventory:

1. Browse to menu: **Automation > Ansible Tower > Explorer** and click **Configured Systems**.
2. Under **All Ansible Tower Configured Systems**, select **Ansible Tower Configured Systems** to display a list.

Executing an Ansible Tower Job or Workflow Template from a Service Catalog

You can execute an Ansible Tower playbook from Infrastructure Automation by creating a service catalog item from an Ansible Tower job or workflow template.

Important: You must first create the job or workflow template in Ansible Tower. The job or workflow templates are automatically discovered by Infrastructure Automation when refreshing your Ansible Tower provider's inventory.

First, create a catalog:

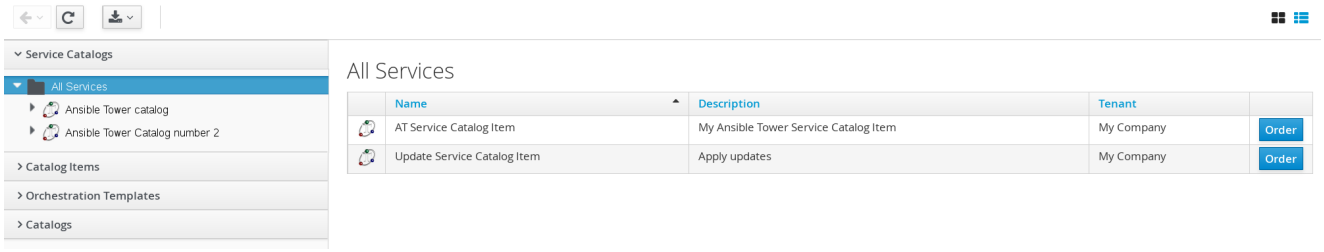
1. Browse to menu: **Services > Catalogs** and click **Catalog Items**.
2. Click **Configuration**, then (**Add a New Catalog**)
3. Enter a **Name** and **Description** for the catalog.
4. Click **Add**.

Then, create an Ansible Tower service catalog item:

1. Browse to menu: **Automation > Ansible Tower > Explorer**, then click the **Templates** according menu.
2. Click **Ansible Tower Templates** and select an Ansible Tower job or workflow template.
3. Click **Configuration**, then (**Create Service Dialog from this Template**).
4. Enter a **Service Dialog Name** (for example, *ansible_tower_job*)and click **Save**.
5. Browse to menu: **Services > Catalogs** and click **Catalog Items**.
6. Click **Configuration**, then (**Add a New Catalog Item**) to create a new catalog item with the following details, at minimum:
    - For **Catalog Item type**, select **Ansible Tower**.
    - Enter a **Name** for the service catalog item.
    - Select **Display in Catalog**.
    - In **Catalog**, select the catalog you created previously.
    - In **Dialog**, select the service dialog you created previously (in this example, *ansible_tower_job*). To ask the user to enter extra information when running the task, **Service Dialog** must be selected. A dialog is required if **Display in Catalog** is chosen.
    - In **Provider**, select your Ansible Tower provider. This brings up the **Ansible Tower Template** option and configures the **Provisioning Entry Point State Machine** automatically.
    - Add configuration information for **Reconfigure Entry Point** and **Retirement Entry Point** as applicable.
    - Select your desired **Ansible Tower Template** from the list. Generally, this is the Ansible Tower job or workflow template previously used to create the service dialog.
7. Click **Add**. The catalog item you created will appear in the **All Service Catalog Items** list.

To execute the Ansible Tower job:

1. Browse to menu: **Service > Catalogs** and click on **Service Catalogs** then click **Ansible Tower catalog**.



2. Click **Order** for the catalog item.
3. Enter any variables requested and click **Submit**.

Infrastructure Automation takes you to the **Requests** queue page and show the status of the job.

The service item's details can be viewed in menu: **Services > My Services** in Infrastructure Automation.

Note: Instead of running a single job at a time, multiple service catalog items can also be grouped together as a catalog bundle to create one deployment with multiple job templates. For more information, see Catalogs and Services.

Executing an Ansible Tower Job Using a Custom Automate Button

Infrastructure Automation can execute Ansible Tower jobs on virtual machines or instances using custom buttons in Automate.

Ansible Tower jobs can either be noncustomizable, which do not require any extra configuration from the user, or alternatively, they can allow the user to specify a parameter (for example, a package name to install). In Ansible Tower jobs containing a dialog, Infrastructure Automation accepts additional information from the user and adds it to the appropriate API call in Automate, and then sends it into Ansible Tower.

Prerequisites.

Before creating an Automate button to execute an Ansible Tower job, the following must be configured:

- An Ansible playbook in Ansible Tower. See the Ansible Tower documentation for instructions.
- Ansible Tower must be able to reach virtual machines or instances deployed by Infrastructure Automation at the IP level.
- The virtual machine template must have the Ansible Tower environment's public SSH key injected. For cloud instances, `cloud-init` can be used and the public SSH key can be passed without rebuilding the image.
- Any dynamic inventory scripts used must be configured to return the virtual machine names exactly as they are stored in Infrastructure Automation, without the UUID appended.

Executing an Ansible Tower Job using a Custom Automate Button.

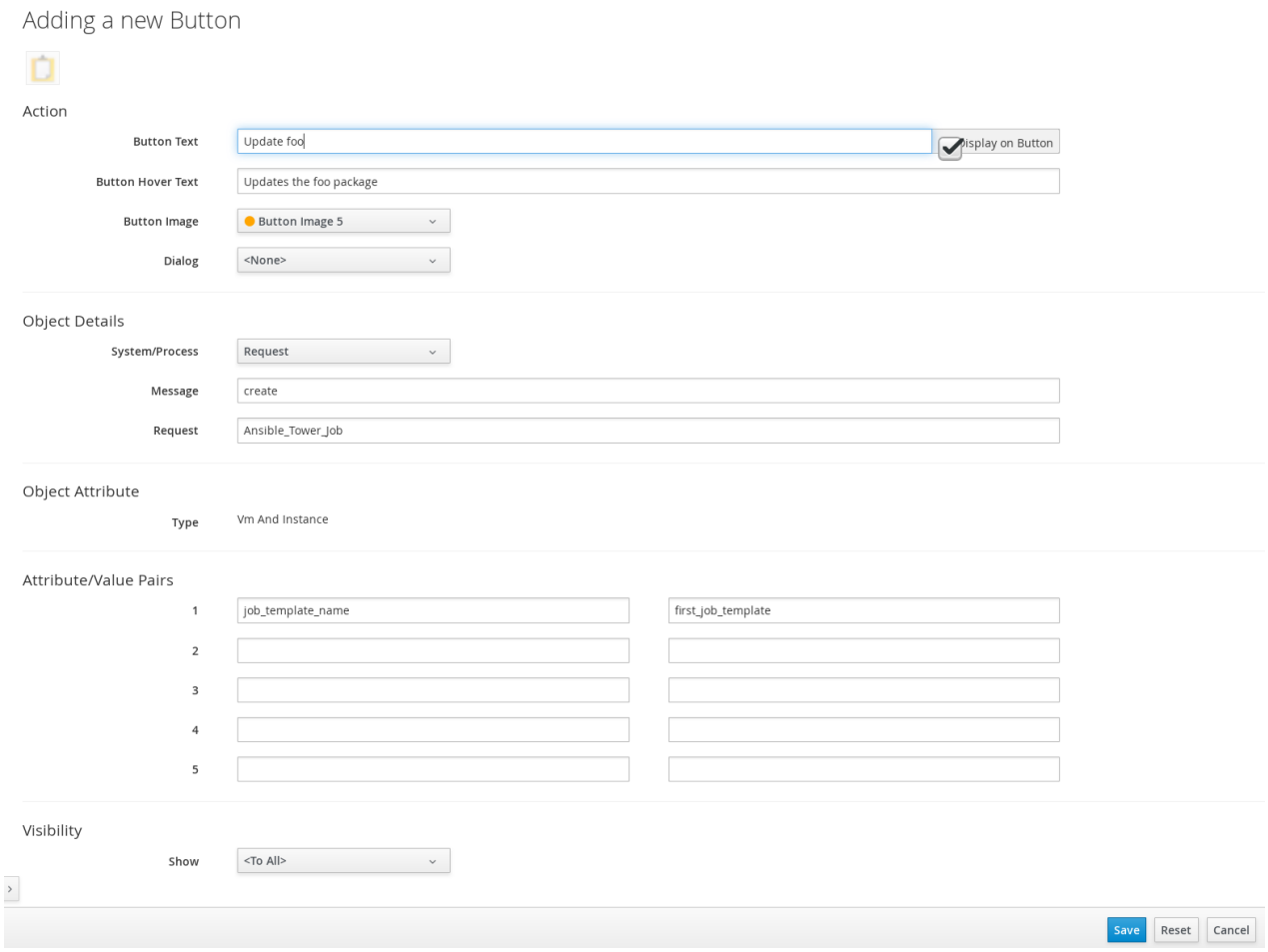
To configure a custom button to execute an Ansible Tower job on a virtual machine or instance, first create the button:

1. Browse to menu: **Automation > Automate > Customization**.
2. Click the **Buttons** accordion menu.
3. Click menu: **VM and Instance > Unassigned Buttons**. This configures the button to run on virtual machines or instances.
4. Click **Configuration**, then click (**Add a new Button**).
    1. In the **Adding a new Button** screen, configure the **Action** parameters as desired. **Dialog** can be left blank if the playbook does not require extra variables. To ask the user to enter extra information when running the task, **Service Dialog** must be selected.
    2. Configure **Object Details** fields with the following request details:

- For **System/Process**, select **Request**.
- For **Message**, enter **create**.
- For **Request**, enter **Ansible_Tower_Job**.

1. Configure **Attribute/Value Pairs** with the following parameters:

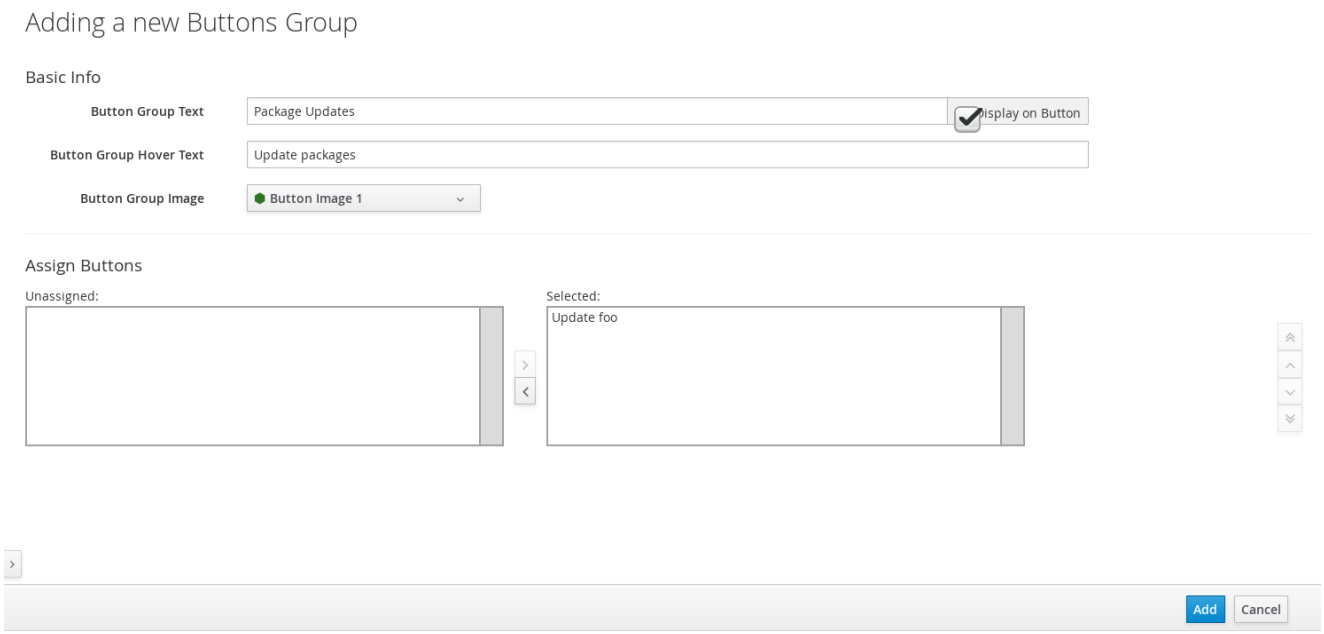
- **job_template_name** is the Ansible Tower job template name to associate with the button. The **job_template_name** field is mandatory; other parameters are provided by the Tower job dialog.
- Configure **Visibility** to all users, or limit visibility by role as desired.



1. Click **Add**.

If you do not have an existing button group to assign the new button to, create a new button group:

1. From menu: **Automation > Automate > Customization**, browse to menu: **Buttons** and click **VM and Instance > Add a new Button Group**, and configure the following:
    1. Configure **Basic Info** as desired. For example, name the button group **`VM Actions`**.
    2. In **Assign Buttons**, select the button you just created from the **Unassigned** list and click to assign it to **Selected**.



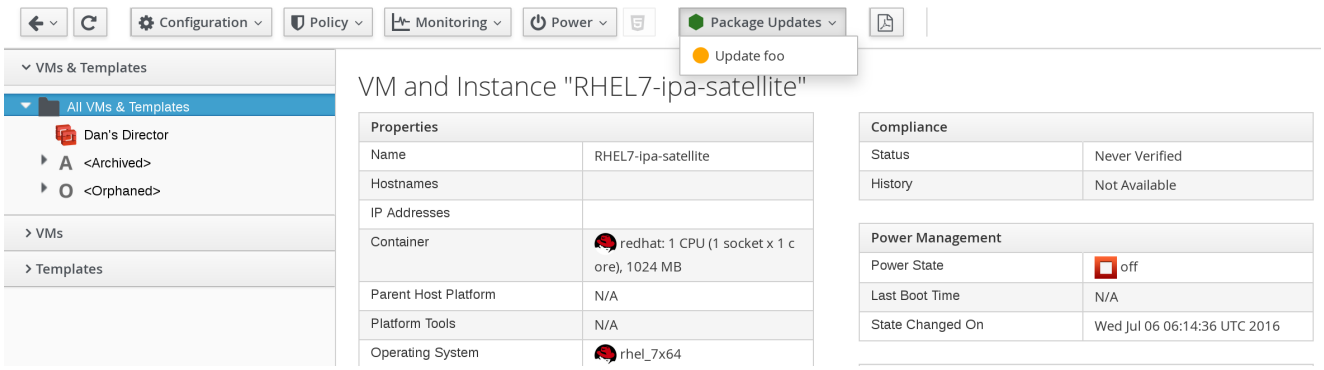
1. Click **Add**.

To assign the button to an existing button group:

1. Browse to menu: **Buttons > VM and Instance > VM Actions > Edit this Button Group**.
2. In **Assign Buttons**, select the button you just created from the **Unassigned** list and click to assign it to **Selected**.
3. Click **Add**.

To use the button to run an Ansible Tower job on a virtual machine:

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**.
2. Select the virtual machine to run the Ansible Tower job template on.
3. Click the **VM Actions** button to show the button you created, and click the button from the list to run the Ansible Tower job template.



4. Click **Submit** to execute the job.

Infrastructure Automation then confirms the job has been executed.

If you selected a service dialog to run when creating the button, Infrastructure Automation will then prompt you to enter variables to complete the task. After entering your desired parameters, Infrastructure Automation takes you to the **Requests** page.

The service item's details can be viewed in menu: **Services > My Services** in Infrastructure Automation.

Embedded Terraform (OpenTofu)

OpenTofu is an open source infrastructure as code tool, which can be used to build, change, and version the infrastructure. You can use OpenTofu to define infrastructure resources in human-readable configuration files that you can use to version, reuse, and share.

OpenTofu is built into Infrastructure Automation so you do not need to install any additional components.

If you want to use the Embedded Terraform feature in Infrastructure Automation that is deployed as a virtual machine appliance, then you need to manually import the Terraform image on an appliance:

Importing OpenTofu image on an appliance

Note: Follow this section if you have Infrastructure Automation that is deployed as a virtual appliance. These steps are not applicable to a containerized deployment (podified) of Infrastructure Automation.

Log in to the registry where the image is stored and import the OpenTofu image on your appliance server:

```
runuser --login manageiq --command 'podman login cp.icr.io -u cp -p <entitlement_key> && podman --
root=/var/www/miq/vmdb/data/containers/storage pull cp.icr.io/cp/cp4waiops/infra-mgmt-opentofu-runner:v4.8.0'
```

You also need to set the docker image name in advanced settings before enabling the server role. Navigate to the **Settings** > **Application Settings** in Infrastructure Automation UI and set the value for **workers/worker_base/opentofu_worker/container_image** field.

An example value of this field is **container_image: cp.icr.io/cp/cp4waiops/infra-mgmt-opentofu-runner:v4.8.0**.

Usage of Embedded Terraform (OpenTofu)

The following sections show the usage of Embedded Terraform in Infrastructure Automation. The following sections apply to Infrastructure Automation that is deployed as a containerized deployment (podified) and Infrastructure Automation that is deployed as a virtual machine appliance:

1. Enable the Embedded Terraform server role.
2. Add a source control repository that contains your templates.
3. Add credentials to access the repository.
4. Create a Service Catalog item with the desired Terraform template.

Enabling the Embedded Terraform Server Role

In Infrastructure Automation, the Embedded Terraform server role is disabled by default. Enable this server role to use Embedded Terraform (OpenTofu).

To enable the Embedded Terraform Server Role, use the following steps:

1. Browse to the settings menu, and click **Configuration** > **Application Settings**.
2. Select the desired server under **Zones**.
3. Set the **Server Role** for **Embedded Terraform** to **On**.

Verifying the Embedded Terraform worker state

Verify that the Embedded Terraform worker is started to use its features:

1. Browse to the settings menu, then click **Configuration** > **Application Settings**.
2. Click **Diagnostics** and click the server that you want to choose.
3. Click **Roles by Servers** tab.

A table of all workers and their status appears from which you can confirm the state of your Embedded Terraform worker.

Adding a Template Repository

To enable Infrastructure Automation to discover and use your Terraform templates, add a repository to store and manage your templates.

If your repository requires credentials for access, then you need to create SCM credentials. For more information about how to create SCM credentials, see SCM credentials.

Use the following steps to add a repository:

1. Browse to the menu and click **Automation > Embedded Terraform > Repositories**.
2. Click **Configuration**, then ⊕ (**Add New Repository**).
3. Provide a Repository Name in the **Name** field.
4. Add a description for the repository in the **Description** field.
5. Add a **URL** and an optional port for the repository.
6. Select the appropriate **SCM Credentials** from the drop-down menu.
7. Provide a branch name in the **SCM Branch** field. This field is optional and default value is set to **master** branch.
8. Click **Save**.

When you save the repository, the Terraform templates are synced, and are available to Infrastructure Automation.

Refreshing Repositories

You can use Infrastructure Automation to refresh specific Terraform repositories or all repositories in your inventory to make sure that your templates are up to date.

Use the following steps to refresh a specific repository:

1. Browse to the menu and click **Automation > Embedded Terraform > Repositories**.
2. Click a repository.

3. Click **Configuration**, then (**Refresh this Repository**).

Alternately, you can refresh some or all of the repositories from the list view:

1. Browse to the menu and click **Automation > Embedded Terraform > Repositories**.
2. Select the repositories that you want to refresh. Click **Check All** to select all repositories.
3. Click **Configuration**, then (**Refresh Selected Ansible Repositories**).

Credentials

Use the following sections to learn more about credentials that are associated with Infrastructure Automation and Embedded Terraform:

Adding Credentials

Credentials are used by Infrastructure Automation for authentication when you connect to cloud providers for infrastructure deployment.

1. Browse to the menu and click **Automation > Embedded Terraform > Credentials**.
2. Click **Configuration**, then (**Add New Credential**).
3. Provide a **Name** for the credential.
4. Select the **Credential Type**. Additional fields might appear depending on the credential type that you chose.
5. Click **Add**.

Credential Types

Each credential type that is used by Infrastructure Automation for the Embedded Terraform is listed in the following sections:

SCM

SCM (source control) credentials are used with projects to clone and update the local source code repositories from a remote revision control system such as Git, Subversion, or Mercurial.

Source Control credentials contain multiple attributes, which you need to configure:

- **Username**: The username for source control system.
- **Password**: The password for source control system.
- **Private key passphrase**: If the SSH private key used is protected by a passphrase, you might need to configure a key passphrase for the private key.
- **Private Key**: Copy or drag-and-drop the actual SSH private key, which is used to authenticate the user to the source control system by using SSH.

Amazon

If you select this credential type, it enables connection between Infrastructure Automation and Amazon Web Services.

Amazon Web Services credentials contain multiple attributes, which you need to configure:

- **Access Key**: User credentials that allow for programmatic calls to Amazon Web Services.
- **Secret Key**: The secret key that corresponds to the user access key.
- **STS Token**: Token generated by Amazon Web Services Security Token Service.
- **AWS region**: Multiple, physically separated and isolated Availability Zones that are connected with low latency, high throughput, and highly redundant networking.

Azure

If you select this credential type, it enables connection between Infrastructure Automation and Microsoft Azure.

Microsoft Azure credentials contain multiple attributes, which you need to configure:

- **Subscription ID**: The Subscription UUID for the Microsoft Azure account.
- **Tenant ID**: The Tenant ID for the Microsoft Azure account.
- **Client Secret**: The Client Secret for the Microsoft Azure account.
- **Client ID**: The Client ID for the Microsoft Azure account.

Google Compute Engine

If you select this credential type, it enables connection between Infrastructure Automation and Google Compute Engine.

Google Compute Engine credentials contain multiple attributes, which you need to configure:

- **Service Account Email Address**: The Service Account email address to connect to the Google Compute Engine.

- **RSA Private Key**: Contents of the PEM file associated with the service account email.
- **Project**: The Google Compute Engine assigned identification. This field is constructed as two words followed by a three-digit number, such as `squeamish-ossifrage-123`.
- **Google Cloud Region**: The default region for the resources. If another region is specified on the resource, it takes precedence.

IBM Cloud Classic Infrastructure

If you select this credential type, it enables connection between Infrastructure Automation and IBM Cloud Classic Infrastructure.

IBM Cloud Classic Infrastructure credentials contain multiple attributes, which you need to configure:

- **IBM Cloud Classic Infrastructure User Name**: The username for IBM Cloud Classic Infrastructure.
- **IBM Cloud Classic Infrastructure API Key**: The API key for IBM Cloud Classic Infrastructure.

OpenStack

If you select this credential type, it enables connection between Infrastructure Automation and OpenStack.

OpenStack credentials contain multiple attributes, which you might need to configure:

- **Username**: The username to connect to OpenStack.
- **Password (API Key)**: The password or API key to connect to OpenStack.
- **Host (Authentication URL)**: The host to be used for authentication.
- **Project (Tenant Name)**: The Tenant name or Tenant ID to connect to OpenStack. This value is usually the same as the username.
- **Domain name**: The Fully qualified domain name (FQDN) to connect to OpenStack.

VMware

If you select this credential type, it enables connection between Infrastructure Automation and VMware vCenter.

Important: If both Infrastructure Automation and a VMware provider are located in the same IPv6-only network, then use a DNS-resolvable hostname for the VMware provider in the **vCenter Host** field when you add the credentials.

VMware credentials contain multiple attributes, which you might need to configure:

- **Username**: The username to connect to vCenter.
- **Password**: The password to connect to vCenter.
- **vCenter Host**: The vCenter hostname or IP address to connect to.

Running a Terraform Template from a Service Catalog

You can run a Terraform Template from Infrastructure Automation by creating a Service Catalog item from a Terraform template.

Use the following listed steps in each section to run the Terraform Template from a Service Catalog:

1. Create a catalog
2. Create a Terraform Service Catalog item
3. Run the Terraform template

Create a catalog

Use the following steps to create a catalog:

1. In the navigation bar, click **Services** > **Catalogs**.
2. Click **Catalogs**.
3. Click **Configuration**, then click **Add a New Catalog**.
4. Enter a **Name** and **Description** for the catalog.
5. Click **Add**.

Create a Terraform Service Catalog item

Use the following steps to create a Terraform Service Catalog item:

1. In the navigation bar click **Automation** > **Embedded Automate** > **Customization**, then click **Service Dialog**.
2. Click **Configuration** > **Create Service Dialog**.
3. Enter a **Service Dialog Name** and add the required fields for the Terraform template.
4. Click **Save**.
5. In the navigation bar, click **Services** > **Catalogs** > **Catalog Items**.
6. Click **Configuration** > **Add a New Catalog Item** to create a new catalog item with the following details at minimum:

- For **Catalog Item type**, select **Terraform Template**.
- Enter a **Name** for the service catalog item.
- Select **Display** in **Catalog**.
- In Catalog, select the catalog that you created previously.
- In Provisioning, select the repository that you previously added and select the Terraform template that you want to deploy.
- In Provisioning, select the **Cloud Type** and then select the **credential** to connect to the cloud.
- In Provisioning, if you want to enter additional information when you run the task, **Service Catalog** must be selected. A dialog is required if **Display in Catalog** is chosen.
- In Provisioning, you have two options to create a Service Dialog:
  - **Use Existing**: When you select **Use Existing** option, you can select the Service Dialog that you previously created.
  - **Create new**: When you select **Create new** option, you can create a new Service Dialog by providing a name. When the catalog item is saved, the new Service Dialog is created with section for **Terraform Template Variables**. The **Terraform Template Variables** section contains input fields for each of the input variables from the Terraform Template. Under the **Terraform Template Variables** section, the boolean field displays as a checkbox. Number field displays as a textbox with regex validation for the number value. JSON fields display as multi-line text area field, with JSON value prettified. String fields display as single line textbox fields.

  To modify the dialog, navigate to **Automation** > **Automate** > **Customization** > **Service Dialogs**, select **Service Dialog**, and from the Configuration menu, select **Edit this dialog** to make the required changes.

7. Click **Save**.

The catalog item that you created appears in the *All Service Catalog Items* list.

Run the Terraform Template:

Use the following steps to run the Terraform Template.

1. In the navigation bar, click **Service** > **Catalogs** > **Service Catalogs** > **created catalog**.
2. Click **Order** for the catalog item.
3. Enter any variables that are requested and click **Submit**.

Infrastructure Automation takes you to the *Requests queue* page and displays the status of the job.

The service item details can be viewed when you navigate to **Services** > **My Services in Infrastructure Automation**.

Retire the Terraform services

When Terraform provisioned service resources are no longer required, the resources can be retired. The retirement action initiates the Terraform backend to run the `destroy` command. This action destroys all the remote resources that are managed by the Terraform configuration.

Use the following steps to retire the service instance immediately:

1. In the navigation bar, click **Service catalog** > **Services**.
2. Click **My Services**.
3. Select the service instance that you want to retire.
4. Click **Lifecycle** > **Retire the Service**.

Use the following steps to set a retirement day and time for a service instance:

1. In the navigation bar, click **Service catalog** > **Services**.
2. Click **My Services**.
3. Click **Lifecycle** > **Set Retirement Dates for this Service**.
4. In **Enter Retirement Date as** field, select **Specific Date and Time** or **Time Delay from Now** to schedule the retirement.
   - To choose a **Specific Date and Time** for Retirement, click the **Retirement Date** to open the calendar:
     - Select a **Retirement Date** by using the calendar control.
     - Select a **Retirement Time** (in UTC) by using the arrows.
   - To retire by using a relative time, select **Time Delay from Now**:
     - From **Time Delay**, specify a retirement time in the form of the number of months, weeks, days, or hours in the using the arrows.
5. Select a **Retirement Warning** if needed.
6. Click **Save**.

Cloud Providers

In Infrastructure Automation, a cloud provider is a cloud computing environment that you can add to a Infrastructure Automation appliance to manage and interact with the resources in that environment. This chapter describes the different types of cloud providers that you can add to Infrastructure Automation, and how to manage them. Most cloud providers are added individually to Infrastructure Automation. Additionally, Amazon EC2 and Azure cloud providers can be discovered automatically by Infrastructure Automation.

The web interface uses virtual thumbnails to represent cloud providers. Each thumbnail contains four quadrants by default, which display basic information about each provider:



1. Number of instances
2. Management system software
3. Number of images
4. Authentication status

Icon	Description
	Validated: Valid authentication credentials have been added.
	Invalid: Authentication credentials are invalid.
	Unknown: Authentication status is unknown or no credentials have been entered.

Provider authentication status

Refreshing Cloud Providers

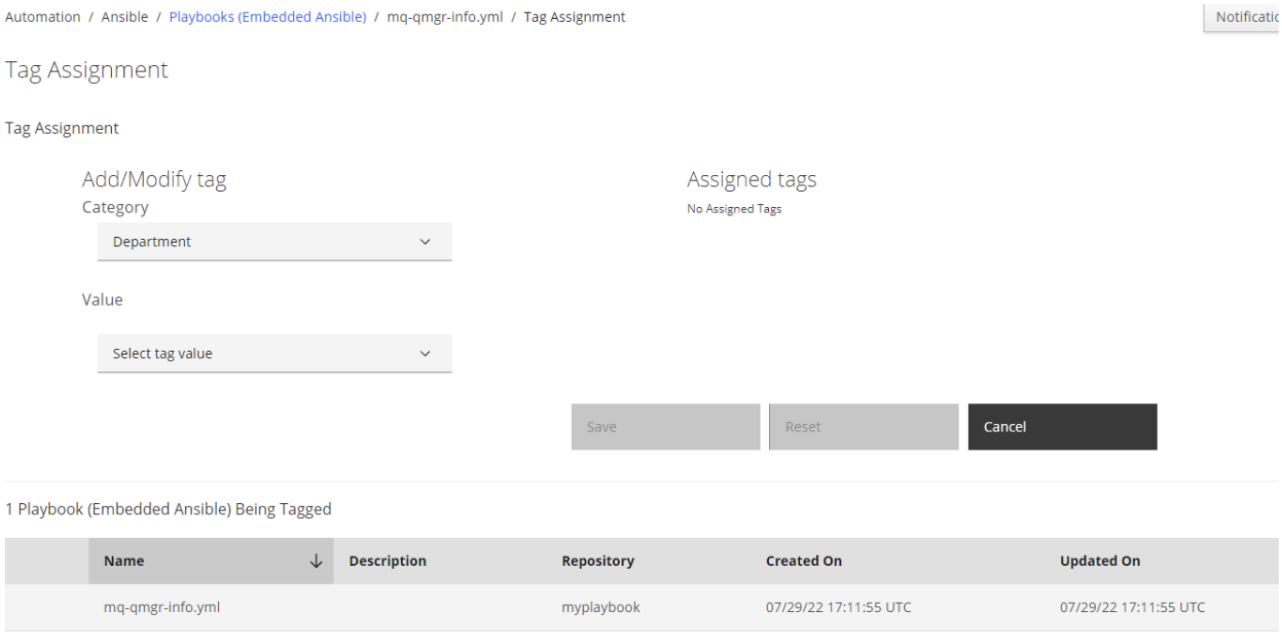
Refresh a cloud provider to find other resources related to it. Ensure the chosen cloud providers have the correct credentials before refreshing.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Select the checkboxes for the cloud providers to refresh.
3. Click **Configuration**, and then (**Refresh Relationships and Power States**).
4. Click **OK**.

Tagging Cloud Providers

Apply tags to all cloud providers to categorize them together at the same time.

1. Navigate to **Compute > Clouds > Providers**.
2. Select the checkboxes for the Cloud Providers to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a customer tag to assign from the first list.



5. Select a value to assign from the second list.
6. Click **Save**.

Removing Cloud Providers

A cloud provider might require removal from the VMDB if it is no longer in use.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Check the cloud providers to remove.
3. Click **Configuration**, and then (**Remove Cloud Providers from the VMDB**).
4. Click **OK**.

Editing a Cloud Provider

Edit information about a provider such as the name, IP address, and login credentials.

Note:

The **Type** value is unchangeable.

To use a different cloud provider, create a new one.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click the cloud provider to edit.
3. Click **Configuration**, and then **(Edit Selected Cloud Provider)**.
4. Edit the **Basic Information**. This varies depending on the **Type** of provider.
5. Fill out the **Credentials** by typing in a **Username**, **Password**, and a verification of this password (**Confirm Password**).
   - If selecting **Amazon EC2**, generate an **Access Key** in the **Security Credentials** of your Amazon AWS account. The **Access Key ID** acts as your **User ID**, and your **Secret Access Key** acts as your **Password**.
   - If selecting **OpenStack**, use the `Keystone User ID` and `Password` for your login credentials.
6. If editing an OpenStack provider, use the **AMQP** subtab to provide credentials required for the Advanced Message Queuing Protocol service on your OpenStack Nova component.
7. Click **Validate** and wait for notification of successful validation.
8. Click **Save**.

Viewing a Cloud Provider's Timeline

View the timeline of events for instances registered to a cloud provider.

1. Navigate to **Compute > Clouds > Providers**.
2. Click the desired cloud provider for viewing the timeline.
3. Click (**Monitoring**), and then (**Timelines**).
4. From **Options**, customize the period of time to display and the types of events to see.
    - Use **Show** to select regular Management Events or Policy Events.
    - Use the **Type** list to select hourly or daily data points.
    - Use **Date** to type the date for the timeline to display.
    - If you select to view a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days.
    - The three **Event Groups** list allow you to select different groups of events to display. Each has its own color.
    - From the **Level** list, select a **Summary** event, or a **Detail** list of events.

- Azure Providers
- Amazon EC2 Providers
- Google Compute Engine Providers
- IBM CIC Providers
- IBM Cloud VPC Provider
- IBM Power Systems Virtual Servers Providers
- IBM PowerVC Providers
- OpenStack Providers
- Oracle Cloud Providers

Physical Infrastructure Providers

In Infrastructure Automation, a physical infrastructure provider is a management console used to manage physical inventory such as physical servers, chassis, racks, and physical networking/storage devices.

ManageIQ supports the following types of physical infrastructure providers:

- Lenovo XClarity
- Redfish

Image Import Workflows

IBM PowerVC to IBM Power Systems Virtual Server

The image import functionality allows you to easily move an image from your on-prem PowerVC environment to an off-prem Power Systems Virtual Server environment using Infrastructure Automation web interface. The enablement of the workflow in Infrastructure Automation requires you to perform certain preparations that are described in detail below.

Preparing Infrastructure Automation hosting server (required once):

The following commands are to be run on a server (or in a corresponding docker container) that is dedicated to hosting your Infrastructure Automation instance.

1. Install `ansible-runner` application using the installation instructions. The application must be available in the the command line of the user under which the Infrastructure Automation web-server is running.

Preparing IBM PowerVC Server (required once):

The following commands are to be run on a PowerVC server that is dedicated to supporting the image import workflow.

1. Make sure that the minimal IBM PowerVC version you are using is 1.4.4.

2. Install **python3.X**, **pip3**, **virtualenv** applications as well as the **libselinux-python3** library. You can achieve this by issuing the following command as a "root" user:

   ```
   yum install -y python3 python-pip3 python-virtualenv libselinux-python3 gcc
   ```

   The standard **python3.X** installation will be used in order to run auxiliary scripts, which will upload PowerVC disk images (in Open Virtualization Format) to the IBM Cloud Object Storage. The **pip3** is required in order to install Cloud Object Storage Python SDK and auxiliary packages for proper functioning of the import workflow. The role of **virtualenv** will be to isolate python package installations that are relevant only to the image import workflow from the Python system packages. The **libselinux-python3** library is needed for the proper functioning of Ansible's **copy** module, which will be used for the copying of auxiliary data between the Infrastructure Automation hosting site and the PowerVC server. Presence of **gcc** is needed as a dependency of one of the Python packages (**pycryptodome**) that will be installed on later steps.

3. Prepare enough local storage for holding transient PowerVC disk images that will be transported into Power Systems Virtual Server and deleted from a local storage upon workflow completion. A typical PowerVC image file size can be many GBs. As an example, a RHEL 8.3 disk image can be of 20GB size or more. You would therefore require more space in your local storage than the mentioned disk image size in order to support the import workflow. You would also need enough space to hold Python virtual environment described in the next steps. Its size however will be no more than 100MB. Additionally, import session will place auxiliary files into the sessions directory during the workflow execution but their collective size will not exceed 5MB. In summary, if you're attempting to import an image whose size is 'G' Gigabytes, then having 'G+1' Gigabytes of free space in that directory would be sufficient.

4. Create an empty sessions directory **/home/sessions** which will have to be located on the local storage with sufficient space as described in the previous step. In addition, make sure that already existing **/var/opt/ibm/powervc/imgstaging** directory also has at least G+1 Gigabytes of free storage as described in previous paragraph since IBM PowerVC internally uses it as a temporary staging area during export of images to the local filesystem.

   ```
   mkdir /home/sessions
   ```

5. The image import workflow will use Ansible via SSH from the Infrastructure Automation server in order to execute commands on the PowerVC server. At this point you need to choose a system user that will be made available to Infrastructure Automation instance for connection to the PowerVC server via SSH. Make sure the user has a read/write access to the **/home/sessions** directory. The user must also be able to execute **powervc-image** application through the command line. In general, a good security practice would be to permit user only as much as it is required to run the import workflow. We will assume that all further commands in this section are executed using this existing or newly created system user:

   ```
   su - your_user
   ```

6. Create a separate virtual Python environment:

   ```
   virtualenv --system-site-packages /home/sessions/venv
   ```

   This environment will hold Python packages that are relevant only to the image import workflow and as such will not be available system-wide. Conversely, the **--system-site-packages** will indicate to the application that the existing Python system packages are to be made available to the virtual environment.

7. Installing packages in the environment requires one to first activate the environment after which all the Python package installations via **pip3** will become environment relative only. Therefore, activate the newly created Python environment:

   ```
   source /home/sessions/venv/bin/activate
   ```

8. Install Python packages inside the virtual environment using:

   ```
   pip3 install pathlib pycryptodome ibm-cos-sdk
   ```

   The **pathlib** is used for handling file paths. The **pycryptodome** is required for encrypting/decrypting IBM Cloud connection credentials and the **ibm-cos-sdk** represents the IBM Cloud Object Storage Python SDK.

9. Upon completing the Python package installations above you should exit the virtual environment by running:

   ```
   deactivate
   ```

10. Make sure to have a valid PowerVC connection/resource file under **/opt/ibm/powervc/powervcrc** (or at your custom location) containing similar definitions (the values below represent an example and not a suggestion for usage).

    ```
    export OS_IDENTITY_API_VERSION=3
    export OS_AUTH_URL=https://host:5000/v3/
    export OS_CACERT=/etc/pki/tls/certs/powervc.crt
    export OS_REGION_NAME=RegionOne
    export OS_PROJECT_DOMAIN_NAME=Default
    export OS_PROJECT_NAME=ibm-default
    export OS_TENANT_NAME=$OS_PROJECT_NAME
    export OS_USER_DOMAIN_NAME=Default
    export OS_USERNAME=root
    export OS_PASSWORD=password
    export OS_COMPUTE_API_VERSION=2.46
    export OS_NETWORK_API_VERSION=2.0
    export OS_IMAGE_API_VERSION=2
    export OS_VOLUME_API_VERSION=2
    ```

11. The last step is to test the exporting of PowerVC disk image to the local filesystem. Firstly, execute:

    ```
    source /opt/ibm/powervc/powervcrc
    ```

    This will acquire all the environmental variables required for issuing commands to the corresponding PowerVC instance. Now run the following command:

    ```
    powervc-image list
    ```

    The output should be similar to the one below:

    ```
    +----------+--------------------------------------+--------+---------+------+-------------+--------------+-----------
    +------------+
    | Name     | ID                                   | Status | Volumes | Size | Description | Architecture | OS Distro
    ```

```
| Exportable |
+----------+--------------------------------------+--------+---------+------+-------------+--------------+-----------
+------------+
| test-IMG | XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX | active | 1       | 20   |             | ppc64        | rhel
| True       |
+----------+--------------------------------------+--------+---------+------+-------------+--------------+-----------
+------------+
```

Note the name of the image you would like to export, which will be **`test-IMG`** in our example. Next export the image of your choice:

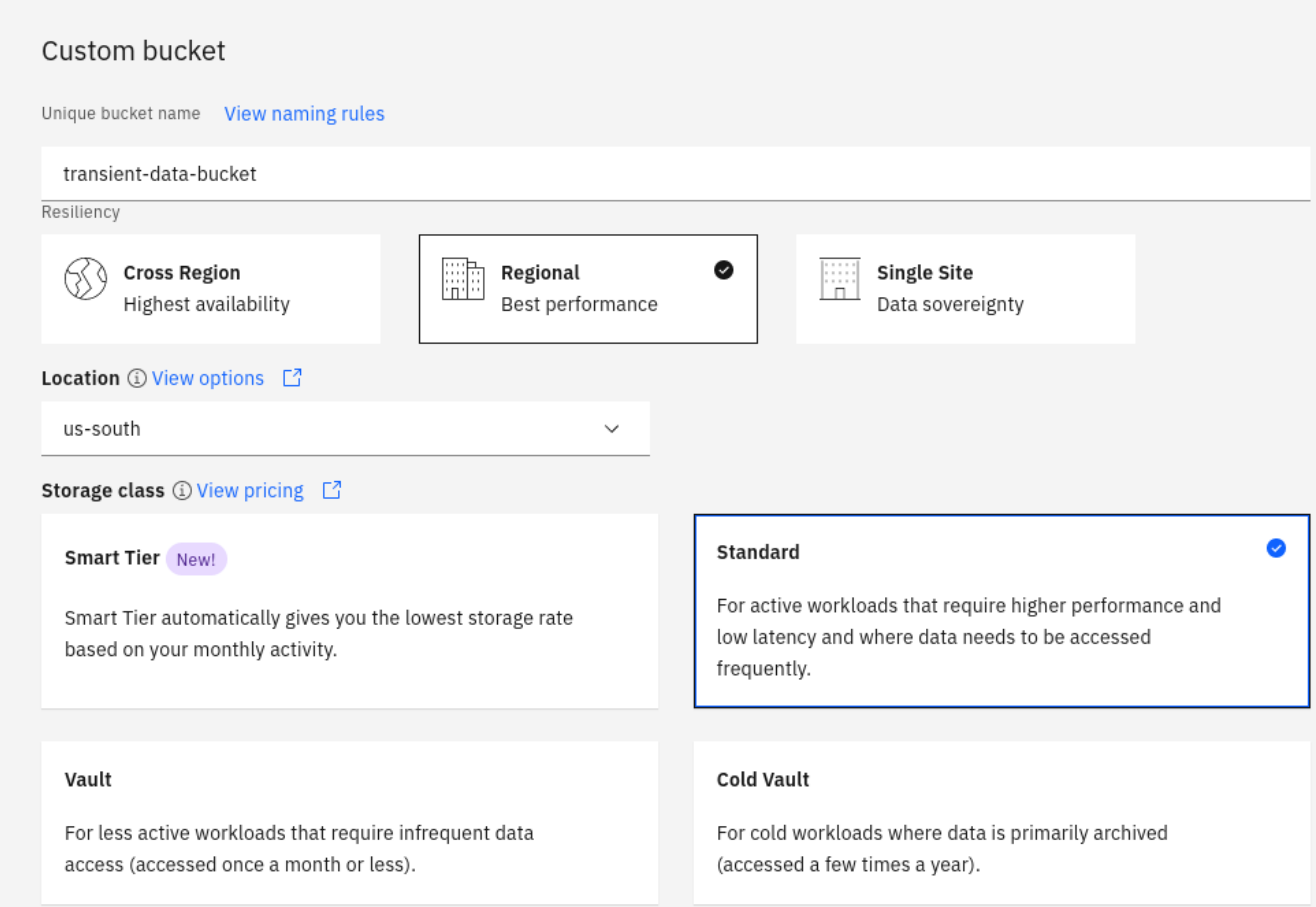
```
powervc-image export -i test-IMG -p /home/sessions/image.ova
```

It is a time-consuming operation with the runtime taking several minutes to complete depending on the size of exported image. Once the operation completes you should be able to find the exported image under **`/home/sessions/image.ova`**. At this stage you can remove the exported image thus concluding the test:

```
rm /home/sessions/image.ova
```

Add a Cloud Object Storage provider in Infrastructure Automation:

1. See the corresponding instructions here.
2. Create a bucket in your Cloud Object Storage that will hold the transient OVA image files.



Make sure to choose the region for the bucket that **corresponds** to the region that you have specified when registering the Cloud Object Storage provider.

NOTE: The “root” user as well as the workflow dedicated user of the PowerVC server from step 5 would both be able to decrypt the Cloud Object Storage’s credentials during the execution of the import workflow and therefore make sure to take this into consideration in your security setup.

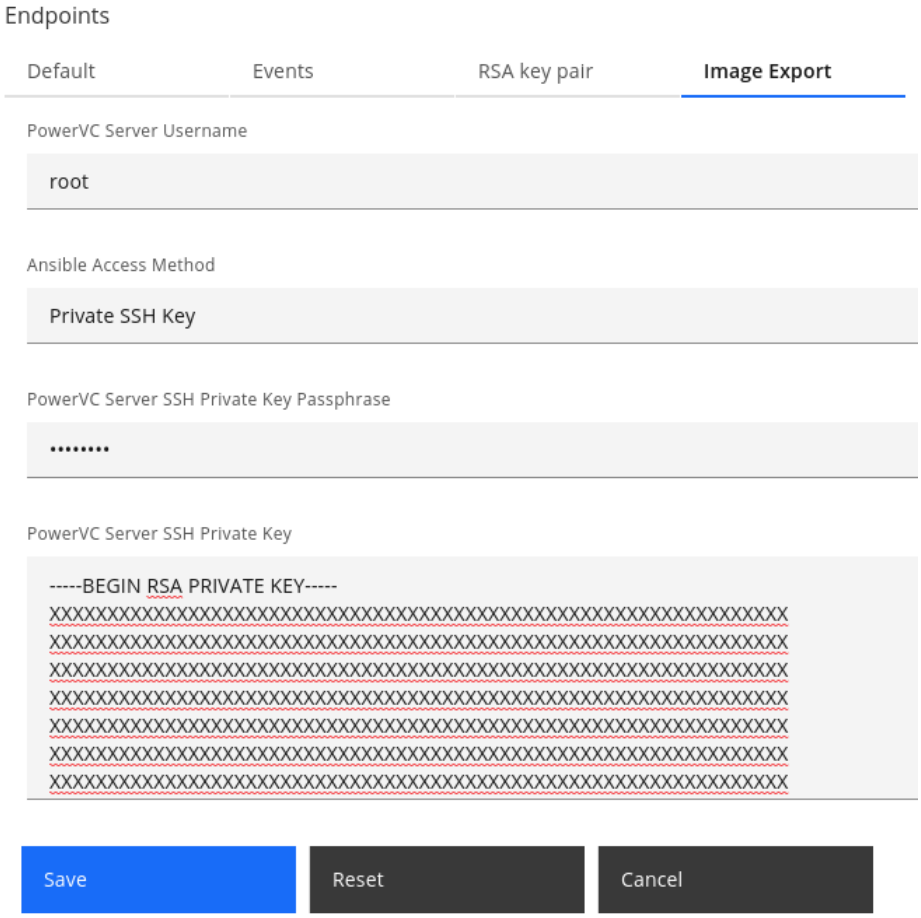
Add an IBM Power Systems Virtual Server provider in Infrastructure Automation:

1. See the corresponding instructions here.

Add an IBM PowerVC provider in Infrastructure Automation:

1. See the corresponding instructions here.
2. Specify the image import related parameters on the PowerVC registration form.
   - Enter the SSH username for access to the PowerVC server. It corresponds to the user on your PowerVC server that is dedicated to running image import related commands via Ansible as described in 5.

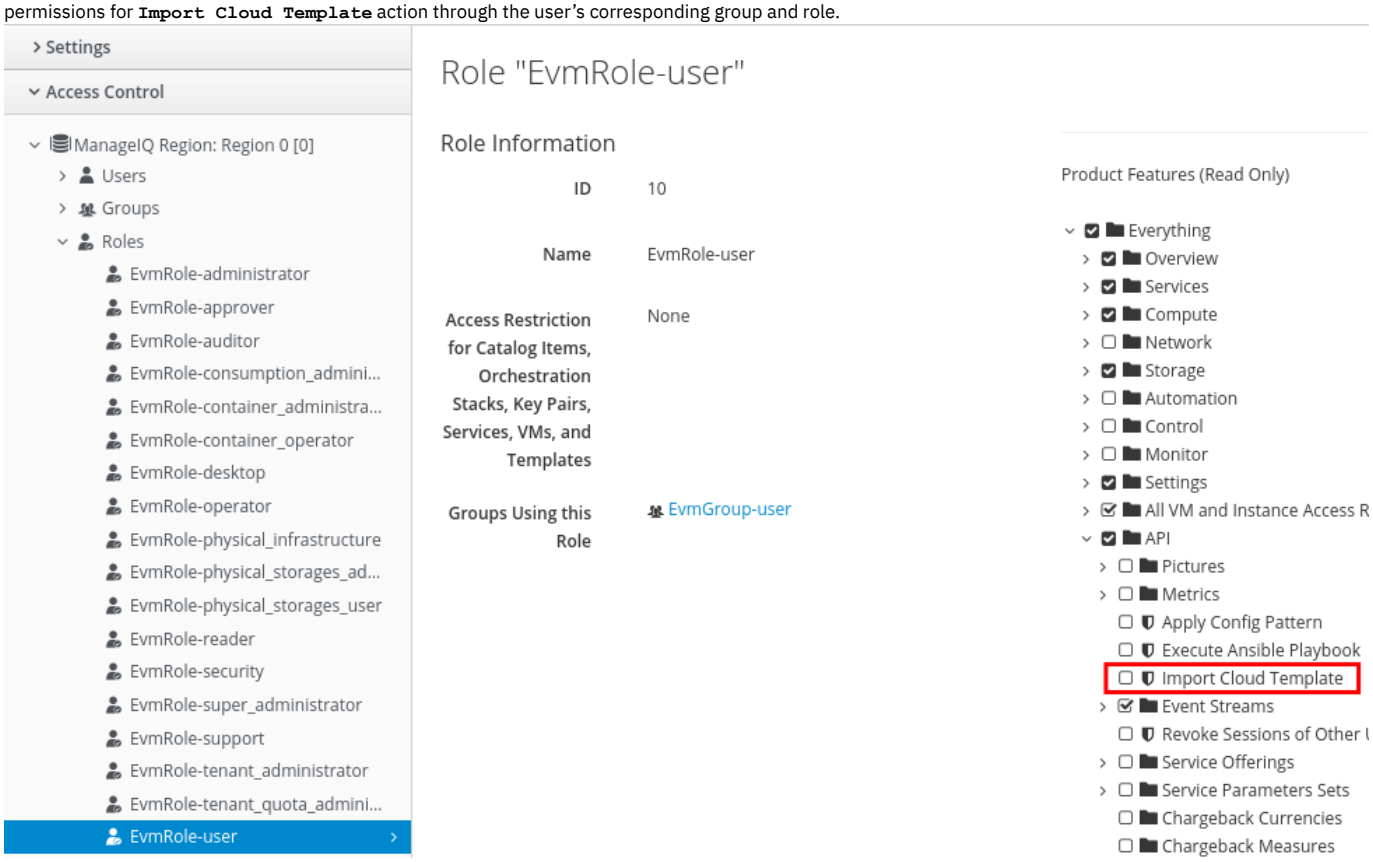
- Enter the private key and the private key passphrase (if present).



Grant Image Import Permissions in Infrastructure Automation:

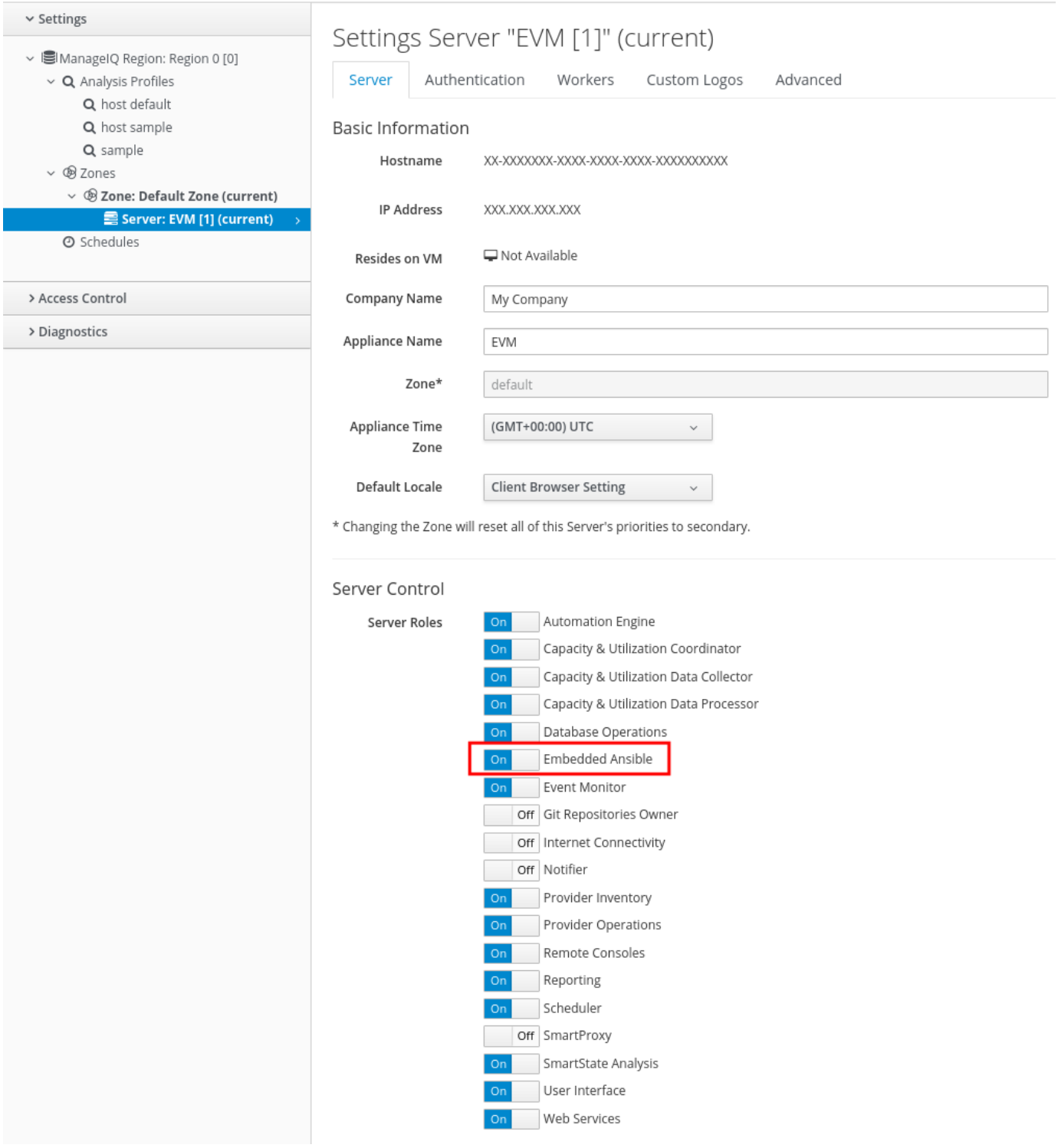
User performing image import needs a corresponding permission in Infrastructure Automation in order to perform this operation. For granting permissions, log-in through an administrative account and navigate to **`Settings -> Application Settings -> Access Control`**. Make sure the user of your choice has

permissions for **`Import Cloud Template`** action through the user's corresponding group and role.



Enable Embedded Ansible in Infrastructure Automation:

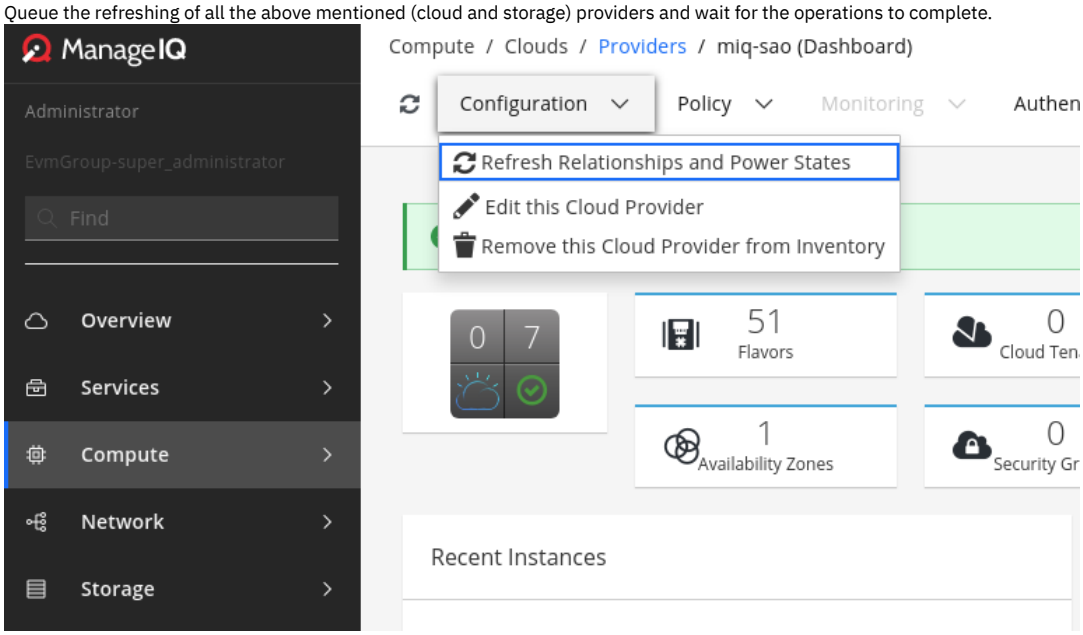
Currently the workflow event is queued as Embedded Ansible playbook upon request submission and as such the corresponding functionality has to be enabled in Infrastructure Automation by navigating to: **`Settings -> Application Settings -> Settings`**



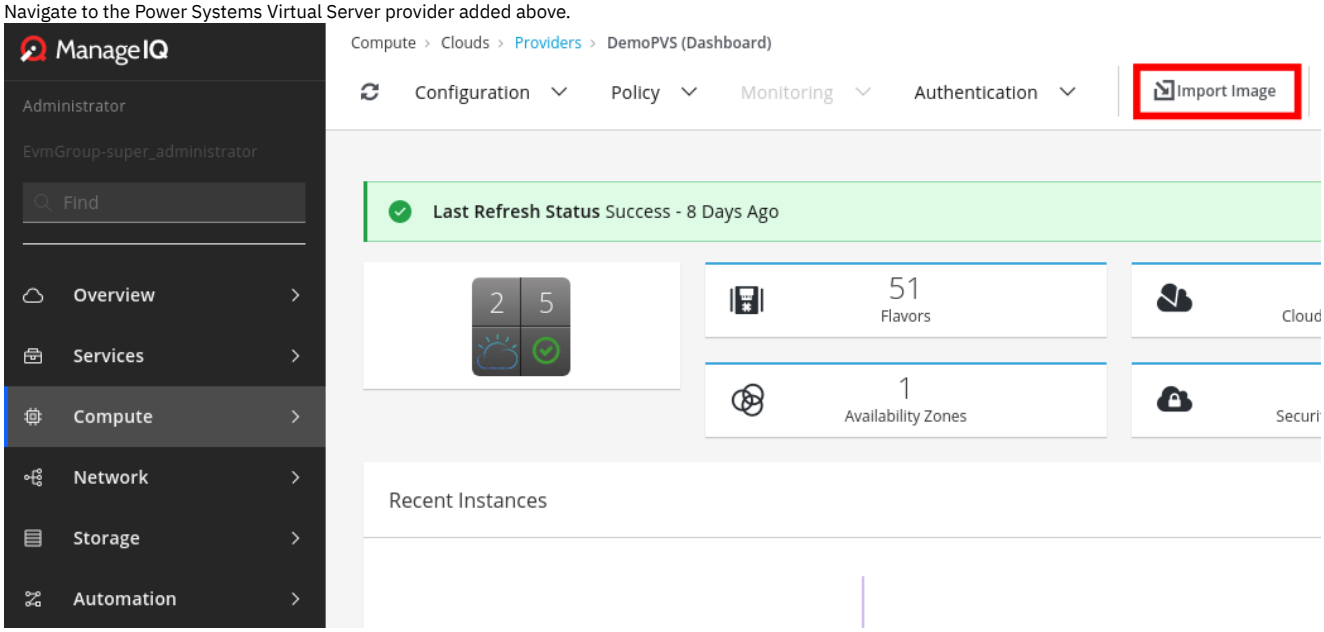
Start the workflow

The workflow operates by exporting the image as an OVA file onto a PowerVC local storage, uploading it to the Cloud Object Storage bucket and then transferring it into Power Systems Virtual Server image registry. In order to initiate the workflow:

1. Queue the refreshing of all the above mentioned (cloud and storage) providers and wait for the operations to complete.

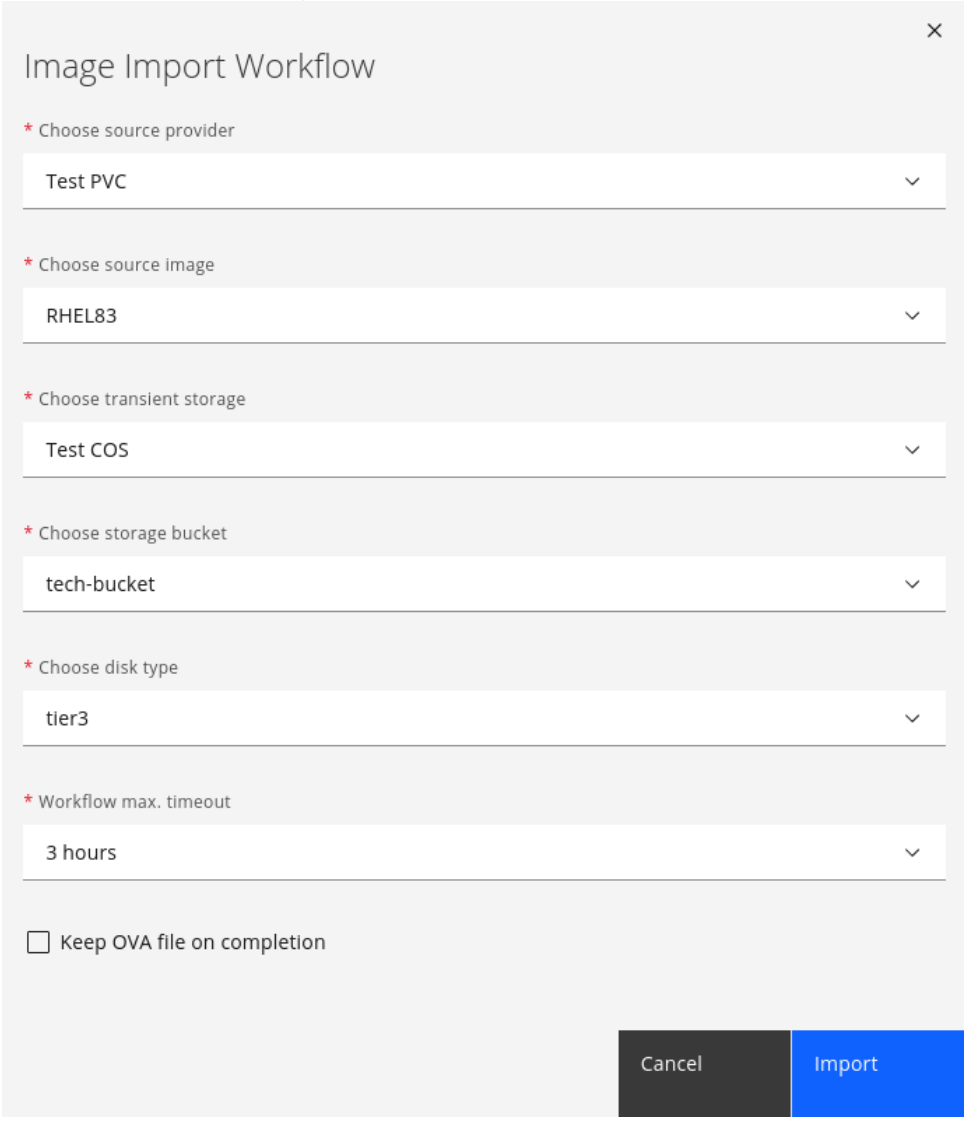


2. Navigate to the Power Systems Virtual Server provider added above.



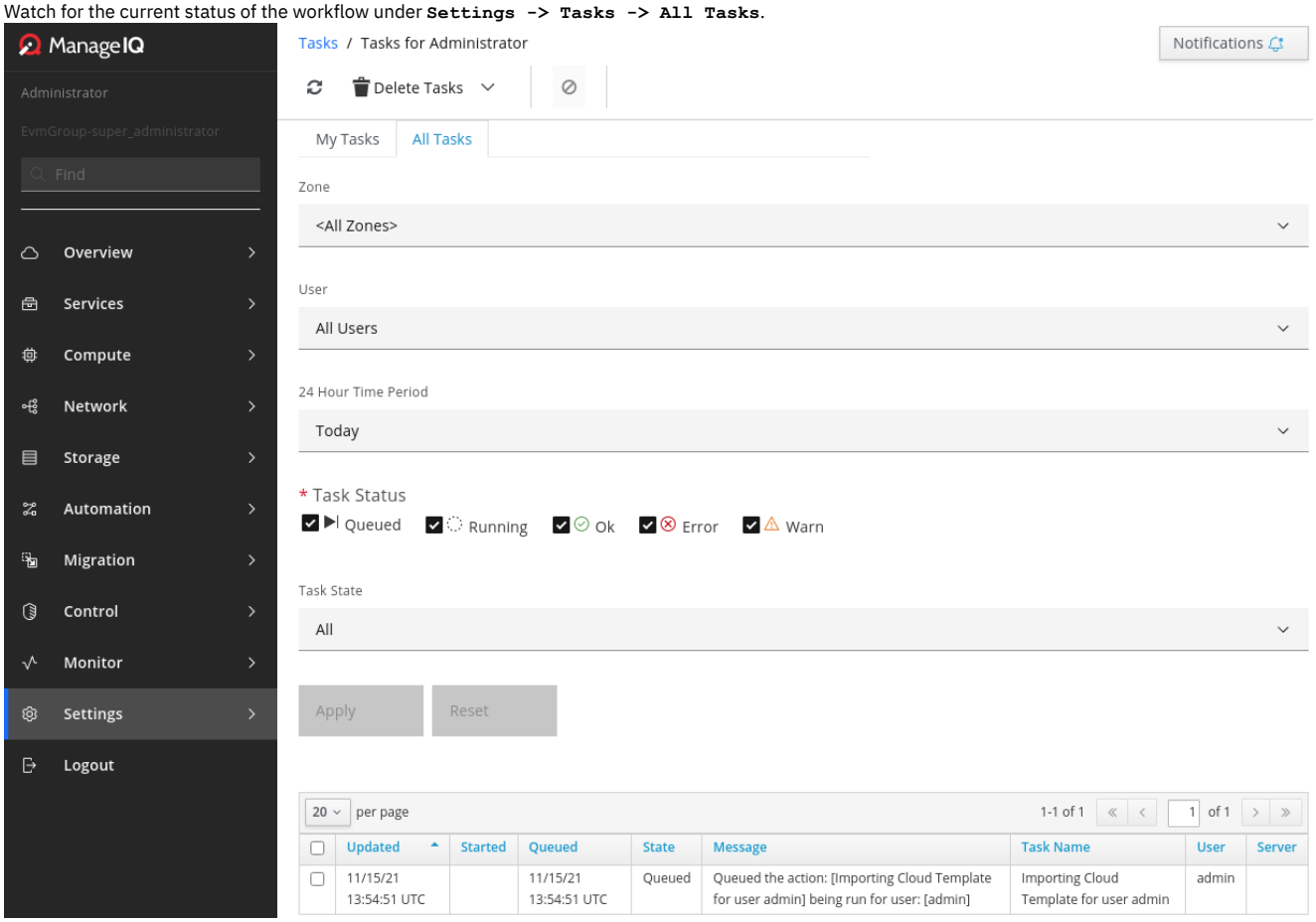
3. Click on the **`Import Image`** button after which an import form will appear.
4. Choose the PowerVC provider added above as a source provider.
5. Choose the image you would like to import.
6. Choose the Cloud Object Storage provider added above.
7. Choose the transient bucket created above.
8. Choose target disk type for the image being imported.
9. Choose the timeout value for your request (default is 3 hours).
10. Check the corresponding box if you wish to keep the OVA file in the cloud bucket upon workflow completion.

11. Click on the **`Import`** button initiating the workflow.



12. The workflow can take an hour or more depending on the image size, bucket location, Power Systems Virtual Server and PowerVC internal load.

13. Watch for the current status of the workflow under **`Settings -> Tasks -> All Tasks`**.



14. Initiate the refreshing of the Power Systems Virtual Server provider upon workflow completion and wait for the newly imported image to appear.

Troubleshooting

- If the Infrastructure Automation UI shows no detailed error description then make sure to check the latest logs with **`journalctl -t evm`** or **`oc logs`** on a server or pod that hosts your Infrastructure Automation instance.
- If **`Infrastructure Automation`** server is terminated during the workflow execution then certain residual objects will have to be removed manually. These are potentially the session directory on the PowerVC server under **`/home/sessions`**, the already uploaded OVA image file in the Cloud Object Storage bucket or its parts as explained here (see **`Abort incomplete multipart uploads`**).
- Currently only one import workflow for a given PowerVC <–> Power Systems Virtual Server pair is allowed. Any newly submitted import workflow, where at least one of the instances in the pair is involved with some other import workflow request, will be queued until the latter completes. You would need to cancel such already executing workflow in order to force immediate execution of the newly submitted one.

Network Managers

In Infrastructure Automation, a network manager is an inventory of networking entities on existing cloud and infrastructure providers managed by your Infrastructure Automation appliance.

This provider type exposes software-defined networking (SDN) providers including *OpenStack Network (Neutron)*, *Azure Network*, and *Amazon EC2 Network*, which enables software-defined networking inventory collection. The OpenStack Network provider collects inventory of floating IPs from OpenStack so that IPs can be allocated without querying OpenStack database every time. Also, it refreshes all Neutron data from both OpenStack and OpenStack Infrastructure, and extracts the Neutron logic to a shared place. Note that management via the network providers configuration is currently disabled.

This chapter describes the different types of network managers available to Infrastructure Automation, and how to manage them. Network managers are discovered automatically by Infrastructure Automation from other connected providers.

Adding or Viewing Network Providers

Note:

All supported network providers—OpenStack Network, Azure Network, and Amazon EC2 Network, are added or removed automatically upon adding or removing the respective cloud provider.

Viewing network providers:

1. Browse to menu: **Networks > Providers** to see a list of all network providers, along with information such as *Name*, *Type*, *EVM Zone*, *Number of Instances*, *Subnets*, and *Region*.
2. Click on a provider from the list to view its summary screen.

Network providers summary:

The summary screen includes tables containing information on *Properties*, *Status*, *Relationships*, *Overview*, and *Smart Management*. Click on rows in the *Relationship* and *Overview* tables to see detailed information for individual entities.

Accordion tabs in the sidebar provide access to **Properties** and **Relationships** details.

Click on **Reload**, **Configuration**, **Policy**, and **Monitoring** actions in the taskbar to manage the selected provider.

Note:

Alternatively, click on a cloud provider to see the cloud provider details and its relationships such as Network Manager, Tenants, Instances among others. In Relationships, click Network Manager to see information about the network provider, and its relationship with the cloud provider, on the summary page.

Refreshing Network Providers

Refresh a network provider to find other resources related to it. Ensure the selected network providers have the correct credentials before refreshing.

1. Browse to menu: **Networks > Providers**.
2. Select the network providers to refresh.
3. Click **Configuration**, and then (**Refresh Relationships and Power States**).
4. Click **OK**.

Tagging Network Providers

Apply tags to network providers to categorize them together at the same time.

1. Navigate to **Networks > Providers**.
2. Select the network providers to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. **Select a customer tag to assign** from the first list.
5. Select a value to assign from the second list.
6. Click **Save**.

Removing Network Providers

Although network providers are added or removed automatically upon adding or removing the respective cloud provider, you can manually remove a network provider if it is no longer in use. This will remove the network provider from the VMDB and any relationship with the cloud provider.

1. Browse to menu: **Networks > Providers**.
2. Click the network provider to remove.
3. Click **Configuration**, and then (**Remove this Network Provider from the VMDB**).
4. Click **OK**.

Viewing a Network Provider's Timeline

View the timeline of events for instances registered to a network provider.

1. Navigate to **Networks > Providers**.
2. Click the network provider you want to monitor the timeline for.

3. Click (**Monitoring**), and then (**Timelines**).
4. From **Options**, select the event type and interval, and customize the period of time to display and the types of events to see.
    - Select *Management Events* or *Policy Events* from the **Show** list.
    - Select an **Interval** between *Hourly* and *Daily*.
    - Select **Date**.
    - If you selected *Daily* for **Interval**, set the number of days in the past to see the event timeline for. The maximum is *31 days back*.
    - Select **Summary** or **Detail** for **Level**.
    - Select the required **Event Groups** from the lists you want to monitor the timeline for.

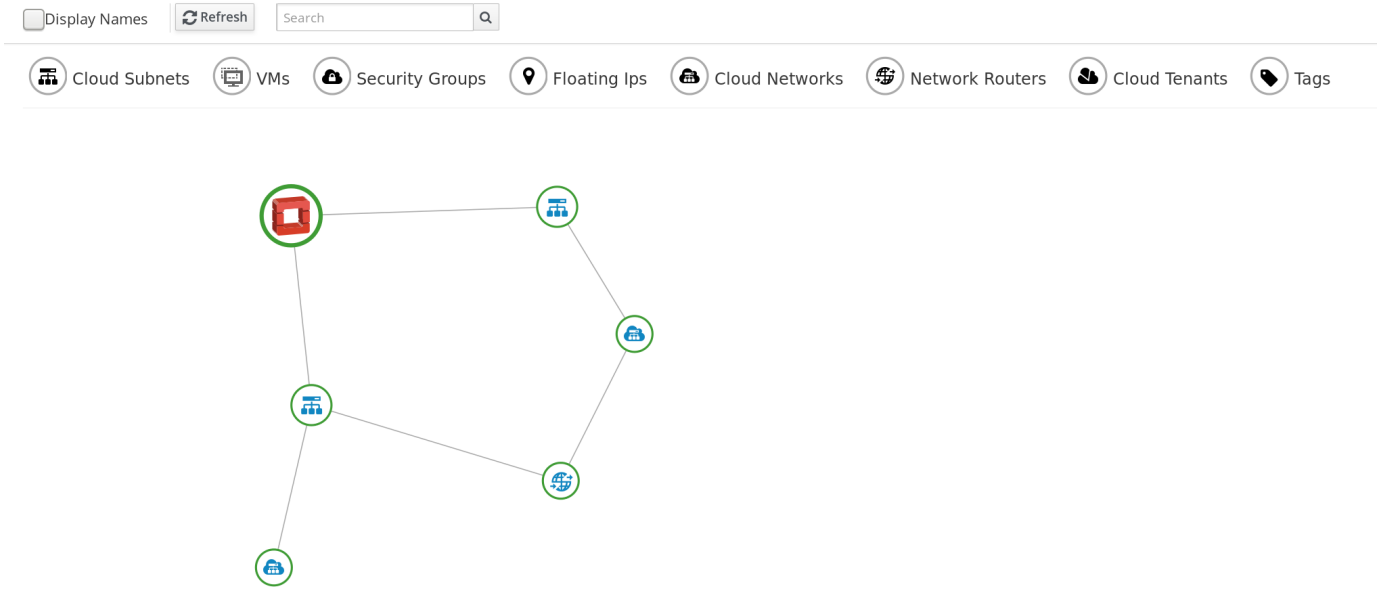
You can also assign policy profiles to network providers, or remove them. The method for doing so is similar to that of any normal policy profile.

Using the Topology Widget for Network Providers

The **Topology** widget is an interactive topology graph, showing the status and relationships between the different entities of the network providers that Infrastructure Automation has access to.

The topology graph includes cloud subnets, virtual machines, security groups, floating IP addresses, cloud networks, network routers, cloud tenants, and tags within the overall network provider environment.

Each entity in the graph displays a color indication of its status: green indicates an active entity, while red indicates inactivity or an issue.



1. Browse to menu: **Networks > Topology**.
2. Click the desired network provider for viewing the provider summary.

Alternatively, you can open the topology widget from the provider summary page by clicking **Topology** under **Overview**.

- Hovering over any individual graph element will display a summary of details for the individual element.
- Double-click an entity in the graph to browse to its summary page.
- Drag elements to reposition the graph.
- Click the symbols in the legend at the top of the graph to show or hide entities.
- Click the **Display Names** checkbox to show or hide entity names.
- Click **Refresh** to refresh the display of the network provider entities.
- Enter a search term in the **Search** box to locate an entity by full or partial name.

Containers Providers

A containers provider is a service that manages container resources, that can be added to the Infrastructure Automation appliance.

Infrastructure Automation can connect to OpenShift Container Platform containers providers and manage them similarly to infrastructure and cloud providers. This allows you to gain control over different aspects of your containers environment and answer questions such as:

- How many containers exist in my environment?
- Does a specific node have enough resources?
- How many distinct images are used?
- Which image registries are used?

When Infrastructure Automation connects to a container's environment, it collects information on different areas of the environment:

- Entities such as pods, nodes, or services.
- Basic relationships between the entities, for example: Which services are serving which pods?
- Advanced insight into relationships, for example: Which two different containers are using the same image?
- Additional information, such as events, projects, routes, and metrics.

You can manage policies for containers entities by adding tags. All containers entities except volumes can be tagged.

Note:

This chapter provides details on managing containers providers. For details on working with the resources within a container environment, see Container Entities in *Managing Infrastructure and Inventory*.

The Infrastructure Automation user interface uses virtual thumbnails to represent containers providers. Each thumbnail contains four quadrants by default, which display basic information about each provider:



1. Number of nodes
2. Container provider software
3. Power state
4. Authentication status

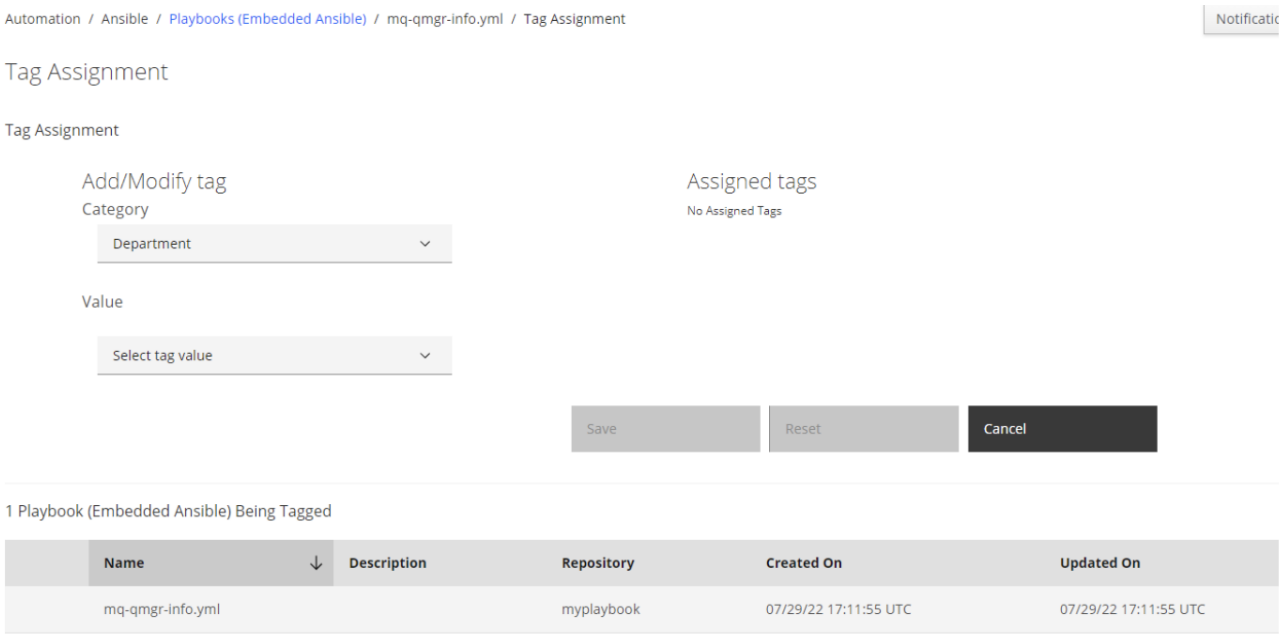
Icon	Description
	Validated: Valid authentication credentials have been added.
	Invalid: Authentication credentials are invalid.
	Unknown: Authentication status is unknown or no credentials have been entered.

Containers provider authentication status

Tagging Containers Providers

Apply tags to all containers providers to categorize them together at the same time.

1. Navigate to **Compute > Containers > Providers**.
2. Select the checkboxes for the containers providers to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a tag to assign from the drop-down menu.



5. Select a value to assign.
6. Click **Save**.

Removing Containers Providers

You may want to remove a containers provider from the VMDB if the provider is no longer in use.

1. Browse to menu: **Compute > Containers > Providers**.
2. Select the checkboxes for the containers providers to remove.
3. Click **Configuration**, and then (**Remove Containers Providers from Inventory**).
4. Click **OK**.

Editing a Containers Provider

Edit information about a provider such as the name, hostname, IP address or port, and credentials as required. If you have just upgraded your Infrastructure Automation environment from an older version, edit the provider to specify the authentication method the provider uses to connect to Infrastructure Automation.

1. Browse to menu: **Compute > Containers > Providers**.
2. Click the containers provider to edit.
3. Click **Configuration**, and then (**Edit Selected Containers Provider**).
4. Edit the **Name** if required.

   Note:

   The **Type** value is unchangeable.
5. Under **Endpoints** in the **Default** tab, edit the following as required:
   1. Select a **Security Protocol** method to specify how to authenticate the provider:
      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
      - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.

        Note:

        You can obtain your OpenShift Container Platform provider's CA certificate for all endpoints (default, metrics, alerts) from `/etc/origin/master/ca.crt`. Paste the output (a block of text starting with `-----BEGIN CERTIFICATE-----`) into the **Trusted CA**

Certificates field.

- **SSL without validation**: Authenticate the provider insecurely (not recommended).

2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.

   Important:

   The **Hostname** must use a unique fully qualified domain name.

3. Enter the **API Port** of the provider. The default port is **8443**.
4. Enter a token for your provider in the **Token** box.

   Note:

   To obtain a token for your provider, run the **oc get secret** command on your provider; see Obtaining an OpenShift Container Platform Management Token.

   For example:

   # oc get secret –namespace management-infra management-admin-token-8ixxs –template='' | base64 –decode

5. Click **Validate** to confirm that Infrastructure Automation can connect to the OpenShift Container Platform provider.

6. Under **Endpoints** in the **Metrics** tab, configure the following for gathering capacity and utilization metrics for Prometheus based on the selection:

   1. Select a **Security Protocol** method to specify how to authenticate the provider:

      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
      - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
      - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)

   2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.
   3. Enter the **API Port** if your provider uses a non-standard port for access. The default port is **443**.
   4. Click **Validate** to confirm that Infrastructure Automation can connect to the endpoint.

7. Under **Endpoints** in the **Alerts** tab, configure the following for Prometheus alerting from the cluster.

   - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
   - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
   - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)
     - Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.
     - Enter the **API Port** if your provider uses a non-standard port for access. The default port is **443**.
     - Click **Validate** to confirm that Infrastructure Automation can connect to the endpoint.

8. Click **Save**.

Hiding Environment Variables for Containers Providers

You can restrict users from viewing container provider environment variables by configuring user roles.

This is useful as the environment variables panel can expose sensitive information, such as passwords, that you may not want certain users to view.

Note:

The default user roles in Infrastructure Automation are read-only. To customize a role's settings, create a new role or a copy of an existing role.

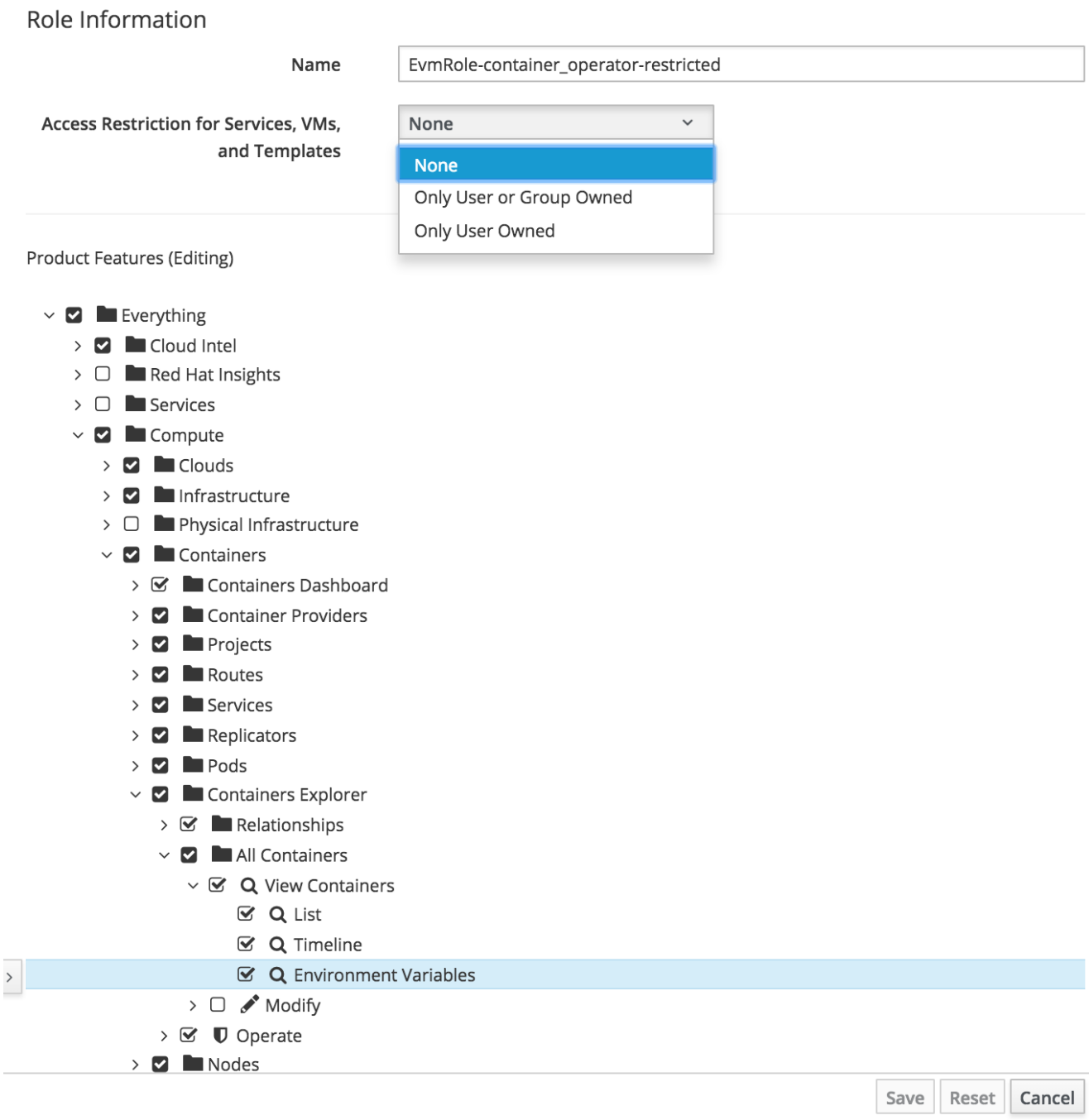
You can view role information and the product features the role can access (marked by a checkmark) by clicking on any role in **Access Control**. Expand the categories under **Product Features** to see further detail.

To configure user access to container environment variables:

1. Click (**Configuration**).
2. Click the **Access Control** accordion, then click **Roles**.
3. Select a existing custom role from the **Access Control Roles** list, and click (**Configuration**), then (**Edit the selected Role**).

Alternatively, to create a new custom role, select a role from the **Access Control Roles** list, and click (**Configuration**), then (**Copy this Role to a new Role**).

4. Edit the name for the role if desired.
5. For **Access Restriction for Services, VMs, and Templates**, select if you want to limit users with this role to only see resources owned by the user or their group, owned by the user, or all resources (**None**).
6. Expand the **Product Features (Editing)** tree options to show **Everything > Compute > Containers > Containers Explorer > All Containers > View Containers**.
7. Clear the **Environment Variables** checkbox to restrict the user role from viewing container environment variables.



8. Click **Save**.

For more information about user roles, see Roles in *General Configuration*.

Viewing a Containers Provider's Timeline

View the timeline of events for instances registered to a containers provider.

1. Navigate to **Compute > Containers > Providers**.
2. Click the desired containers provider for viewing the timeline.
3. Click  (**Monitoring**), and then  (**Timelines**).
4. From **Options**, customize the period of time to display and the types of events to see.
    - Use **Show** to select regular Management Events or Policy Events.
    - Use the **Interval** dropdown to select hourly or daily data points.
    - Use **Date** to type the date for the timeline to display.
    - If you select to view a daily timeline, use **Show** to set how many days back to go. The maximum history is 31 days.
    - From the **Level** dropdown, select a **Summary** event, or a **Detail** list of events.
    - The three **Event Groups** dropdowns allow you to select different groups of events to display. Each has its own color.

Click on an item for more detailed information.

- Azure Kubernetes Providers
- Red Hat OpenShift Providers
- IBM Cloud Kubernetes Service Providers
- Oracle Kubernetes Engine Providers
- VMware Tanzu Providers

Storage Managers

In Infrastructure Automation, a storage manager is a service providing storage resources that you can manage from a Infrastructure Automation appliance. This chapter describes the different types of storage managers used by Infrastructure Automation, and how they are added to Infrastructure Automation.

There are three types of storage managers currently available to Infrastructure Automation:

- Amazon Elastic Block Store Managers
- OpenStack Block Storage Managers
- OpenStack Object Storage Managers
- IBM Cloud Object Storage Managers

Appendix

Using a Self-Signed CA Certificate

Adding a self-signed Certificate Authority (CA) certificate for SSL authentication requires more configuration on OpenStack Platform and Microsoft System Center Virtual Machine Manager (SCVMM) providers.

Note: This procedure is not required for OpenShift Container Platform, or Red Hat Virtualization, which have the option to select **SSL trusting custom CA** as a **Security Protocol** in the user interface. These steps are needed only for providers without this option in the user interface.

Before adding the provider, configure the following:

1. Copy your provider's CA certificate in PEM format to **`/etc/pki/ca-trust/source/anchors/`** on your Infrastructure Automation appliance.
2. Update the trust settings on the appliance:

    ```
    # update-ca-trust
    ```

3. Restart the EVM processes on the server:

    ```
    # rake evm:restart
    ```

The CA certificate is added to the appliance, and you can add the provider to Infrastructure Automation.

- Red Hat Virtualization Providers
    - Enabling Red Hat Virtualization Capacity and Utilization Data Collection
    - Adding a Red Hat Virtualization Provider
    - Authenticating Red Hat Virtualization Hosts

Red Hat Virtualization Providers

To use a Red Hat Virtualization provider, add it to the appliance and authenticate its hosts. You can also configure capacity and utilization data collection to help track usage and find common issues.

Enabling Red Hat Virtualization Capacity and Utilization Data Collection

Configure the following to collect capacity and utilization data from a Red Hat Virtualization provider:

- In Infrastructure Automation, enable the capacity and utilization server roles from the settings menu, in **Configuration > Server > Server Control**. For more information on capacity and utilization collection, see Assigning the Capacity and Utilization Server Roles in the *Deployment Planning Guide*.
- For information on selecting clusters and datastores used to collect data, see Capacity and Utilization Data Collected in the *General Configuration Guide*.
- In your Red Hat Virtualization environment, install the Data Warehouse and Reports components, and create a Infrastructure Automation user in the Data Warehouse database:
  - To install the Data Warehouse and Reports components in a Red Hat Virtualization environment, see the Data Warehouse Guide.
  - To create a Infrastructure Automation user in the Data Warehouse database, see Data Collection for Red Hat Enterprise Virtualization in the *Deployment Planning Guide*.

Adding a Red Hat Virtualization Provider

After initial installation and creation of a Infrastructure Automation environment, add a Red Hat Virtualization provider to the appliance.

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click **Configuration**, then click **Add a New Infrastructure Provider**.
3. Select **Red Hat Virtualization** from the **Type** list.
4. Enter a **Name** for the provider.
5. Select the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **default**.
6. Under **Endpoints** in the **Default** tab, configure the following:
   - Enter the **Hostname** or IPv4 or IPv6 address of the Red Hat Virtualization Manager.

     Important:

     The **Hostname** must be a unique fully qualified domain name.
   - Select **Verify** or **Do not verify** for **SSL Verification** to specify whether to authenticate securely to the provider using SSL.
     - If you select **Verify** for **SSL Verification**, you can either paste a custom certificate in the **Trusted CA Certificates** field in PEM format, or leave the **Trusted CA Certificates** field empty if your Red Hat Virtualization provider has a trusted Certificate Authority.
   - Enter the **API Port** if your provider uses a non-standard port for access.
   - Provide the login credentials for the Red Hat Virtualization administrative user:
     - Enter the user name (formatted as `admin@internal`) in the **Username** field.
     - Enter the password in the **Password** field.
     - Click **Validate** to confirm Infrastructure Automation can connect to the Red Hat Virtualization Manager.
7. Under **Endpoints** in the **Metrics** tab, you can configure capacity and utilization metrics collection by providing login credentials for the Infrastructure Automation user of the Red Hat Virtualization Data Warehouse database. You can also configure this later by editing the provider. Configure the following in the **Metrics** tab:

   Important

   To collect capacity and utilization data from a Red Hat Virtualization provider, the capacity and utilization server roles must be enabled in Infrastructure Automation. The Red Hat Virtualization environment must also contain the Data Warehouse and Reports components and a Infrastructure Automation user. Specific clusters, hosts, and datastores can also be configured for collection. See Enabling Red Hat Virtualization Capacity and Utilization Data Collection for configuration details.
   - Enter the database hostname or IPv4 or IPv6 address in **Hostname**.
   - Enter the **API Port** if your provider uses a non-standard port for access.
   - Enter the database user name in the **Username** field.
   - Enter the user password in the **Password** field.
   - Enter the **Database Name**.
   - Click **Validate** to confirm Infrastructure Automation can connect to the database.

8. Click **Add** to finish adding the Red Hat Virtualization provider.

Authenticating Red Hat Virtualization Hosts

After adding a Red Hat Virtualization infrastructure provider, you must authenticate its hosts to enable full functionality.

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click on a provider to display its summary screen.
3. On the summary screen, click **Hosts** in the **Relationships** information box to display the hosts on that provider.
4. Select the hosts to authenticate. You can select all hosts using the **Check All** option.
5. Click **Configuration**.
6. Click (**Edit this item**).
7. In the **Credentials** area, enter credentials for the following, as required:
    1. **Default**: This field is mandatory. Users should have privileged access such as, root or administrator.
    2. **Remote Login**: Credentials for this field are required if SSH login is disabled for the **Default** account.
    3. **Web Services**: This tab is used for access to Web Services in Red Hat Virtualization.
    4. **IPMI**: This tab is used for access to IPMI.
8. Click **Validate**.
9. If editing multiple hosts:
    1. Select a host from the **Select Host to validate against** list.
    2. If required, enter credentials for **Remote Login**, **Web Services**, and **IPMI** in their respective tabs; click **Validate**.
    3. Select another host to validate each of these credentials against.
10. Click **Add**.

- OpenStack Infrastructure Providers
    - Adding an OpenStack Infrastructure Provider
    - Configuring the Undercloud to Store Events

OpenStack Infrastructure Providers

Enable an OpenStack Infrastructure provider by adding it to the appliance.

Adding an OpenStack Infrastructure Provider

After initial installation and creation of a Infrastructure Automation environment, add an OpenStack infrastructure provider to the appliance. Infrastructure Automation supports operating with the OpenStack **`admin`** tenant. When creating an OpenStack infrastructure provider in Infrastructure Automation, select the OpenStack infrastructure provider's **`admin`** user because it is the default administrator of the OpenStack **`admin`** tenant. When using the **`admin`** credentials, a user in Infrastructure Automation provisions into the **`admin`** tenant, and sees images, networks, and instances that are associated with the **`admin`** tenant.

Note:

- You can set whether Infrastructure Automation should use the Telemetry service or Advanced Message Queueing Protocol (AMQP) for event monitoring. If you choose Telemetry, you should first configure the **ceilometer** service on the undercloud to store events. See Configuring the Undercloud to Store Events for instructions. For more information, see OpenStack Telemetry (ceilometer) in the Red Hat OpenStack Platform *Architecture Guide*.
- To authenticate the provider using a self-signed Certificate Authority (CA), configure the Infrastructure Automation appliance to trust the certificate using the steps in Using a Self-Signed CA Certificate before adding the provider.

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click **Configuration**, then click Add a New Infrastructure Provider (**Add a New Infrastructure Provider**).
3. Enter the **Name** of the provider to add. The **Name** is how the device is labeled in the console.
4. Select **OpenStack Platform Director** from the **Type** list.
5. Select the **API Version** of your OpenStack provider's Keystone service from the list. The default is **`Keystone v2`**.

    Note:

    - With Keystone API v3, domains are used to determine administrative boundaries of service entities in OpenStack. Domains allow you to group users together for various purposes, such as setting domain-specific configuration or security options. For more information, see OpenStack Identity (keystone) in the Red Hat OpenStack Platform *Architecture Guide*.

- The provider you are creating will be able to see projects for the given domain only. To see projects for other domains, add it as another cloud provider. For more information on domain management in OpenStack, see Domain Management in the Red Hat OpenStack Platform *Users and Identity Management Guide*.

1. Select the appropriate **Zone** for the provider. By default, the zone is set to **default**.

   Note:

   For more information, see the definition of host aggregates and availability zones in OpenStack Compute (nova) in the Red Hat OpenStack Platform *Architecture Guide*.

2. In the **Default** tab, under **Endpoints**, configure the host and authentication details of your OpenStack provider:

   1. Select a **Security Protocol** method to specify how to authenticate the provider:

      - **SSL without validation**: Authenticate the provider insecurely using SSL.
      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option. This is the recommended authentication method.
      - **Non-SSL**: Connect to the provider insecurely using only HTTP protocol, without SSL.

   2. Enter the **Host Name or IP address(IPv4 or IPv6)** of the provider. If your provider is an undercloud, use its hostname (see Setting the Hostname for the System in Red Hat OpenStack Platform *Director Installation and Usage* for more details)
   3. In **API Port**, set the public port used by the OpenStack Keystone service. By default, OpenStack uses port 5000 for non-SSL security protocol. For SSL, API port is 13000 by default.
   4. In the **Username** field, enter the name of an OpenStack user with privileged access (for example, **admin**). Then, provide its corresponding password in the **Password** and **Confirm Password** fields.
   5. Click **Validate** to confirm Infrastructure Automation can connect to the OpenStack provider.

3. Next, configure how Infrastructure Automation should receive events from the OpenStack provider. Click the **Events** tab in the **Endpoints** section to start.

   - To use the Telemetry service of the OpenStack provider, select **Ceilometer**. Before you do so, the provider must first be configured accordingly. See Configuring the Undercloud to Store Events for details.
   - If you prefer to use the AMQP Messaging bus instead, select **AMQP**. When you do: In **Hostname (or IPv4 or IPv6 address)** (of the **Events** tab, under **Endpoints**), enter the public IP or fully qualified domain name of the AMQP host.
      - In the **API Port**, set the public port used by AMQP. By default, OpenStack uses port 5672 for this.
      - In the **Username** field, enter the name of an OpenStack user with privileged access (for example, **admin**). Then, provide its corresponding password in the **Password** field.
      - Click **Validate** to confirm the credentials.

4. You can also configure SSH access to all hosts managed by the OpenStack infrastructure provider. To do so, click on the **RSA key pair** tab in the **Endpoints** section.

   1. From there, enter the **Username** of an account with privileged access.
   2. If you selected **SSL** in **Endpoints > Default > Security Protocol** earlier, use the **Browse** button to find and set a private key.

5. Click **Add** after configuring the infrastructure provider.

Note:

Infrastructure Automation requires that the **`adminURL`** endpoint for all OpenStack services be on a non-private network. Accordingly, assign the adminURL endpoint an IP address of something other than **`192.168.x.x`**. The **`adminURL`** endpoint must be accessible to the Infrastructure Automation appliance that is responsible for collecting inventory and gathering metrics from the OpenStack environment. Additionally, all the Keystone endpoints must be accessible, otherwise refresh will fail.

Configuring the Undercloud to Store Events

To allow Infrastructure Automation to receive events from a Red Hat OpenStack Platform environment, you must configure the **notification_driver** option for the Compute service and Orchestration service in that environment. See Installing the Undercloud and Configuring the Director in Red Hat OpenStack Platform *Director Installation and Usage* for related details.

- VMware vCenter Providers
  - Adding a VMware vCenter Provider
  - Using a Non-Administrator Account for vCenter Hosts
  - Authenticating VMware vCenter Hosts

VMware vCenter Providers

To use a VMware vCenter provider, add it to the appliance and authenticate its hosts.

Adding a VMware vCenter Provider

After initial installation and creation of a Infrastructure Automation environment, add a VMware vCenter provider to the appliance.

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click **Configuration**, then click (**Add a New Infrastructure Provider**).
3. Enter the **Name** of the provider to add. The **Name** is how the device is labeled in the console.
4. Select **VMware vCenter** from the **Type** list.
5. Enter the **Host Name or IP address(IPv4 or IPv6)** of the provider.

   The **Host Name** must use a unique fully qualified domain name.
6. Select the appropriate **Zone** for the provider. By default, the zone is set to **default**.
7. Optional: Enter the **Host Default VNC Start Port** and **Host Default VNC End Port**. These properties are used for remote console access with VNC. If VNC is available from the hosts, list the available port range using the start and end port.
8. In the **Credentials** area, under **Default**, provide the login credentials required for the VMware vCenter administrative user:
   - Enter the user name in the **Username** field.
   - Enter the password in the **Password** field.
   - Confirm the password in the **Confirm Password** field.
   - Click **Validate** to confirm Infrastructure Automation can connect to the VMware vCenter.
9. Click **Add**.

Using a Non-Administrator Account for vCenter Hosts

After adding a VMware vCenter infrastructure provider, you must authenticate its hosts to enable full functionality. You can use administrator credentials, or create another user assigned to a role created for Infrastructure Automation. See the VMware documentation for instructions on how to create a role.

The following privileges should be enabled for the non-administrator user:

From the Global group, check:

- Cancel task
- Diagnostics
- Log Event
- Set custom attribute
- Settings

Check the entire set of privileges for the following groups:

- Alarms
- Datastores
- dvPort Group
- Host
- Network
- Resource
- Scheduled Task
- Tasks
- Virtual Machine
- vSphere Distributed Switch

Additionally, you must assign the new role to the following objects:

- **Datacenter**: At the Datacenter the Infrastructure Automation user/group must have at least the read-only role at the Datacenter level (Not Propagated) to be able to see the datacenter. Without this access, relationships cannot be made. Specifically, the datastores will not show up.
- **Cluster**: Each Cluster that the Infrastructure Automation needs access to must have the new role assigned and propagated.
- **Folders**: Each Folder that Infrastructure Automation needs access to must have the new role assigned and propagated.
- **Datastores**: Each Datastore that Infrastructure Automation needs access to must have the new role assigned and propagated.
- **Networking**: Each vLAN or Port Group that Infrastructure Automation needs access to must have the new role assigned and propagated.

Authenticating VMware vCenter Hosts

The procedure below describes how to authenticate the VMware vCenter hosts.

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click on a provider to display its summary screen.
3. On the summary screen, click **Hosts** in the **Relationships** information box to display the hosts on that provider.
4. Select the hosts to authenticate. You can select all hosts using the **Check All** option.
5. Click **Configuration**
6. Click  (**Edit Selected items**).
7. In the **Credentials** area, under **Default**, provide the VMware ESXi login credentials:
    - Enter the user name in the **Username** field.
    - Enter the password in the **Password** field.
    - Confirm the password in the **Confirm Password** field.
    - Click **Validate** to confirm Infrastructure Automation can connect to the VMware vCenter host.
8. If editing multiple hosts, select a host from the **Select Host to validate against** list; provide the VMware ESXi login credentials and click **Validate**.
9. Click **Save**.

- IBM Power HMC Providers
    - Adding an IBM Power HMC Provider

IBM Power HMC Providers

To use an IBM Power HMC provider, add it to the appliance and authenticate its hosts.

Adding an IBM Power HMC Provider

After initial installation and creation of a Infrastructure Automation environment, add an IBM Power HMC provider to the appliance.

1. Browse to menu: **Compute > Infrastructure > Providers**.
2. Click **Configuration**, then click  (**Add a New Infrastructure Provider**).
3. Select **IBM Power HMC** from the **Type** list.
4. Enter the **Name** of the provider to add. The **Name** is how the device is labeled in the console.
5. Select the appropriate **Zone** for the provider. By default, the zone is set to **default**.
6. Under **Endpoints**, configure the following:
    - Select **SSL** or **SSL without validation** for **Security Protocol** to specify whether to authenticate securely to the provider using SSL.
    - Enter the **Hostname (or IPv4 or IPv6 address)** of the provider. The **Hostname** must use a unique fully qualified domain name.
    - Enter the **API port** if your provider uses a non-standard port for access.
    - Enter the user name in the **Username** field.
    - Enter the password in the **Password** field.
    - Click **Validate** to confirm Infrastructure Automation can connect to the IBM Power HMC.
7. Click **Add**.

- OpenShift Virtualization (KubeVirt) Providers

OpenShift Virtualization (KubeVirt) Providers

OpenShift Virtualization (KubeVirt) is a virtualization layer which allows Virtual Machines to be managed along side containers using standard Kubernetes utilities.

In order to add a OpenShift Virtualization (KubeVirt) provider to ManageIQ you must first have a Kubernetes or OpenShift provider and then add a Virtualization endpoint.

Once added you should see VMs, Hosts, Clusters, and Storages like any other Infrastructure provider.

- IBM Terraform provider
  - Adding an IBM Terraform Provider
    - In the Endpoint section:
  - Triggering a Refresh of an IBM Terraform Provider

IBM Terraform provider

Infrastructure Automation includes an IBM Terraform provider that provides functionality to integrate with the Managed Services component. Managed services uses open source Terraform to manage and deliver cloud infrastructure as code. Terraform is an Open Source software that is developed by HashiCorp that enables predictable and consistent provisioning of many cloud platforms, classic infrastructure, and virtual private cloud (VPC) infrastructure resources by using a high-level scripting language. You can use Terraform to automate your cloud resource provisioning, rapidly build complex, multi-tier cloud environments, and enable Infrastructure as Code (IaC). Infrastructure Automation provides functionality to integrate with Terraform and take advantage of features. This includes:

- Discovering the inventory from Managed Services, including templates, stacks, and virtual machines provisioned using Terraform.
- Providing traceability between VM resources provisioned by Terraform and discovered by Infrastructure Automation.
- Applying Infrastructure Automation policy tags to hosts.

Adding an IBM Terraform Provider

To start discovering the Terraform templates and virtual machines, you need at least one IBM Terraform provider added to Infrastructure Automation.

1. Browse to menu: **Configuration > Providers**.
2. Select menu: **Configuration > Add a new Provider**.
3. Select **Type** as **IBM Terraform Configuration**
4. Enter a **Name** for the provider.
5. Select a **Zone**, choose **default**.

In the Endpoint section:

6. Enter the **Managed Services URL** for the provider.

   This is the URL for Managed Services and can be either an IP address or a hostname.

   For example, `https://cam.apps.test.example.com`.

7. In the `Endpoint` section, enter the **CloudPak for MCM URL**. This is the console URL for the IBM CloudPak for Multicloud Management system. This is used for authentication and access token.

   For example, `https://cp-console.apps.test.example.com`.

8. Select **SSL verification** as **Verify** or **Do not verify**.
9. Enter a **Username** for a user on the provider. This must be a user in IBM CloudPak for Multicloud Management.
10. Enter a **Password**.
11. Click **Validate** to test your connection.
12. Click **Add** to confirm your settings and save the provider.

Infrastructure Automation saves the IBM Terraform provider in its database and triggers a refresh of resources detected in the provider.

Triggering a Refresh of an IBM Terraform Provider

Your IBM Terraform provider can still create new hosts independently of Infrastructure Automation. Your Infrastructure Automation detects these changes after an automatic refresh period. However, you can trigger a manual refresh to avoid waiting for the automatic refresh.

1. Navigate to menu: **Configuration > Providers**.
2. Select your Terraform provider using the checkbox, and click menu: **Configuration > Refresh Relationships and Power States**. This triggers the refresh.
3. When the refresh is complete, select the Terraform provider to check the updated list of Configuration profiles and Configured systems in the provider.

- Red Hat Satellite 6
  - Defining the Workflow
  - Defining the Hostgroup Hierarchy
  - Adding a Satellite 6 Provider
  - Triggering a Refresh of a Satellite 6 Provider
  - Displaying Red Hat Satellite 6 Contents
  - Reprovisioning a Bare Metal Host

- Tagging a Bare Metal Host

Red Hat Satellite 6

Satellite 6 is a subscription and system management tool that provides a way to provision hosts (both virtual and bare metal) and configure them using a set of Puppet modules. Infrastructure Automation provides functionality to integrate with a Red Hat Satellite 6 server and take advantage of its features. This includes:

- Monitoring the inventory of your Red Hat Satellite 6 server, including independent hosts and hosts provisioned using hostgroups.
- Reprovisioning existing bare metal system hosts to new host groups.
- Applying Infrastructure Automation policy tags to hosts.

Infrastructure Automation only reprovisions existing systems in a Red Hat Satellite 6 environment. Provisioning systems from Red Hat Satellite 6's bare metal discovery service is planned for a future release.

Defining the Workflow

This section uses the following workflow:

1. Add Red Hat Satellite 6 server details to Infrastructure Automation.
2. Refresh the state of your Red Hat Satellite 6 provider in Infrastructure Automation.
3. Select an existing bare metal host from Red Hat Satellite 6 for reprovisioning.
4. Apply policy tags to Red Hat Satellite 6 hosts.

Defining the Hostgroup Hierarchy

Infrastructure Automation displays the Red Hat Satellite 6 infrastructure in a host group and host relationship. A host group defines a set of default values that hosts inherit when placed in that group. Hosts can belong to only one host group, but host groups can be nested in hierarchies. You can create a **"base"** or **"parent"** host group that represents all hosts in your organization, and then create nested or **"child"** host groups under that parent to provide specific settings.

Adding a Satellite 6 Provider

To start provisioning bare metal machines, you need at least one Red Hat Satelllite 6 provider added to Infrastructure Automation.

1. Browse to menu: **Configuration > Management**.
2. Select menu: **Configuration > Add a new Provider**.
3. Enter a **Name** for the provider.
4. Enter a **URL** for the provider. This is the root URL for the Satellite 6 server and can be either an IP address or a hostname. For example, `http://satellite6.example.com`.
5. Select **Verify Peer Certificate** to use encrypted communication with the provider. This requires the **SSL certificates** from your Red Hat Satellite 6 provider.
6. Enter a **Username** for a user on the provider. Ideally, this would be a user in Satellite 6 with administrative access.
7. Enter a **Password**, and then enter it again in **Confirm Password**.
8. Click **Validate** to test your connection with the Red Hat Satellite 6 server.
9. Click **Add** to confirm your settings and save the provider.

Infrastructure Automation saves the Satellite 6 provider in its database and triggers a refresh of resources detected in the provider.

Triggering a Refresh of a Satellite 6 Provider

Your Satellite 6 provider can still create new hosts independently of Infrastructure Automation. Your Infrastructure Automation appliance detects these changes after an automatic refresh period. However, you can trigger a manual refresh to avoid waiting for the automatic refresh.

1. Navigate to **Configuration > Management**.
2. Select your Red Hat Satellite 6 provider using the checkbox, and click **Configuration > Refresh Relationships and Power States**. This triggers the refresh.
3. When the refresh is complete, select the Red Hat Satellite 6 provider to check the updated list of hosts groups in the provider.

Displaying Red Hat Satellite 6 Contents

Infrastructure Automation provides two methods for viewing the contents of a Red Hat Satellite 6 provider:

- **Providers** - This presents the Red Hat Satellite 6 contents as a hierarchy of host groups belonging to a provider, and then individual hosts belonging to each provider.

- **Configured Systems** - This presents a list of all hosts on your Red Hat Satellite 6 server. This also provides a method to apply predefined filters to organized specific machines.

Change between these two views using the accordion menu on the left of the user interface.

Reprovisioning a Bare Metal Host

This procedure provides an example of reprovisioning an existing bare metal system into a new hostgroup. For this example, your Red Hat Satellite 6 environment requires the following:

- An existing bare metal system stored as a host object in your Red Hat Satellite 6 server. This system can be one of the following:
  - A standalone system previously provisioned without a host group.
  - A system previously provisioned using a host group.
- A target host group. This host group contains the system configuration to apply to the host when reprovisioning it. This includes:
  - A new operating system installation, including a new partition table.
  - A new networking configuration that the Red Hat Satellite 6 server defines and manages.
  - Registration to any Red Hat subscriptions and repositories assigned to the host group.
  - Application of any Puppet modules assigned to the host group.

1. Browse to menu: **Configuration > Management**.
2. Select **Configured Systems** from the accordion menu on the left. This displays the system list.
3. Select one or more hosts to reprovision.
4. Select **Lifecycle > Provision Configured Systems**.
5. Under the **Request** tab, enter the following details:
   1. **E-Mail address**
   2. **First Name**
   3. **Last Name**
   4. This form also contains optional fields for users to enter a plain text **Note** to inform Infrastructure Automation administrators of any special details, and a field to provide a manager's name in case administrators require approval from a user's manager.
6. Select the **Purpose** tab and select any Infrastructure Automation policy tags that apply to the system.
7. Select the **Catalog** tab. This screen displays the list of chosen machines to reprovision and their current details. Select a **target host group** from the **Configuration Profile** list. Infrastructure Automation communicates with Red Hat Satellite to apply the configuration from this host group to the selected host and reprovision the system.
8. Select the **Customize** tab. This screen displays some customizable fields for the selected system. You can change the **Root Password** or change the **Hostname** and **IP Address**. Note that these fields are optional, because the host group in Red Hat Satellite 6 contains this information. The fields here will override the settings from the host group.

   Provisioning bare metal systems still requires access to the network that Red Hat Satellite 6 manages. This is because Red Hat Satellite controls PXE booting, kickstarts, and Puppet configuration for bare metal systems. Ensure the IP address you enter in Infrastructure Automation can access a DHCP service that Red Hat Satellite 6 provides either through the main server or through a Red Hat Satellite 6 Capsule server.
9. Select the **Customize** tab. This screen allows you to either launch the provisioning process immediately on approval or using a schedule. Click **Schedule** to show the date and time fields used to schedule the provisioning.
10. Click **Submit**.

Depending on the request settings on your Infrastructure Automation appliance, this provisioning request might require approval from an administrator. If not, the provisioning request launches depending on your choice for the schedule.

Note:

Previously provisioned hosts might require manual selection of PXE boot from the boot menu, otherwise they might boot to hard disk and not reprovision.

Tagging a Bare Metal Host

Infrastructure Automation can also control policy settings of bare metal systems from Red Hat Satellite 6 through tagging. Tagging attaches levels of metadata to help define the policy rules required for a set of systems.

1. Navigate to **Configuration > Management**.
2. Select **Configured Systems** from the accordion menu on the left. This displays the system list.
3. Select one or more hosts to tag.

4. Select **Policy > Edit Tags**.
5. Under **Tag Assignment**, select a tag from **Select a customer tag to assign** and then choose a value from **Select a value to assign**. For example, you can tag this system as located in Chicago by selecting `Location` as the tag and `Chicago` as the value. Once selected, the user interface automatically adds this tag and value to the table below.
6. Click **Save**.

The bare metal system is now configured with a set of policy tags.

- Azure Providers
  - Adding Azure Providers
  - Adding Azure Cloud Regions
  - Disabling Azure Cloud Regions

Azure Providers

Adding Azure Providers

Infrastructure Automation supports Microsoft Azure providers. Before Infrastructure Automation can be authenticated to Microsoft Azure, you must complete a series of prerequisite steps using the Azure portal. For more information, see Create Active Directory application and service principal account using the Azure portal.

Follow the steps to set up an Azure Active Directory (Azure AD) and assign the required permissions to it, then create an Azure Active Directory application, and obtain the **Application ID** (Client ID), **Directory ID** (Tenant ID), **Subscription ID**, and **Key Value** (Client Key) that are required to add and connect to the Azure instance as a provider in Infrastructure Automation. Currently, all of these steps can be performed using either the Azure Resource Manager or Service Manager (Classic) mode.

Note:

In the steps described in Create Active Directory application and service principal account using the Azure portal:

- The **Application ID** obtained during *Get Application ID and Authentication Key* is your **Client ID**. In the same section, after providing a description and a duration for the key, the **VALUE** displayed after clicking **Save** is your **Client Key**. If you choose an expiring key, make sure to note the expiration date, as you need to generate a new key before that day in order to avoid an interruption.
- The **Directory ID** obtained during *Get Tenant ID* is your **Tenant ID**. In Azure Active Directory (Azure AD), a tenant is a dedicated instance of the Azure AD service and is representative of an organization. It houses the users in a company and the information about them - their user profile data, permissions, groups, applications, and other information related to an organization and its security. To allow Azure AD users to sign in to your application, you must register your application in a tenant of your own which is assigned a Tenant ID (Directory ID).
- During *Assign Application to Role*, select the **Contributor** role and not the **Reader** role.
- To obtain your **Subscription ID**, log in to the Azure portal and click **Subscriptions** on the slide-out menu. Find the appropriate subscription and see your Azure **Subscription ID** associated with it. Note that if the **Subscriptions** tab is not visible, then click on **More services** to find it. The Azure **Subscription ID** is like a billing unit for all of the services consumed in your Azure account, including virtual machines and storage. The **Subscription ID** is in the form of a Globally Unique Identifier (GUID).

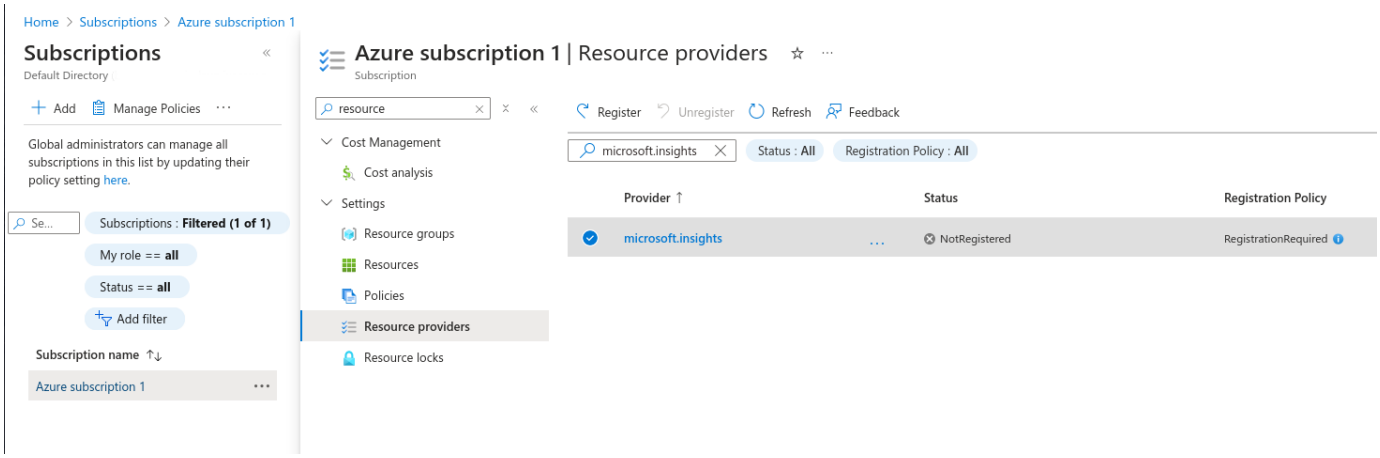
When the service principal account (instance of an application in a directory) is created using the Azure portal, the following four pieces of information becomes available within the Azure AD module.

- Directory ID (Tenant ID)
- Subscription ID
- Application ID (Client ID)
- Client Key

You can now use these values in the procedure below to add an Azure cloud instance as a provider to Infrastructure Automation.

To capture Events or Capacity and Utilization metrics from your Azure account you need to register the Microsoft Insights resource provider.

1. Select your subscription, and click **Resource providers**.
2. Enter **microsoft.insights** in the search box.
3. If the provider isn't registered, then select the checkbox, and click **Register**.



Add an Azure Cloud Provider:

1. Browse to the menu and click **Compute** > **Clouds** > **Providers**.
2. Click **Configuration**, then click ⊕ (**Add a New Cloud Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **Azure**.
5. Select a region from the **Region** list. One provider will be created for the selected region.
6. Enter **Tenant ID**.
7. Enter **Subscription ID**.
8. Enter **Zone**.
9. In the **Credentials** section, enter the **Client ID** and **Client Key**; click **Validate**.
10. Click **Add**.

Adding Azure Cloud Regions

Infrastructure Automation allows administrators to add additional Azure cloud regions on the appliance server. You can use this capability to add new regions that have been set up since Infrastructure Automation was released. Once added the region it will be available when creating a new Azure provider.

1. Click ⚙ **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the additional region to that server only.
3. Click on the **Advanced** tab.
4. Search for `:ems_azure:`, and enter the regions you want to add under `:additional_regions:`.

   Example. To add hypothetical `lunar-base-1` `lunar-base-2` regions:

```
:ems_azure:
  :additional_regions:
    :lunar-base-1:
      :name: lunar-base-1
      :description: Lunar Base 1
    :lunar-base-2:
      :name: lunar-base-2
      :description: Lunar Base 2
```

5. Click **Save**.

Disabling Azure Cloud Regions

Infrastructure Automation allows administrators to disable Azure cloud regions on the appliance server. You can use this capability to disable certain classified regions. Once disabled, the region will not be available when adding a new Azure provider.

1. Click ⚙ **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the disabled region to that server only.
3. Click the zone where the Infrastructure Automation server is located, then click on the EVM server.
4. Click on the **Advanced** tab.

5. Search for `:ems_azure:`, and enter the regions you want to disable under `:disabled_regions:`.

   Example. To disable the `us-gov-arizona` and `us-gov-texas` regions:

   ```
   :ems_azure:
     :disabled_regions:
     - us-gov-arizona
     - us-gov-texas
   ```

6. Click **Save**.

- Amazon EC2 Providers
  - Permissions for Amazon EC2 Providers
  - Manually Creating an Amazon EC2 Role
- Adding Amazon EC2 Providers
  - Enabling Public AMIs from Amazon EC2
  - Enabling AWS Config Notifications
  - Enabling Amazon EC2 Events
  - Creating a CloudTrail
  - Creating CloudWatch Rules Based on Event Patterns
  - Adding Amazon Cloud Regions
  - Disabling Amazon Cloud Regions

Amazon EC2 Providers

Permissions for Amazon EC2 Providers

Further limiting API access limitations can limit Automate capabilities, as Automate scripts directly access the AWS SDK to create brand new application functionality.

The AWS services primarily accessed by the Infrastructure Automation API include:

- Elastic Compute Cloud (EC2)
- CloudFormation
- CloudWatch
- Elastic Load Balancing
- Simple Notification Service (SNS)
- Simple Queue Service (SQS)

Manually Creating an Amazon EC2 Role

To eliminate the need to assign *Admin* group privileges to the Amazon EC2 provider, create an IAM role following the procedure described in Creating a Role for an AWS Service (Console) in the Amazon Web Services documentation.

Use the following parameters:

1. Select **EC2** as the service the role will use.
2. Attach the following permissions:
   1. **`AmazonEC2FullAccess`**
   2. **`AmazonS3FullAccess`**
   3. **`AmazonSQSFullAccess`**
3. Enter **`smartstate`** for the **Role name**.

Once the IAM role is created, assign the provider *Power User* privileges as described in Permissions for Amazon EC2 Providers.

Adding Amazon EC2 Providers

Complete the following procedure to add an Amazon EC2 cloud provider in Infrastructure Automation.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click **Configuration**, then click ⊕ (**Add a New Cloud Provider**).
3. Enter a **Name** for the provider.

4. From the **Type** list, select **Amazon EC2**.
5. Select a **Region**.
6. Select the appropriate **Zone** if you have more than one available.
7. Under **Endpoints**, click the **Default** tab.
    1. Enter the **Endpoint URL**.

       Note:

       AWS allows users to set a custom endpoint URL when connecting to certain services, which you can add in the Infrastructure Automation user interface per Amazon EC2 provider. See Interface VPC Endpoints (AWS PrivateLink) for more information.

    2. Generate an **Access Key** in the **Security Credentials** of your Amazon AWS account. The **Access Key ID** acts as your **User ID**, and your **Secret Access Key** acts as your **Password**.
    3. Click **Validate** to validate the credentials.
8. Click the **SmartState Docker** tab.
    1. Enter the **SmartState Docker User Name** and **SmartState Docker Password**.
9. Click **Add**.

Enabling Public AMIs from Amazon EC2

By default, public AMIs from an Amazon EC2 provider are not viewable in Infrastructure Automation. To make these images viewable, you must edit the main configuration file for the appliance.

Note:

Syncing all public images may require additional memory resources. Also, bear in mind that syncing happens in each configured Amazon EC2 provider, which will require a similar amount of total memory resources.

1. Browse to the settings menu, then menu: **Configuration > Zone > Advanced**.
2. Select the configuration file to edit from the **File** list. If not already automatically selected, select **EVM Server Main Configuration**.
3. Set the **get_public_images** parameter:
    1. Set the parameter to **get_public_images: true** to make public images viewable.
    2. Set the parameter to **get_public_images: false** to make public images not viewable.
4. Optionally, configure an array of filters in **public_images_filters** to restrict which images are synced. See http://docs.aws.amazon.com/sdkforruby/api/Aws/EC2/Client.html#describe_images-instance_method for more details.

Enabling AWS Config Notifications

Amazon's AWS Config notifies subscribers of changes in a region through its Simple Notification Service (SNS). Infrastructure Automation subscribes to the SNS service for AWS Config deltas and converts the deltas into Infrastructure Automation events.

1. Enable the AWS Config service in the AWS Management Console. See the AWS Config Developer Guide for more information.
2. Create a new Amazon SNS topic named **`AWSConfig_topic`**. Infrastructure Automation automatically connects to this topic.
3. (Optional) Configure the frequency of delta creation in the AWS Management Console.

You can assign Infrastructure Automation policies to the AWS events listed below. The appliance performs a provider refresh on all these events except for **`AWS_EC2_Instance_UPDATE`**.

Event	Policies	Refresh
AWS_EC2_Instance_CREATE	src_vm vm_create	ems
AWS_EC2_Instance_UPDATE	N/A	ems
AWS_EC2_Instance_running	src_vm vm_start	ems
AWS_EC2_Instance_stopped	src_vm vm_power_off	ems
AWS_EC2_Instance_shutting-down	src_vm vm_power_off	ems

Enabling Amazon EC2 Events

After adding an Amazon EC2 provider and configuring an SNS topic in Enabling AWS Config Notifications, create a CloudTrail, then configure CloudWatch rules on your EC2 provider to automatically get events in Infrastructure Automation for monitoring the provider.

Note:

The following procedures are accurate at time of publishing. See the Amazon AWS documentation for further details on these steps.

Creating a CloudTrail

In the CloudTrail area of the AWS Management Console, create a trail and an S3 bucket:

1. Create a **Trail** with a custom name.
2. (Optional) If you want to apply the trail to all of your Infrastructure Automation regions, select **Yes** for **Apply trail to all regions**.
3. For **Management Events**, select **Read/Write events: All**.
4. Create a new S3 bucket.

Creating CloudWatch Rules Based on Event Patterns

In the CloudWatch area of the AWS Management Console, create three rules: one rule each for EC2, volumes, and snapshots.

When an SNS topic is deleted and recreated (manually or by Infrastructure Automation), CloudWatch rules must be recreated as well, even though the SNS target topic for CloudWatch rules appears to be assigned to these rules. The CloudWatch rule does not send events to this recreated topic until it is recreated too.
To create a CloudWatch rule for EC2:

1. Browse to menu: **Events > Rules** and click **Create rule**.
2. Select the **Event Pattern** radio button to specify the event source.
3. Edit the **Event Pattern Preview** box, and paste and save the following code to create a rule based on a custom event pattern:

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "ec2.amazonaws.com"
    ]
  }
}
```

4. Click **Add target** and specify the following attributes:
   - **Type**: `SNS Topic`
   - **Topic**: `AWSConfig_topic`
   - **Input**: `Matched event`
5. Click **Configure Details** to save these details.
6. Configure a name and description for the rule if desired. Ensure the **Enabled** checkbox is selected for **State**.
7. Click **Create rule** to save the CloudWatch rule.

Repeat the same procedure to create a CloudWatch rule for EC2 Instance State-changes:

1. Browse to menu: **Events > Rules** and click **Create rule**.
2. Select the **Event Pattern** radio button to specify the event source.
3. Edit the **Event Pattern Preview** box, and paste and save the following code to create a rule based on a custom event pattern:

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EC2 Instance State-change Notification"
  ]
}
```

4. Click **Add target** and specify the following attributes:
   - **Type**: `SNS Topic`
   - **Topic**: `AWSConfig_topic`
   - **Input**: `Matched event`

5. Click **Configure Details** to save these details.
6. Configure a name and description for the rule if desired. Ensure the **Enabled** checkbox is selected for **State**.
7. Click **Create rule** to save the CloudWatch rule.

Repeat the same procedure to create a CloudWatch rule for volumes, pasting the code snippet below to the **Event Pattern Preview** box:

1. Browse to menu: **Events > Rules** and click **Create rule**.
2. Select the **Event Pattern** radio button to specify the event source.
3. Edit the **Event Pattern Preview** box, and paste and save the following code to create a rule based on a custom event pattern:

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
}
```

4. Click **Add target** and specify the following attributes:
   - **Type**: `SNS Topic`
   - **Topic**: `AWSConfig_topic`
   - **Input**: `Matched event`
5. Click **Configure Details** to save these details.
6. Configure a name and description for the rule if desired. Ensure the **Enabled** checkbox is selected for **State**.
7. Click **Create rule** to save the CloudWatch rule.

Repeat the same procedure to create a CloudWatch rule for snapshots, pasting the code snippet below to the **Event Pattern Preview** box:

1. Browse to menu: **Events > Rules** and click **Create rule**.
2. Select the **Event Pattern** radio button to specify the event source.
3. Edit the **Event Pattern Preview** box, and paste and save the following code to create a rule based on a custom event pattern:

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Snapshot Notification"
  ]
}
```

4. Click **Add target** and specify the following attributes:
   - **Type**: `SNS Topic`
   - **Topic**: `AWSConfig_topic`
   - **Input**: `Matched event`
5. Click **Configure Details** to save these details.
6. Configure a name and description for the rule if desired. Ensure the **Enabled** checkbox is selected for **State**.
7. Click **Create rule** to save the CloudWatch rule.

EC2 can now automatically refresh events in Infrastructure Automation.

Adding Amazon Cloud Regions

Infrastructure Automation allows administrators to add Amazon cloud regions on the appliance server. You can use this capability to add new regions that have been set up since Infrastructure Automation was released. Once adding the region it will be available when creating a new Amazon EC2 provider.

1. Click **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the additional region to that server only.
3. Click on the **Advanced** tab.
4. Search for `:ems_amazon:`, and enter the regions you want to add under `:additional_regions:`.

```
Example. To add hypothetical `lunar-base-1` `lunar-base-2` regions:
```

```
:ems_amazon:
  :additional_regions:
    :lunar-base-1:
      :name: Lunar Base 1
      :hostname: ec2.lunar-base-1.amazonaws.com
      :description: Lunar Base 1
    :lunar-base-2:
      :name: Lunar Base 2
      :hostname: ec2.lunar-base-2.amazonaws.com
      :description: Lunar Base 2
```

5. Click **Save**.

Disabling Amazon Cloud Regions

Infrastructure Automation allows administrators to disable Amazon cloud regions on the appliance server. Use this capability to disable certain classified regions like AWS GovCloud. Once disabled, the region will not be available when adding an Amazon EC2 provider.

1. Click **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the additional region to that server only.
3. Click on the **Advanced** tab.
4. Search for `:ems_amazon:`, and enter the regions you want to disable under `:disabled_regions:`.

   Example. To disable the `ap-northeast-1` region:

   ```
   :ems_amazon:
     :disabled_regions:
     - us-gov-west-1
     - ap-northeast-1
   ```

5. Click **Save**.

- Google Compute Engine Providers
- Adding Google Compute Engine Providers
- Enabling Google Compute Engine Events
  - Configuring Google Compute Engine to Export Events
  - Viewing Google Compute Engine Events in Infrastructure Automation

Google Compute Engine Providers

Adding Google Compute Engine Providers

After initial installation and creation of a Infrastructure Automation environment, add a Google Compute Engine provider by following this procedure.

Prerequisites.

To add a Google Compute Engine provider to Infrastructure Automation, you need:

- A Google Cloud Platform account
- A Google Compute Engine project with the Google Compute Engine API enabled
- A service account JSON key for your project

  Note:

  You can generate a private JSON key for your project in **IAM & Admin > Service Accounts** in Google Cloud Platform. This key is used to authenticate against your provider.

  For additional information, see the Google Cloud Platform documentation.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click **Configuration**, then click (**Add a New Cloud Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **Google Compute Engine**.
5. Select your **Preferred Region** from the list.
6. Enter your Google Compute Engine Project ID for **Project**.
7. Select the appropriate **Zone** if you have more than one available. Red Hat recommends creating a new zone for your Google Compute Engine provider.

8. Copy your project's **Service Account** JSON key contents to the **Service Account JSON** field.
9. Click **Validate** to validate the credentials.
10. Click **Add**.

Note:

Make sure that NTP synchronization is enabled and working. When clocks are not synchronized, the following error will be raised:

```
Credential validation was not successful: Authorization failed. Server message: { "error" : "invalid_grant",
"error_description" : "Invalid JWT: Token must be a short-lived token and in a reasonable timeframe" }
```

Enabling Google Compute Engine Events

After adding Google Compute Engine as a provider in Infrastructure Automation, enable events for the provider so that you can monitor the system from Infrastructure Automation.

Events are set up on a per-project basis by using Google Stackdriver logging combined with Google Pub/Sub. Stackdriver logging is a service that aggregates and exposes log events from Google services and applications. Stackdriver exports the log events to Google Pub/Sub, a messaging service. This section describes how to export activity log entries for a Google Compute Engine project so that events are captured in Infrastructure Automation.

- You must have owner permission on the project you are exporting.
- The Google Cloud Pub/Sub API must be enabled for your project. To enable the API:

1. In Google Cloud Platform, select your project from the top menu bar.
2. Click  to show the **Products and Services** menu. Click **API Manager** to go to https://console.cloud.google.com/apis/library/.
3. In the API Manager **Overview** tab, search for `Pub/Sub` in the **Google APIs** search bar and select **Google Cloud Pub/Sub API** from the results. Click the **Enable** button.
4. If Google Cloud Pub/Sub API is already enabled, the **Enable** button will not show, and instead **Google Cloud Pub/Sub API** will be listed under **Enabled APIs**.

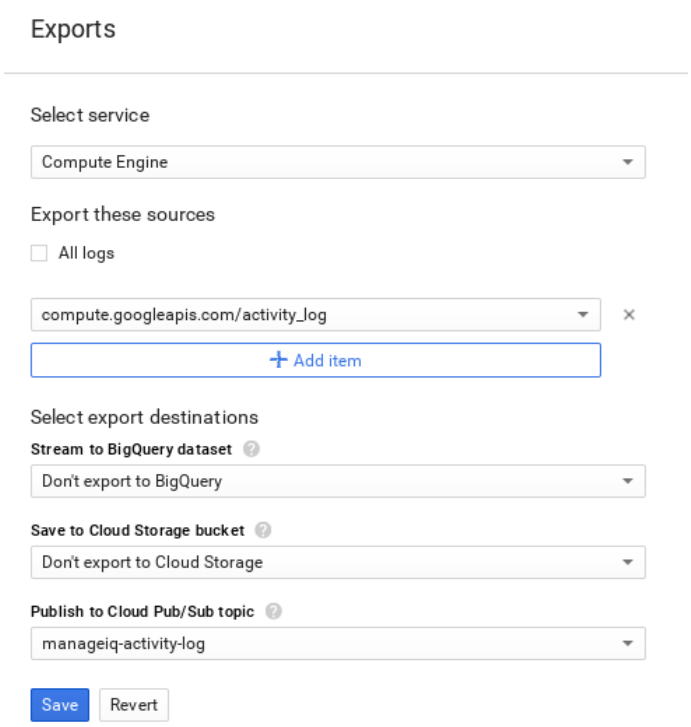
- The Stackdriver logging service must have permission to publish to your project's Pub/Sub service. To add the required permissions:

1. In Google Cloud Platform, select your project and browse to  **Products and Services > IAM & Admin > IAM** to go to https://console.cloud.google.com/iam-admin/iam/.
2. Assign **Logs Configuration Writer** permissions to your project:
    1. If the `cloud-logs@system.gserviceaccount.com` account is already listed under **Members**, ensure **Logs Configuration Writer** is selected under **Role(s)**.
    2. If the `cloud-logs@system.gserviceaccount.com` account is not listed under **Members**:
        1. Click **Add** to add the permissions.
        2. In the dialog box, enter `cloud-logs@system.gserviceaccount.com` in **Members** to add the Google APIs service account to the permissions list.
        3. In the **Select a Role** dropdown, select **Logging > Logs Configuration Writer** and click **Add**.

Configuring Google Compute Engine to Export Events

After you have completed the steps from itemizedlist_title, set up your Google Compute Engine project to export events to Infrastructure Automation with the following steps:

1. In Google Cloud Platform, click  to show the **Products and Services** menu, and click **Logging** to go to https://console.cloud.google.com/logs/.
2. Select your project from the top menu bar.
3. Click **Exports** from the **Logging** menu.
4. In the **Select service** list, select **Compute Engine**.
5. Under **Export these sources**, click **Add item**, and select `compute.googleapis.com/activity_log` from the list.
6. Under **Select export destinations**, click the **Publish to Cloud Pub/Sub topic** dropdown and click **Add new topic...**
7. In the **Create Cloud Pub/Sub Topic** dialog, enter `manageiq-activity-log` as the **Name**. Click **Create**.



8. Click **Save**.

When changes occur to Google Compute Engine instances, Infrastructure Automation is now notified and reports these changes as events.

Note:

For additional information about Google Compute Engine, see the Google Cloud Platform documentation:

- For information on setting up a cloud logging export on Google Cloud Platform, see https://cloud.google.com/logging/docs/export/configure_export.
- For information on Google Cloud Pub/Sub API operations and costs, see https://cloud.google.com/pubsub/.

Viewing Google Compute Engine Events in Infrastructure Automation

In Infrastructure Automation, view events for your Google Compute Engine project by following these steps:

1. Browse to menu: **Compute > Clouds > Providers** and select your Google Compute Engine project.
2. Click menu: **Monitoring > Timelines** on the provider summary page to see an events timeline for the project.

- IBM Cloud Infrastructure Center (CIC) Providers
  - IBM Cloud Infrastructure Center (CIC)
  - Adding IBM CIC Providers

IBM Cloud Infrastructure Center (CIC) Providers

IBM Cloud Infrastructure Center (CIC)

IBM Cloud Infrastructure Center (CIC) is an advanced infrastructure management (OpenStack compatible) software that includes on-premises deployments of IBM z/VM®- and KVM-based Linux® virtual machines on the IBM Z® and LinuxONE platforms.

Adding IBM CIC Providers

Complete the following procedure to add an IBM CIC cloud provider in Infrastructure Automation.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click **Configuration**, then click (**Add a New Cloud Provider**).
3. From the **Type** list, select **IBM Cloud Infrastructure Center**.
4. Enter a **Name** for the provider.
5. Select the appropriate **Zone**.

6. Enter the CIC region in **Provider Region**.
7. Select the appropriate **Domain ID**.
8. (Optional) Enable tenant mapping by toggling the **Tenant Mapping Enabled** option to **On**. This synchronizes resources and users between the IBM CIC provider and Infrastructure Automation. By default, tenant mapping is disabled.
9. Under **Endpoints**, in the **Default** tab:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL without validation**: Authenticate the provider insecurely using SSL.
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option. This is the recommended authentication method.
        - **Non-SSL**: Connect to the provider insecurely using only HTTP protocol, without SSL.
    2. In **CIC API Endpoint (Hostname or IPv4/IPv6 address)**, enter the public IP or fully qualified domain name of the CIC API endpoint.
    3. In **API Port**, set the public port used by the CIC API. By default, CIC is configured to use port 5000.
    4. In the **API Username** field, enter the name of a user in the CIC environment.

       The user must have the **admin** role for the relevant domain.
    5. In the **API Password** field, enter the password for the user.
    6. Click **Validate** to confirm Infrastructure Automation can connect to the CIC provider.
10. Click **Add**.

- IBM Cloud VPC Providers
    - IBM Cloud Virtual Server for VPC
    - Adding IBM Cloud VPC Providers
- Configuring Metrics Collection for the IBM Cloud VPC Provider
- Configuring Event Handling for the IBM Cloud VPC Provider
    - Adding IBM Cloud VPC Regions
    - Disabling IBM Cloud VPC Regions

IBM Cloud VPC Providers

IBM Cloud Virtual Server for VPC

IBM Cloud Virtual Server for VPC (virtual private cloud) consists of virtual server instances within IBM's VPC infrastructure, a public cloud construct. Virtual server instances for VPC enable the deployment of a secured private space in a publicly accessible environment by way of a single-tenant infrastructure in a multi-tenant public cloud. VPCs are logically isolated networks where cloud compute, storage and networking resources are deployed and used to host business workloads, applications and websites. Like networks in traditional on-premises data centers, VPCs utilize cloud security functions and models that match the functionality of those found in offline data centers.

Adding IBM Cloud VPC Providers

Complete the following procedure to add an IBM Cloud VPC provider in Infrastructure Automation.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click **Configuration**, then click **Add a New Cloud Provider**.
3. From the **Type** list, select **IBM Cloud VPC**.
4. Enter a **Name** for the provider.
5. Select the appropriate **Zone**.
6. Select the appropriate **Region**.
7. Under **Endpoint**, enter your **IBM Cloud API Key**.

   For more information on managing IBM Cloud API keys, including how to create one, see Managing user API keys.
8. Click **Validate** to validate the credentials.

   Note: The connection uses port 443. Make sure the port 443 is opened on any firewalls.
9. Click **Add**.

Configuring Metrics Collection for the IBM Cloud VPC Provider

1. On IBM Cloud, navigate to the **Overview** page for the instance in which metrics will be collected
2. Navigate to the **Monitoring** tab and click on **Add Monitoring**
3. Create the IBM Cloud Monitoring instance

Note: An IBM Cloud Monitoring instance with 'platform metrics' enabled can be used to collect metrics for all VM instances in the deployed region

In order to enable metrics functionality on ManageIQ, the instance GUID is required when adding the VPC cloud provider. This can be obtained using the IBM Cloud CLI:

1. Log in to IBM Cloud CLI i.e. **ibmcloud login -sso**
2. Use the following command: **ibmcloud resource service-instance <INSTANCE_NAME> --output json | jq -r '.[].guid'**

Configuring Event Handling for the IBM Cloud VPC Provider

1. On IBM Cloud, create an IBM Cloud Activity Tracker service instance **Note:** Export API will be used therefore **Lite** plan cannot be used
2. Open the dashboard for the instance and navigate to **Settings -> Organization -> API Keys**
3. Copy the service key or generate a new one
4. On ManageIQ, when creating an IBM Cloud VPC provider, place the service key in the field under the **Events** tab on the provider creation form

For further information, see https://cloud.ibm.com/docs/activity-tracker?topic=activity-tracker-service_keys

Adding IBM Cloud VPC Regions

Infrastructure Automation allows administrators to add additional IBM Cloud VPC regions on the appliance server. You can use this capability to add new regions that have been set up since Infrastructure Automation was released. Once added the region it will be available when creating a new IBM Cloud VPC provider.

1. Browse to **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the additional region to that server only.
3. Click on the **Advanced** tab.
4. Search for **:ems_ibm_cloud_vpc:**, and enter the regions you want to add under **:additional_regions:**.

   Example. To add hypothetical `lunar-base-1` `lunar-base-2` regions:

```
:ems_ibm_cloud_vpc:
  :additional_regions:
    :lunar-base-1:
      :name: Lunar Base 1
      :hostname: lunar-base-1.iaas.cloud.ibm.com
      :description: Lunar Base 1
    :lunar-base-2:
      :name: Lunar Base 2
      :hostname: lunar-base-2.iaas.cloud.ibm.com
      :description: Lunar Base 2
```

5. Click **Save**.

Disabling IBM Cloud VPC Regions

Infrastructure Automation allows administrators to disable IBM Cloud VPC regions on the appliance server. You can use this capability to disable certain classified regions. Once disabled, the region will not be available when adding a new IBM Cloud VPC provider.

1. Browse to **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the disabled region to that server only.
3. Click on the **Advanced** tab.
4. Search for **:ems_ibm_cloud_vpc**, and enter the regions you want to disable under **:disabled_regions:**.

   Example. To disable the `au-syd` and `br-sao` regions:

```
:ems_ibm_cloud_vpc:
  :disabled_regions:
  - au-syd
  - br-sao
```

5. Click **Save**.

- IBM Power Systems Virtual Servers Providers
  - IBM Power Systems Virtual Servers

- Adding IBM Power Systems Virtual Servers Providers
- Adding IBM Power Systems Virtual Server Regions
- Disabling IBM Power Systems Virtual Server Regions

IBM Power Systems Virtual Servers Providers

IBM Power Systems Virtual Servers

IBM Power Systems Virtual Servers is a Power Systems infrastructure as a service (IaaS) offering with connectivity to the catalog of IBM Cloud Services that you can use to deploy a virtual server, also known as a logical partition (LPAR), in a matter of minutes. IBM Power Systems clients who have typically relied upon on-premises-only infrastructure can now quickly and economically extend their Power IT resources off-premises.

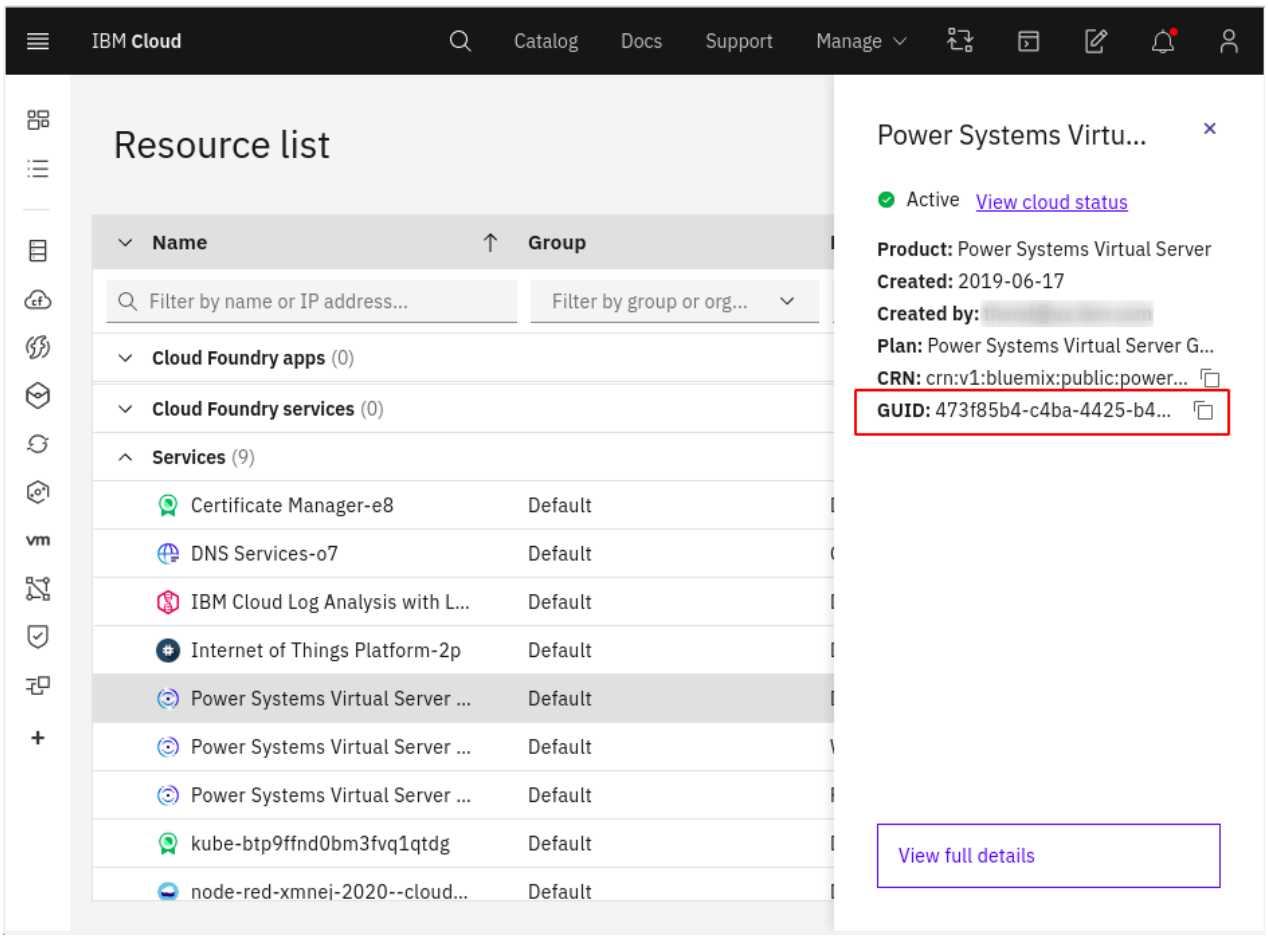
Adding IBM Power Systems Virtual Servers Providers

Complete the following procedure to add an IBM Power Systems Virtual Servers cloud provider in Infrastructure Automation.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click **Configuration**, then click (**Add a New Cloud Provider**).
3. From the **Type** list, select **IBM Power Systems Virtual Servers**.
4. Enter a **Name** for the provider.
5. Select the appropriate **Zone**.
6. Under **Endpoints**, in the **Default** tab.
    1. Enter your **IBM Cloud API Key**.

       For more information on managing IBM Cloud API keys, including how to create one, refer to IBM Cloud Docs: Managing user API keys
    2. Enter the **PowerVS Service GUID** of the IBM Power Systems Virtual Servers service instance you want to add.

       You can easily copy the GUID from the IBM Cloud Resources page, by selecting “Services” and clicking on the whitespace of a PowerVS instance row.



3. Click **Validate** to validate the credentials.

7. Click **Add**.

Adding IBM Power Systems Virtual Server Regions

Infrastructure Automation allows administrators to add additional IBM Power Systems Virtual Server regions on the appliance server. You can use this capability to add new regions that have been set up since Infrastructure Automation was released. Once added the region it will be available when creating a new IBM Power Systems Virtual Server provider.

1. Browse to **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the additional region to that server only.
3. Click on the **Advanced** tab.
4. Search for `:ems_ibm_cloud_power_virtual_servers:`, and enter the regions you want to add under `:additional_regions:`.

```
Example. To add hypothetical `lunar-base-1` `lunar-base-2` regions:

:ems_ibm_cloud_power_virtual_servers:
  :additional_regions:
    :lunar-base-1:
      :name: Lunar Base 1
      :hostname: lunar-base-1.power-iaas.cloud.ibm.com
      :description: Lunar Base 1
    :lunar-base-2:
      :name: Lunar Base 2
      :hostname: lunar-base-2.power-iaas.cloud.ibm.com
      :description: Lunar Base 2
```

5. Click **Save**.

Disabling IBM Power Systems Virtual Server Regions

Infrastructure Automation allows administrators to disable IBM Power Systems Virtual Server regions on the appliance server. You can use this capability to disable default regions. Once disabled, the region will not be available for new and existing IBM Power Systems Virtual Server providers.

1. Browse to **Settings** > **Application Settings**.
2. Click on the **Settings** accordion, then click **Infrastructure Automation: Region** at the top. Optionally, a server can be selected (within Zones) to limit the disabled region to that server only.
3. Click on the **Advanced** tab.
4. Search for `:ems_ibm_cloud_power_virtual_servers:`, and enter the regions you want to disable under `:disabled_regions:`.

```
Example. To disable the `dal` and `eu-de` regions:

:ems_ibm_cloud_power_virtual_servers:
  :disabled_regions:
  - dal
  - eu-de
```

5. Click **Save**.

- IBM PowerVC Providers
  - IBM PowerVC
  - Adding IBM PowerVC Providers
  - Enabling Capacity and Utilization data from IBM PowerVC Providers
  - Enabling Events in IBM PowerVC
  - Viewing IBM PowerVC Events in Infrastructure Automation

IBM PowerVC Providers

IBM PowerVC

IBM PowerVC is an advanced virtualization and cloud management offering. Built on OpenStack, it provides simplified virtualization management and cloud deployments for IBM AIX®, IBM i and Linux virtual machines (VMs) running on IBM Power Systems™. The offering is designed to build private cloud capabilities on Power Systems servers and improve administrator productivity. It can further integrate with cloud environments through higher-level cloud orchestrators.

Adding IBM PowerVC Providers

The corresponding Infrastructure Automation plugin supports IBM PowerVC **v1.4.4 or higher**.

Complete the following procedure to add an IBM PowerVC cloud provider in Infrastructure Automation.

1. Browse to menu: **Compute > Clouds > Providers**.

2. Click **Configuration**, then click (**Add a New Cloud Provider**).
3. From the **Type** list, select **IBM PowerVC**.
4. Enter a **Name** for the provider.
5. Select the appropriate **Zone**.
6. Enter the PowerVC region in **Provider Region**.
7. Select the appropriate **Domain ID**.
8. (Optional) Enable tenant mapping by toggling the **Tenant Mapping Enabled** option to **On**. This synchronizes resources and users between the IBM PowerVC provider and Infrastructure Automation. By default, tenant mapping is disabled.
9. Under **Endpoints**, in the **Default** tab:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL without validation**: Authenticate the provider insecurely using SSL.
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option. This is the recommended authentication method.
        - **Non-SSL**: Connect to the provider insecurely using only HTTP protocol, without SSL.
    2. In **PowerVC API Endpoint (Hostname or IPv4/IPv6 address)**, enter the public IP or fully qualified domain name of the PowerVC API endpoint.
    3. In **API Port**, set the public port used by the PowerVC API. By default, PowerVC is configured to use port 5000.
    4. In the **API Username** field, enter the name of a user in the PowerVC environment.

       The user must have the **admin** role for the relevant domain.
    5. In the **API Password** field, enter the password for the user.
    6. Click **Validate** to confirm Infrastructure Automation can connect to the PowerVC provider.
10. Click **Add**.

Enabling Capacity and Utilization data from IBM PowerVC Providers

In addition to enabling general Capacity and Utilization Collection, complete the following procedure to enable Capacity and Utilization data for an IBM PowerVC cloud provider in Infrastructure Automation.

Virtual Machines for which you want to monitor capacity and utilization data need to be created with a Compute Template with performance information collection enabled.

1. In the IBM PowerVC web interface, navigate to **Virtual Machines > Compute Templates** in the side menu.
2. In the Compute Templates Menu, select the **Create +** button.
3. During creation of the Compute Template, select the **Miscellaneous** tab.
4. Then select the checkbox with the description **Enable performance information collection**.
5. Continue with template creation. When ready, select **Create** to create your template.

Any Virtual Machines created using that template will be able to transmit Capacity and Utilization data to Infrastructure Automation.

Enabling Events in IBM PowerVC

Service events must be enabled in IBM PowerVC in order for Infrastructure Automation to collect and display them.

1. Login to the PowerVC CLI and check if the following services are enabled:

```
powervc-config image audit
powervc-config compute audit
powervc-config storage audit
```

2. If the services are not enabled, use the following commands to enable them:

```
powervc-config image audit --enable --restart
powervc-config compute audit --enable --restart
powervc-config storage audit --enable --restart
```

After enabling the services Infrastructure Automation will be able to collect and display event data.

Viewing IBM PowerVC Events in Infrastructure Automation

In Infrastructure Automation, view events for your IBM PowerVC Provider by following these steps:

1. Browse to menu: **Compute > Clouds > Providers** and select your IBM PowerVC Provider.

2. Go to **Monitoring > Timelines** on the provider summary page to view an events timeline for the provider.

- OpenStack Providers
  - Adding OpenStack Providers
  - Configuring the Overcloud to Store Events

OpenStack Providers

Adding OpenStack Providers

Infrastructure Automation supports operating with the OpenStack **admin** tenant. When creating an OpenStack provider in Infrastructure Automation, select the OpenStack provider's **admin** user because it is the default administrator of the OpenStack **admin** tenant. When using the **admin** credentials, a user in Infrastructure Automation provisions into the **admin** tenant, and sees images, networks, and instances that are associated with the **admin** tenant.

Note:

In OpenStack, you must add **admin** as a member of all tenants that users want to access and use in Infrastructure Automation.

See Cloud Tenants in *Managing Infrastructure and Inventory* for information on working with OpenStack tenants (projects) in Infrastructure Automation.

When adding an OpenStack cloud or infrastructure provider, enable *tenant mapping* in Infrastructure Automation to map any existing tenants from that provider.

This means Infrastructure Automation will create new cloud tenants to match each existing OpenStack tenant; each new cloud tenant and its corresponding OpenStack tenant will have identical resource assignments (including user and role synchronization) with the exception of quotas. Tenant quotas are not synchronized between Infrastructure Automation and OpenStack, and are available for reporting purposes only. You can manage quotas in Infrastructure Automation but this will not affect the quotas created in OpenStack.

During a provider refresh, Infrastructure Automation will also check for any changes to the tenant list in OpenStack. Infrastructure Automation will create new cloud tenants to match any new tenants, and delete any cloud tenants whose corresponding OpenStack tenants no longer exist. Infrastructure Automation will also replicate any changes to OpenStack tenants to their corresponding cloud tenants.

If you leave tenant mapping disabled, Infrastructure Automation will not create cloud tenants or tenant object hierarchy from OpenStack.

Note:

You can set whether Infrastructure Automation should use the Telemetry service or Advanced Message Queueing Protocol (AMQP) for event monitoring. If you choose Telemetry, you should first configure the **ceilometer** service on the overcloud to store events. See Configuring the Overcloud to Store Events for instructions.

For more information, see OpenStack Telemetry (ceilometer) in the Red Hat OpenStack Platform *Architecture Guide*.

Note:

To authenticate the provider using a self-signed Certificate Authority (CA), configure the Infrastructure Automation appliance to trust the certificate using the steps in Using a Self-Signed CA Certificate before adding the provider.

1. Browse to menu: **Compute > Clouds > Providers**.
2. Click **Configuration**, then click 1862 (**Add a New Cloud Provider**).
3. From the **Type** list, select **OpenStack**.
4. Enter a **Name** for the provider.
5. Select the appropriate **Zone** for the provider. By default, the zone is set to **default**.

   Note:

   For more information, see the definition of host aggregates and availability zones in OpenStack Compute (nova) in the Red Hat OpenStack Platform *Architecture Guide*.
6. Enter the OpenStack region in **Provider Region**.
7. (Optional) Select an associated **Openstack Infra Provider**
8. Select the appropriate **API Version** from the list. The default is **Keystone v2**.

   If you select **Keystone v3**, enter the **Keystone V3 Domain ID** that Infrastructure Automation should use. This is the domain of the user account you will be specifying later in the **Default** tab. If domains are not configured in the provider, enter **default**.

   Note:

   Keystone API v3 is required to create cloud tenants on OpenStack cloud providers.

   Note:

   - With Keystone API v3, domains are used to determine administrative boundaries of service entities in OpenStack. Domains allow you to group users together for various purposes, such as setting domain-specific configuration or security options. For more information, see OpenStack Identity (keystone) in the Red Hat OpenStack Platform *Architecture Guide*.

- The provider you are creating will be able to see projects for the given domain only. To see projects for other domains, add it as another cloud provider. For more information on domain management in OpenStack, see Domain Management in the Red Hat OpenStack Platform *Users and Identity Management Guide*.

1. (Optional) Enable tenant mapping by toggling the **Tenant Mapping Enabled** option to **Yes**. This synchronizes resources and users between the OpenStack cloud provider and Infrastructure Automation. By default, tenant mapping is disabled.
2. In the **Default** tab, under **Endpoints**, configure the host and authentication details of your OpenStack provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL without validation**: Authenticate the provider insecurely using SSL.
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option. This is the recommended authentication method.
        - **Non-SSL**: Connect to the provider insecurely using only HTTP protocol, without SSL.
    2. In **Hostname (or IPv4 or IPv6 address)**, enter the public IP or fully qualified domain name of the OpenStack Keystone service.

        Note:

        The hostname required here is also the **OS_AUTH_URL** value in the **~/overcloudrc** file generated by the director (see Accessing the Overcloud in Red Hat OpenStack Platform *Director Installation and Usage*), or the **~/keystonerc_admin** file generated by Packstack (see Evaluating OpenStack: Single-Node Deployment).
    3. In **API Port**, set the public port used by the OpenStack Keystone service. By default, OpenStack uses port 5000 for non-SSL security protocol. For SSL, API port is 13000 by default.
    4. In the **Username** field, enter the name of a user in the OpenStack environment.

        In environments that use Keystone v3 authentication, the user must have the **admin** role for the relevant domain.
    5. In the **Password** field, enter the password for the user.
    6. Click **Validate** to confirm Infrastructure Automation can connect to the OpenStack provider.
3. Next, configure how Infrastructure Automation should receive events from the OpenStack provider. Click the **Events** tab in the **Endpoints** section to start.
    - To use the Telemetry service of the OpenStack provider, select **Ceilometer**. Before you do so, the provider must first be configured accordingly. See Configuring the Overcloud to Store Events for details.
    - If you prefer to use the AMQP Messaging bus instead, or eventing is not enabled on Ceilometer, select **AMQP** and configure the following:
        - Select a **Security Protocol** method.
        - In **Hostname (or IPv4 or IPv6 address)** (of the **Events** tab, under **Endpoints**), enter the public IP or fully qualified domain name of the AMQP host.
        - In the **API Port**, set the public port used by AMQP. By default, OpenStack uses port 5672 for this.
        - In the **Username** field, enter the name of an OpenStack user with privileged access (for example, **admin**). Then, provide its corresponding password in the **Password** field.
        - Click **Validate** to confirm the credentials.
4. Click **Add** after configuring the cloud provider.

Note:

- To collect inventory and metrics from an OpenStack environment, the Infrastructure Automation appliance requires that the adminURL endpoint for the OpenStack environment be on a non-private network. Hence, the OpenStack adminURL endpoint should be assigned an IP address other than `192.168.x.x`. Additionally, all the Keystone endpoints must be accessible, otherwise refresh will fail.
- Collecting capacity and utilization data from an OpenStack cloud provider requires selecting the **Collect for All Clusters** option under **Configuration**, in the settings menu. For information, see Capacity and Utilization Collections in the *General Configuration Guide*.

Configuring the Overcloud to Store Events

By default, the Telemetry service does not store events emitted by other services in a Red Hat OpenStack Platform environment. The following procedure outlines how to enable the Telemetry service on your OpenStack cloud provider to store such events. This ensures that events are exposed to Infrastructure Automation when a Red Hat OpenStack Platform environment is added as a cloud provider.

1. Log in to the undercloud host.
2. Create an environment file called *ceilometer.yaml*, and add the following contents:

```
parameter_defaults:
  CeilometerStoreEvents: true
```

3. Please see the below **NOTE**.

If your OpenStack cloud provider was not deployed through the undercloud, you can also set this manually. To do so:

1. Log in to your Controller node.

2. Edit */etc/ceilometer/ceilometer.conf*, and specify the following option:

   ```
   store_events = True
   ```

Note:

Passing the newly created environment file to the overcloud deployment is environment specific and requires executing commands in particular order depending on use of variables. For further information please see *Director Installation and Usage* in the Red Hat OpenStack Platform documentation.

- Azure Kubernetes Service (AKS) Container Provider

Azure Kubernetes Service (AKS) Container Provider

After initial installation and creation of a Infrastructure Automation environment you can add an Azure Kubernetes Service Container provider.

You will need to retrieve the cluster hostname and the cluster_admin token.

Navigate to your AKS cluster in the Azure Portal (https://portal.azure.com). In the **Overview** tab search for the **API server address**. This is the **hostname** that you will need when you create your provider.

Next you need to retrieve your cluster_admin token, click on the **Connect** button. This will open a panel with instructions on running the **az** command line utility in **Azure Cloud Shell** (you can also run this locally if you have the **az** utility installed).

```
az account set --subscription SUBSCRIPTION_ID
az aks get-credentials --admin --resource-group RESOURCE_GROUP_NAME --name CLUSTER_NAME
Merged "CLUSTER_NAME-admin" as current context in ~/.kube/config
```

Now grab the token for your user out of the **~.kube/config** file. If this is the only cluster configured in your **~/.kube/config** then you can simply run:

```
grep 'token: ' .kube/config
    token:
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwx
```

If you already have multiple clusters configured search for your AKS cluster admin user, it will be named
clusterAdmin_${RESOURCE_GROUP_NAME}_${CLUSTER_NAME}

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click (**Add a New Containers Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **Azure Kubernetes Service**.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **default**.
6. In the **Default** tab, configure the following for the AKS provider:
   1. Select a **Security Protocol** method to specify how to authenticate the provider:
      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
      - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
      - **SSL without validation**: Authenticate the provider insecurely (not recommended).
   2. Enter the **Hostname** of the provider that you obtained earlier.

      Important:

      The **Hostname** must use a unique fully qualified domain name.
   3. Enter the **API Port** of the provider. The default port is **443**.
   4. Enter the token that you obtained earlier for your provider in the **Token** box.
   5. Click **Validate** to confirm that Infrastructure Automation can connect to the Azure Kubernetes Service provider.
7. Click **Add**.

- Red Hat OpenShift
  - Obtaining an OpenShift Container Platform Management Token
  - Adding an OpenShift Container Platform Provider
  - Enabling OpenShift Cluster Metrics

Red Hat OpenShift

Obtaining an OpenShift Container Platform Management Token

First you have to create a service-account with the proper permissions for Infrastructure Automation.

1. Create a namespace for the service account

```
project_name="management-manageiq" # Pick a name for your project
oc adm new-project $project_name --description="ManageIQ Project"
```

2. Create a service account in that project

```
service_account_name="management-admin"
oc create serviceaccount $service_account_name -n $project_name
```

3. Create the cluster role

```
echo '{"apiVersion": "authorization.openshift.io/v1", "kind": "ClusterRole", "metadata": {"name": "management-
manageiq-admin"}, "rules": [{"resources": ["pods/proxy"], "verbs": ["*"]}]}' | oc create -f -
```

4. Apply roles and policies to the service account

```
oc policy add-role-to-user -n $project_name admin -z $service_account_name
oc policy add-role-to-user -n $project_name management-manageiq-admin -z $service_account_name
oc adm policy add-cluster-role-to-user cluster-reader system:serviceaccount:$project_name:$service_account_name
oc adm policy add-scc-to-user privileged system:serviceaccount:$project_name:$service_account_name
oc adm policy add-cluster-role-to-user self-provisioner system:serviceaccount:$project_name:$service_account_name
```

Note:

See the OpenShift Container Platform documentation for a list of the default roles.

Run the following to obtain the token needed to add an OpenShift Container Platform provider:

Run the following to obtain the token needed to add a Red Hat OpenShift provider:

1. Obtain the **management** service account token name:

```
# oc describe sa -n $project_name $service_account_name
...
Tokens:  management-admin-token-0f3fh
         management-admin-token-q7a87
```

2. Select and describe one of the tokens to retrieve the full token output, replacing **management-admin-token-0f3fh** with the name of your token:

```
# oc describe secret -n $project_name management-admin-token-0f3fh
...
Data
====
token:  eyJhbGciOiJSUzI1NiI...
```

Adding an OpenShift Container Platform Provider

After initial installation and creation of a Infrastructure Automation environment, add an OpenShift Container Platform provider using the token obtained in Obtaining an OpenShift Container Platform Management Token and following the procedure below.

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click Add a New Containers Provider (**Add a New Containers Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **OpenShift Container Platform**.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **default**.
6. In the **Default** tab, configure the following for the OpenShift provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.

          Note:

          You can obtain your OpenShift Container Platform provider's CA certificate for all endpoints (default, metrics, alerts) from **/etc/origin/master/ca.crt**. Paste the output (a block of text starting with **-----BEGIN CERTIFICATE-----**) into the **Trusted CA Certificates** field.
        - **SSL without validation**: Authenticate the provider insecurely (not recommended).
    2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.

       Important:

The **Hostname** must use a unique fully qualified domain name.

3. Enter the **API Port** of the provider. The default port is **6443**.
4. Enter a token for your provider in the **Token** box.

   Note:

   To obtain a token for your provider, run the **oc get secret** command on your provider; see Obtaining an OpenShift Container Platform Management Token.

   For example:

   # oc get secret –namespace management-infra management-admin-token-8ixxs –template='' | base64 –decode

5. Click **Validate** to confirm that Infrastructure Automation can connect to the OpenShift Container Platform provider.

7. In the **Metrics** endpoint tab optionally configure the metrics service details:
   1. Select a service type (e.g.: Prometheus) or leave disabled
   2. Select a **Security Protocol** method to specify how to authenticate the service:
      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
      - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
      - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)
   3. Enter the **Hostname** (or IPv4 or IPv6 address) of the metrics service.

      In order to find the hostname you can use the **oc get route** command.

      You can retrieve the Prometheus route by running:

      ```
      oc get route prometheus-k8s -n openshift-monitoring
      ```

      Example:

      ```
      oc get route prometheus-k8s -n openshift-monitoring
      NAME             HOST/PORT                                          PATH              SERVICES   PORT
      TERMINATION          WILDCARD
      prometheus-k8s   prometheus-k8s-openshift-monitoring.ocp.example.com  prometheus-k8s   web
      reencrypt/Redirect   None
      ```

      In this example you would use **prometheus-k8s-openshift-monitoring.ocp.example.com** as the metrics hostname.

   4. Enter the **API Port** of your metrics endpoint uses a non-standard port for access. The default port is **443**.
   5. Click **Validate** to confirm that Infrastructure Automation can connect to the metrics endpoint.

8. In the **Virtualization** endpoint tab optionally configure the virtualization service details:
   1. Select a service type (OpenShift Virtualization) or leave disabled
   2. Select a **Security Protocol** method to specify how to authenticate the service:
      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
      - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
      - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)
   3. Enter the **Hostname** (or IPv4 or IPv6 address) of the virtualization service.
   4. Enter the **API Port** of your virtualization service. The default port is **443**.
   5. Enter a token for your provider in the **Token** box.

9. Click the **Advanced** tab to add image inspector settings for scanning container images on your provider using OpenSCAP.

   Note:

   - These settings control downloading the image inspector container image from the registry and obtaining the Common Vulnerabilities and Exposures (CVE) information (for effective scanning) via a proxy.
   - CVE URL that Infrastructure Automation requires to be open for OpenSCAP scanning: https://www.redhat.com/security/data/metrics/ds/. This information is based on the source code of OpenSCAP.

   1. Enter the proxy information for the provider in either **HTTP**, **HTTPS**, or **NO Proxy** depending on your environment.
   2. Enter the **Image-Inspector Repository** information. For example, **openshift3/image-inspector**.

3. Enter the **Image-Inspector Registry** information. For example, `registry.access.redhat.com`.
4. Enter the **Image-Inspector Tag** value. A tag is a mark used to differentiate images in a repository, typically by the application version stored in the image.
5. Enter `https://www.redhat.com/security/data/metrics/ds/` in **CVE location**.

10. Click **Add**.

Note:

You can also set global default image-inspector settings for all OpenShift providers in the advanced settings menu by editing the values under `ems_kubernetes`, instead of setting this for each provider.

For example:

```
:image_inspector_registry: registry.access.redhat.com
:image_inspector_repository: openshift3/image-inspector
```

Enabling OpenShift Cluster Metrics

Use the OpenShift Cluster Metrics plug-in to collect node, pod, and container metrics into one location. This helps track usage and find common issues.

- Configure Infrastructure Automation to allow for all three Capacity & Utilization server roles.
- Enable cluster metrics using the OpenShift Container Platform documentation.
- IBM Cloud Kubernetes Service (IKS) Container Provider
  - Configuring Metrics Collection for IBM Cloud Kubernetes Service (IKS) Provider

IBM Cloud Kubernetes Service (IKS) Container Provider

After initial installation and creation of a Infrastructure Automation environment you can add an IBM Cloud Kubernetes Service Container provider.

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click (**Add a New Containers Provider**).
3. From the **Type** list, select **IBM Cloud Kubernetes Service**.
4. Enter a **Name** for the provider.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to `default`.
6. Select a **Security Protocol** method to specify how to authenticate the provider:

```
- **SSL**: Authenticate the provider securely using a trusted
Certificate Authority. Select this option if the provider
has a valid SSL certificate and it is signed by a trusted
Certificate Authority. No further configuration is required
for this option.

- **SSL trusting custom CA**: Authenticate the provider with a
self-signed certificate. For this option, copy your
provider's CA certificate to the **Trusted CA
Certificates** box in PEM format.

- **SSL without validation**: Authenticate the provider
insecurely (not recommended).
```

7. Enter the **Hostname** and **API Port** of the cluster. On IBM Cloud, these values can be retrieved by accessing the `Overview` page for your cluster. Under `Networking`, click `Copy link` to copy the `Public enabled` service endpoint URL for your cluster. The URL will be copied in the following format: `https://<Hostname>:<API Port>` and the necessary values can be extracted.
8. Enter your IBM Cloud **API Key**. For detailed instructions on how to create an IBM Cloud API key, see https://cloud.ibm.com/docs/account?topic=account-userapikey&interface=ui#create_user_key **Note:** The API key must belong to the IBM Cloud user that created the cluster
9. Click **Validate** to confirm that Infrastructure Automation can connect to the IBM Cloud Kubernetes Service.
10. Click **Add**.

Configuring Metrics Collection for IBM Cloud Kubernetes Service (IKS) Provider

1. On IBM Cloud, create an `IBM Cloud Monitoring` service instance
2. Log in to IBM Cloud CLI i.e. `ibmcloud login -sso`
3. Use the following command to retrieve the monitoring instance ID: `ibmcloud resource service-instance <INSTANCE_NAME> --output json | jq -r '.[].guid'`

Refer to the following table to determine the hostname for the **IBM Cloud Monitoring** service instance:

Region	Hostname
US South	us-south.monitoring.cloud.ibm.com
EU DE	eu-de.monitoring.cloud.ibm.com
EU GB	eu-gb.monitoring.cloud.ibm.com
JP OSA	jp-osa.monitoring.cloud.ibm.com
JP TOK	jp-tok.monitoring.cloud.ibm.com
US East	us-east.monitoring.cloud.ibm.com
AU SYD	au-syd.monitoring.cloud.ibm.com
CA TOR	ca-tor.monitoring.cloud.ibm.com
BR SAO	br-sao.monitoring.cloud.ibm.com

1. Enter the **Hostname** and **Monitoring Instance ID** under the **Metrics** tab when creating the IKS provider

- Oracle Kubernetes Engine (OKE) Container Provider

Oracle Kubernetes Engine (OKE) Container Provider

After initial installation and creation of a Infrastructure Automation environment you can add an Oracle Kubernetes Engine Container provider.

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click (**Add a New Containers Provider**).
3. From the **Type** list, select **Oracle Kubernetes Engine**.
4. Enter a **Name** for the provider.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **default**.
6. Select the provider region where your Oracle Kubernetes Cluster is located
7. Next you will need to gather your Oracle Tenant ID from https://cloud.oracle.com/tenancy, and use the **Copy** link to fetch your tenant OCID.
8. Now you need to enter the Cluster ID of your OKE cluster. This can be found on the **Cluster Details** tab on your cluster's dashboard at **Cluster Id**. Use the **Copy** link to copy this ID.
9. In the **Default** tab, configure the following for the OKE provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
        - **SSL without validation**: Authenticate the provider insecurely (not recommended).
    2. Enter the **Hostname** of the cluster, you can obtain this from the **Cluster Details** page at **Kubernetes API Public Endpoint**
    3. Enter the **API Port** of the cluster. The default port is **6443**.
    4. Enter the Oracle User ID for the user that you want Infrastructure Automation to use to access your account. You can get this from the **Profile** dropdown and clicking on your username. Then use the **Copy** link to copy the OCID for your user.
    5. Next copy and paste the public key for your user's Auth Token. If you do not have one yet you can create an Auth Token by going to your user profile and selecting **Auth Tokens** under **Resources** and select the **Generate Token** button.
    6. Now copy and paste your private key from the same auth token.
    7. Click **Validate** to confirm that Infrastructure Automation can connect to the Oracle Kubernetes Engine provider.
10. Click **Add**.

- VMware Tanzu Container Provider

VMware Tanzu Container Provider

After initial installation and creation of a Infrastructure Automation environment you can add a VMware Tanzu Container provider.

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click (**Add a New Containers Provider**).

3. From the **Type** list, select **VMware Tanzu**.
4. Enter a **Name** for the provider.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **`default`**.
6. In the **Default** tab, configure the following for the VMware Tanzu provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
        - **SSL without validation**: Authenticate the provider insecurely (not recommended).
    2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.

        Important:

        The **Hostname** must use a unique fully qualified domain name.
    3. Enter the **API Port** of the provider. The default port is **`8443`**.
    4. Enter the **Username** of your user. This user must have cluster level access to the VMware Tanzu cluster.
    5. Enter the **Password** of your user.
    6. Click **Validate** to confirm that Infrastructure Automation can connect to the VMware Tanzu provider.
7. Click **Add**.

- Amazon Elastic Block Store Managers

Amazon Elastic Block Store Managers

The Amazon Elastic Block Store service provides and manages persistent block storage resources that Amazon EC2 instances can consume.

To use the Amazon Elastic Block Store service as a storage manager, you must first add an Amazon EC2 cloud provider to your Infrastructure Automation appliance. The Amazon Elastic Block Store service is automatically discovered by Infrastructure Automation, and added to the **Storage Managers** list. See Adding Amazon EC2 Providers for instructions on adding an Amazon EC2 cloud provider.

- OpenStack Block Storage Managers
    - OpenStack Block Storage Manager Targeted Refresh
- Viewing a Cinder Manager's Timeline

OpenStack Block Storage Managers

The OpenStack Block Storage service (**`openstack-cinder`**) provides and manages persistent block storage resources that OpenStack infrastructure instances can consume.

To use OpenStack Block Storage as a storage manager, you must first add an OpenStack cloud provider to your Infrastructure Automation appliance and enable events. The Block Storage service will be automatically discovered by Infrastructure Automation and added to the **Storage Managers** list in Infrastructure Automation. See Adding OpenStack Providers for instructions on adding a cloud provider and enabling events.

OpenStack Block Storage Manager Targeted Refresh

Infrastructure Automation supports targeted refresh of OpenStack Block Storage (Cinder) volumes. This functionality automatically pulls down new inventory data for an object when Infrastructure Automation receives an event from the OpenStack provider. For example, if Infrastructure Automation receives a **`volume.update.end`** event from Cinder, instead of refreshing the entire inventory, it performs a targeted refresh of the volume referenced in the event.

Viewing a Cinder Manager's Timeline

In Infrastructure Automation, view events for a Cinder storage manager by following these steps:

1. Browse to menu: **Storage > Storage Managers**.
2. Select your OpenStack Cinder manager to go to the Cinder manager's summary page.
3. Click Monitoring (**Monitoring**), and then Timelines (**Timelines**) to view the events timeline for the manager.

4. A timeline of either management events or policy events can be viewed.
    1. To view management events, select **Management Events**.
    2. Specify the type of event to view.
    3. Specify the timeline for the events to view.
    4. Click **Apply**.
5. To view policy events, select **Policy Events**.
    1. Specify if you want to view successful events, failed events, or both.
    2. Specify the timeline for the events to view.
    3. Click **Apply**.

- OpenStack Object Storage Managers
    - Viewing Object Stores
- Viewing a Swift Manager's Timeline

OpenStack Object Storage Managers

The OpenStack Object Storage (`openstack-swift`) service provides cloud object storage.

To use the OpenStack Object Storage service as a storage manager, you must first add an OpenStack cloud provider to your Infrastructure Automation appliance and enable events. The Object Storage service will be automatically discovered by Infrastructure Automation and added to the **Storage Managers** list in Infrastructure Automation. See Adding OpenStack Providers for instructions on adding a cloud provider and enabling events.

Viewing Object Stores

The object store summary page shows details including the object store's size, parent cloud, storage manager, cloud tenant, and the number of cloud objects on the object store.

In Infrastructure Automation, view object stores on a object storage manager by following these steps:

1. Browse to menu: **Storage > Object Stores** to display a list of object store containers.
2. Click a container to open a summary page for that object store container.
3. Click **Cloud Objects** to view a list of object stores in the object store container.
4. Click an object store from the list to view the object store's summary page.

Viewing a Swift Manager's Timeline

In Infrastructure Automation, view events for a Swift storage manager by following these steps:

1. Navigate to **Storage > Storage Providers**.
2. Select your OpenStack Swift manager to go to the Swift manager's summary page.
3. Click Monitoring (**Monitoring**), and then Timelines (**Timelines**) to view the events timeline for the manager.
4. View a timeline of either management events or policy events:
    1. To view management events, select **Management Events**.
    2. Specify the type of event to view.
    3. Specify the timeline for the events to view.
    4. Click **Apply**.
5. To view policy events, select **Policy Events**.
    1. Specify if you want to view successful events, failed events, or both.
    2. Specify the timeline for the events to view.
    3. Click **Apply**.

- IBM Cloud Object Storage Managers
    - Adding IBM Cloud Object Storage:

IBM Cloud Object Storage Managers

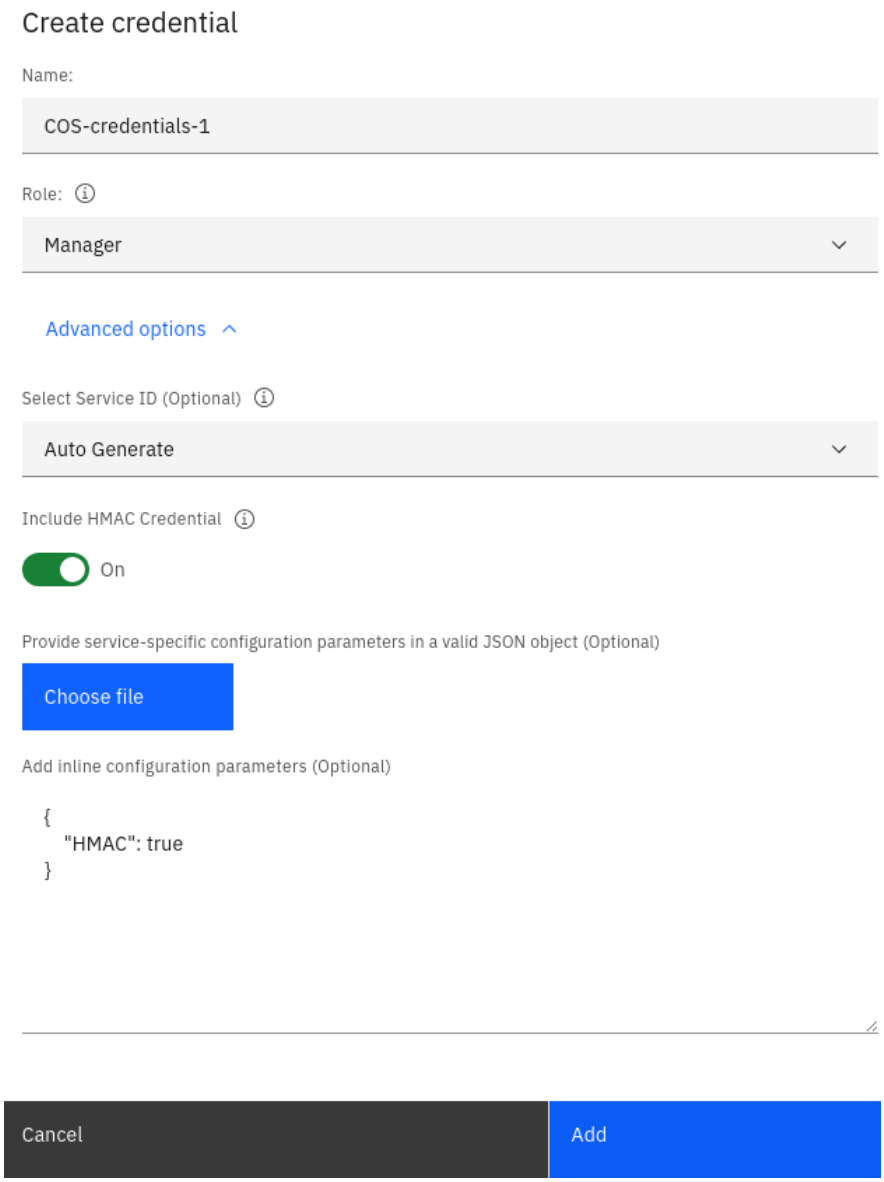
IBM Cloud Object Storage is a highly available, durable, and secure platform for storing unstructured data. Unstructured data (sometimes called binary or “blob” data) refers to data that is not highly structured in the manner of a database. Object storage is the most efficient way to store PDFs, media files, database backups, disk images, or even large structured datasets.

The files that are uploaded into IBM Cloud Object Storage are called objects. They are organized into buckets that serve as containers for objects, and which can be configured independently from one another in terms of locations, resiliency, billing rates, security, and object lifecycle.

Adding IBM Cloud Object Storage:

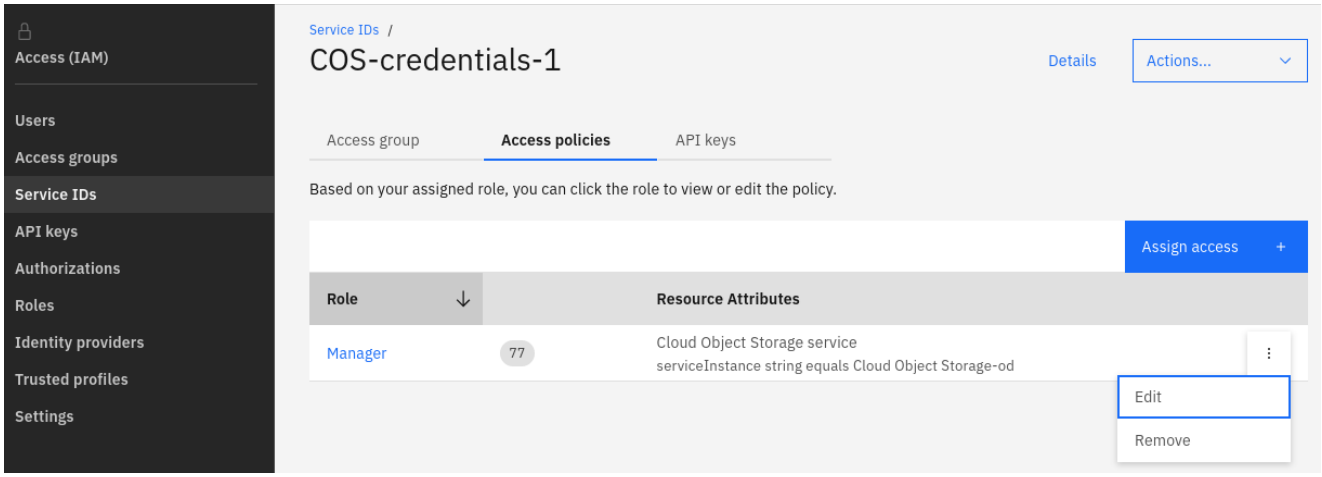
The Cloud Object Storage provider in Infrastructure Automation is region centric. When registering this provider a given region must be chosen such that only buckets and objects of that region will be available / visible to the provider user in Infrastructure Automation. For a list of supported regions please refer to this page.

1. Create new Cloud Object Storage instance in IBM Cloud Console.
2. Create access credentials by navigating to **Service Credentials -> New Credential** in your new instance.

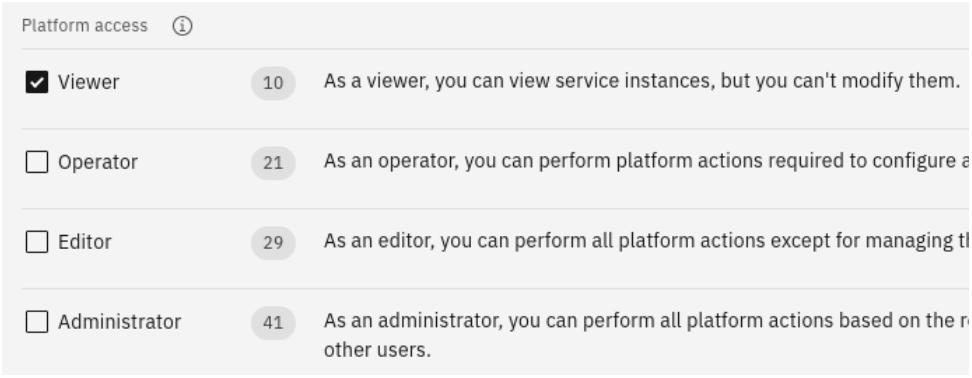


Make sure to choose the **Manager** role and to enable **Include HMAC Credential** option.

3. Next navigate to **`Manage -> Access (IAM) -> Service IDs`**, find your newly created credential and edit it.



4. Select **`Viewer`** checkbox under **`Platform Access`** group and press **`Add`**.



5. Browse to Infrastructure Automation menu **Storage > Managers**.
6. Click **Configuration**, then click (**Add a New Storage Provider**).
7. From the **Type** list, select **IBM Cloud Object Storage**.
8. Enter a **Name** for the storage manager.
9. Set **default** as a value for the **Zone**
10. Enter the desired region in the corresponding **Region** field. As explained above your provider will be region-specific.
11. Specify the **Resource Instance Id**, which you can find as the value of the **`resource_instance_id`** item of the credentials.
12. **Endpoint** depends on the region you have chosen to connect your provider to. For more explanation and the listing of the endpoints please refer to this page. If your region is e.g. **`us-south`** then the public endpoint for it would be **`https://s3.us-south.cloud-object-storage.appdomain.cloud`**.
13. Specify the **Apikey**, which you can find as the value of the **`apikey`** item of the credentials.
14. Specify the **Access Key**, which you can find as the value of the **`cos_hmac_keys -> access_key_id`** item of the credentials.
15. Specify the **Secret Key**, which you can find as the value of the **`cos_hmac_keys -> secret_access_key`** item of the credentials.
16. Once you have inputted the above values press on **Validate** button and wait for the validation to complete. Upon successful validation the **Add** button will be enabled, clicking on which will add the new provider.

- Provisioning Virtual Machines and Instances
- Infrastructure Automation Lifecycle
  - Provisioning
  - Provisioning Requests
    - Requirements for Provisioning Virtual Machines and Instances
    - Requirements for Provisioning Virtual Machines from Red Hat Virtualization Manager
    - PXE Provisioning
      - Connecting to a PXE Server
      - Creating System Image Types for PXE
      - Setting the PXE Image Type for a PXE Image
    - ISO Provisioning
      - Adding an ISO Datastore
      - Refreshing an ISO Datastore
      - Creating System Image Types for ISO
      - Setting the Image Type for an ISO Image

- Customization Templates for Virtual Machine and Instance Provisioning
- Customization Script Additions for Virtual Machine and Instance Provisioning
- Adding a Customization Template
- Provisioning Virtual Machines
  - Provisioning a Virtual Machine from a Template
  - Provisioning a Virtual Machine using Cloud-Init via REST API
  - Cloning a Virtual Machine
  - Publishing a Virtual Machine to a Template (VMware Virtual Machines Only)
  - Renaming a Provisioned Virtual Machine (VMware Virtual Machines Only)
- Provisioning Instances
  - Provisioning an EC2 Instance from an Image
  - Provisioning an OpenStack Instance from an Image, Volume or Volume Snapshot
  - Provisioning an IBM Power Systems Virtual Servers Instance from an Image
  - Customizing Provisioning Dialogs
    - Adding a Provision Dialog for All Users
    - Creating a Custom Provision Dialog
  - Provisioning Profiles
    - Creating a Provisioning Profile Instance
    - Setting Provisioning Scope Tags
  - Managing Key Pairs

- Working with Requests
  - Provisioning Request Approval Methods
  - Working with Provisioning Requests
    - Reloading the Status of Provisioning Requests
    - Approving a Provisioning Request
    - Denying a Provisioning Request
    - Copying a Provisioning Request
    - Editing a Provisioning Request
    - Deleting a Provisioning Request
    - Automatically Approving Requests
      - Enabling Global Defaults for Automatic Approval
      - Template Specific Approval Defaults
      - Assigning Tags to a Template for Auto Approval
      - Setting Provisioning Notification Email Addresses
- Fulfilling Requests
  - Fulfilling a Request
  - Default Execution Steps in States Instance
    - Quotas
      - Applying User or Group Quotas
      - Using Tags for Owner and Group Quotas
      - Applying a Tag to a User or User Group
      - State Machines
      - Customizing Provisioning States
      - Editing the Default State Instance
      - Viewing the Status of a Provisioning Request
      - Viewing a Provisioned Virtual Machine or Instance
      - Viewing a Virtual Machine or Instance Summary
- Catalogs and Services
  - Generic Objects
    - Viewing Generic Objects Classes
    - Creating Generic Objects Classes
    - Editing Generic Object Classes
    - Removing Generic Objects Classes
    - Exporting Generic Objects
    - Importing Generic Objects
  - Service Dialogs
    - Adding a Service Dialog
      - Creating a Service Dialog from a Container Template
      - Importing Service Dialogs
      - Exporting Service Dialogs
    - Methods
      - Creating a Method to Associate with the Dialog
      - Creating a Method in the Service Class
      - Creating an Instance in the Service Class
      - Associating a Method with an Automate Instance
  - Provisioning Dialog
    - Create a workflow (optional)
    - Add a Workflow Repository
    - Create a Provisioning Dialog
    - Create a Service Catalog Item

        - Provisioning a Virtual Machine
        - Iterate
    - Catalogs
        - Creating a Catalog Bundle
        - Copying a Catalog Bundle
        - Creating a Catalog Item
        - Copying a Catalog Item
        - Creating a Generic Catalog Item
        - Creating an Ansible Playbook Service Catalog Item
        - Creating an Ansible Tower Service Catalog Item
        - Creating an Amazon Service Catalog Item
        - Creating an Azure Service Catalog Item
        - Creating an OpenShift Template Catalog Item
        - Creating an Orchestration Catalog Item
        - Creating a VMware Content Library OVF Template Catalog Item
        - Provisioning a Service
    - Orchestration Stacks
        - Creating an Orchestration Template
        - Editing Orchestration Templates
        - Copying Orchestration Templates
        - Deleting Orchestration Templates
- Retirement
    - Retiring Virtual Machines
        - Retiring Virtual Machines and Instances
        - Using the Console to Retire a Virtual Machine
        - Retiring a Virtual Machine Immediately
        - Setting a Retirement Date and Time for a Virtual Machine or Instance
        - Removing a Retirement Date for a Virtual Machine or Instance
    - Setting Retirement in a Provision Request
        - Scheduling Retirement in a Provision Request
    - Extending Retirement Dates
        - Creating a Custom Button to Extend Retirement
        - Changing the Number of Days to Extend Retirement

Provisioning Virtual Machines and Instances

Infrastructure Automation Lifecycle

This guide discusses lifecycle activities such as provisioning and retirement that are part of the Infrastructure Automation Automate component. Infrastructure Automation Automate enables real-time, bidirectional process integration and adaptive automation for management events and administrative or operational activities.

- Operations Management with service level resource enforcement.
- Resource Management including datastore cleanup, snapshot aging and enforcement, and virtual machine or instance aging and retirement.
- Configuration and Change Management including enforced closed loop change management.
- Lifecycle Management such as provisioning, customization, reconfiguration, approval, CMDB updates, and retirement.

Provisioning requires the **Automation Engine** server role enabled. Check your server role settings in the settings menu, **Configuration > Server > Server Control**.

Provisioning

When a virtual machine or cloud instance is provisioned, it goes through multiple phases. First, the request must be made. The request includes ownership information, tags, virtual hardware requirements, the operating system, and any customization of the request. Second, the request must go through an approval phase, either automatic or manual. Finally, the request is executed. This part of provisioning consists of pre-processing and post-processing. Pre-processing acquires IP addresses for the user, creates CMDB instances, and creates the virtual machine or instance based on information in the request. Post-processing activates the CMDB instance and emails the user. The steps for provisioning may be modified at any time using Infrastructure Automation.



Provisioning Requests

The following options are available when making provisioning requests:

- Set an owner (User can do this using LDAP lookup)
- Assign a purpose (tag)
- Select a template or image from which to create a new virtual machine or instance respectively
- Choose placement
- Set hardware requirements
- Specify the vLan
- Customize the guest operating system
- Schedule the provisioning



Requirements for Provisioning Virtual Machines and Instances

Infrastructure Automation supports the provisioning of VMware ESX hypervisors. To provision a virtual machine from VMware providers, you must have an appliance with the Automation Engine role enabled.

If you are using a Windows template, the following configuration is required:

- To customize settings that are inside the operating system, Sysprep must be copied to the appropriate directory on your vCenter computer. Usually this location is: **`C:\Documents and Settings\All Users\Application Data\VMware\VMware VirtualCenter\sysprep`**. Copy the Sysprep tools to the relevant operating system subdirectory. If you are running a standard Win2008 operating system, this step is unnecessary as Sysprep is included as standard.
- The Windows template must have the latest version of VMware tools for its ESX Server. Check the VMware Site for more information. If you are creating a new password for the Administrator account, the Administrators password must be blank on the template. This is a limitation of Microsoft Sysprep.

See the VMware documentation for a complete list of customization requirements.

Requirements for Provisioning Virtual Machines from Red Hat Virtualization Manager

Item	Requirements
Red Hat Virtualization Manager version 4.0 or higher	Red Hat Virtualization Manager properly installed with API in default location https://server:8443/api
Red Hat Virtualization Manager History Database	Red Hat Virtualization Manager Data Warehouse (DWH) properly installed with access to the PostgreSQL database on the Red Hat Virtualization Manager server. Port 5432 open in iptables. md5 authentication allowed to Infrastructure Automation appliances in **`pg_hba.conf`**. PostgreSQL set to listen for connections on **`*:5432`** in **`postgresql.conf`**. Credentials provided during database setup to be used in Infrastructure Automation UI.
Storage Supported for Infrastructure Automation Virtual Machine Analysis	NFS - Infrastructure Automation server must be able to mount NFS storage domain. iSCSI / FCP - Cluster must use full Red Hat Enterprise Linux (not Red Hat Virtualization Hypervisor) Hosts. DirectLUN Hook installed on each host and registered to Red Hat Virtualization Managers. Must have Infrastructure Automation appliance in each Cluster with this storage type. Infrastructure Automation appliance virtual machine container must have DirectLUN attribute set. Local storage - Not yet supported (Red Hat does not recommend due to single point of failure).

PXE Provisioning

PXE is a boot method that allows you to load files from across a network link. Infrastructure Automation uses it for files required for provisioning virtual machines. PXE can be used for provisioning for either Red Hat Virtualization Manager or VMware.

Procedure Overview

1. Connect to the **PXE Server**.
2. Create a **System Image Type**.
3. Associate each **PXE** image with an image type.
4. Create a customization template.

Requirements for PXE Provisioning

- DHCP server configured with required PXE implementation
- PXE implementation for Linux virtual machine provisioning
- NFS or SAMBA read and write access to create and modify files on the PXE server
- Infrastructure Automation Server uses NFS mount to read and write the response files
- HTTP read access to the NFS share location as virtual machines use this URL to access PXE images and Kickstart or Cloud-Init configuration files
- Operating system installation media available to be streamed from PXE server
- Images configured for desired operating systems
- Kickstart or Cloud-Init templates to configure operating systems with desired packages

Additional Requirements for Provisioning Linux Virtual Machines

- Linux distribution kernel and ramdisk available over HTTP
- Linux sources available over HTTP
- Sample PXE menu item that boots this kernel

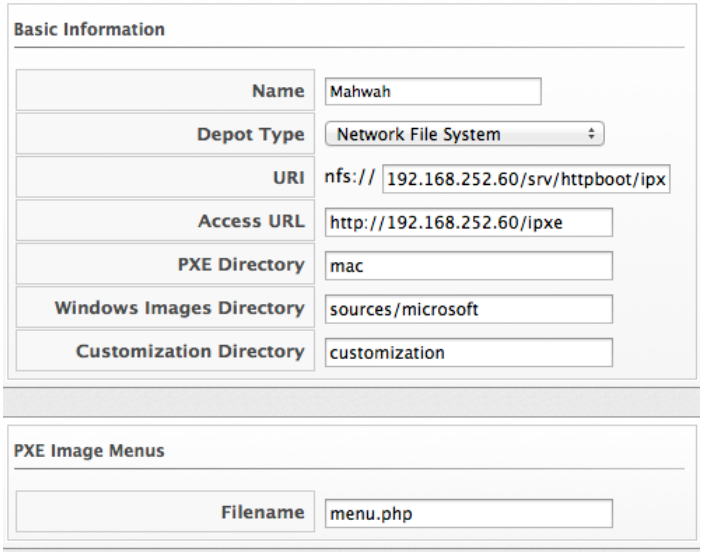
Additional Requirements for Provisioning Windows Virtual Machines

- WinPE ISO built with rhev-agent-tools (for RHEV-M environments) and configured to mount shares for Windows source files and Sysprep files and configured to run customization script
- Windows based WIM file with operating system installed and configured with Sysprep
- Sample Sysprep unattend file to be used with the operating system
- Sample PXE menu item that downloads WinPE ISO, mount in memdisk and boot into WinPE environment

Connecting to a PXE Server

The following procedure connects to a PXE server and adds its details to Infrastructure Automation.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click (**Configuration**), then (**Add a New PXE Server**).
3. In **Basic Information**, type a **Name** that will be meaningful in your environment.



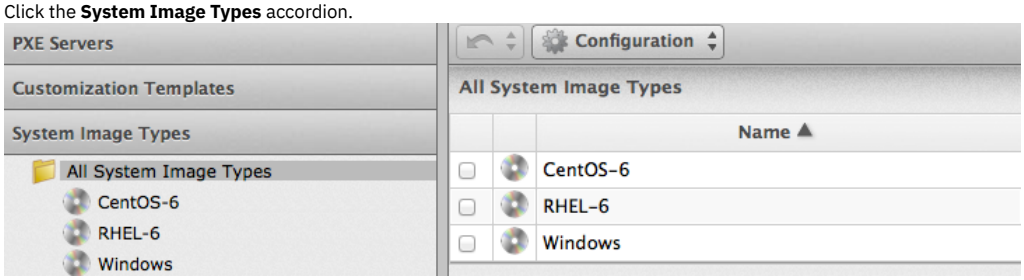
4. For **Depot Type**, select either **Network File System** (NFS) or **Samba**. The fields to enter in the dialog depend on the **Depot Type**.
   - For NFS, type in the **URI**, **Access URL**, **PXE Directory**, **Windows Images Directory**, and **Customization Directory**. When you provision, Infrastructure Automation writes a text file to the **PXE Directory**. The file is named after the MAC address of the NIC that is assigned to the virtual machine. It contains where to get the kernel and initrd image. This file is removed after a successful provision. The **Windows Images Directory** is where the files are located on your NFS for the provisioning of Windows operating systems. The **Customization Directory** is where your Kickstart and Sysprep files are located.
   - If using a **Depot Type** of **Samba**, you will not need **Access URL**, but you will need a **User ID**, and **Password**, in addition to the items required for NFS.
5. For **PXE Image Menus**, type the **Filename** for the PXE Boot menu.
6. Click **Add**.
7. Select the new PXE server from the tree on the left, and click (**Configuration**), then (**Refresh**) to see your existing images.

Next, create PXE Image types to associate with the customization templates and to specify if the image type is for a virtual machine, a host, or both.

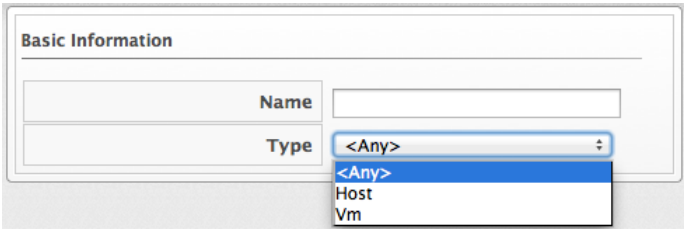
Creating System Image Types for PXE

The following procedure creates a system image type for PXE servers.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **System Image Types** accordion.



3. Click (**Configuration**), then (**Add a new System Image Type**).
4. In **Basic Information**, type in a **Name** and select a **Type**.



   - Use **Vm** if you want this image type to only apply to virtual machines.
5. Click **Add**.

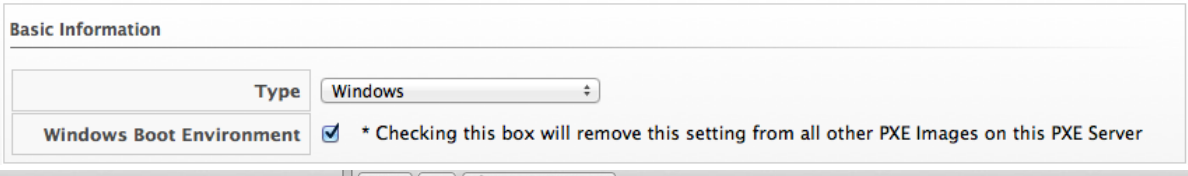
Example:

After creating the System Image Types, assign the types to each image on your PXE servers. To do this, you will select each image on the PXE server and identify its type.

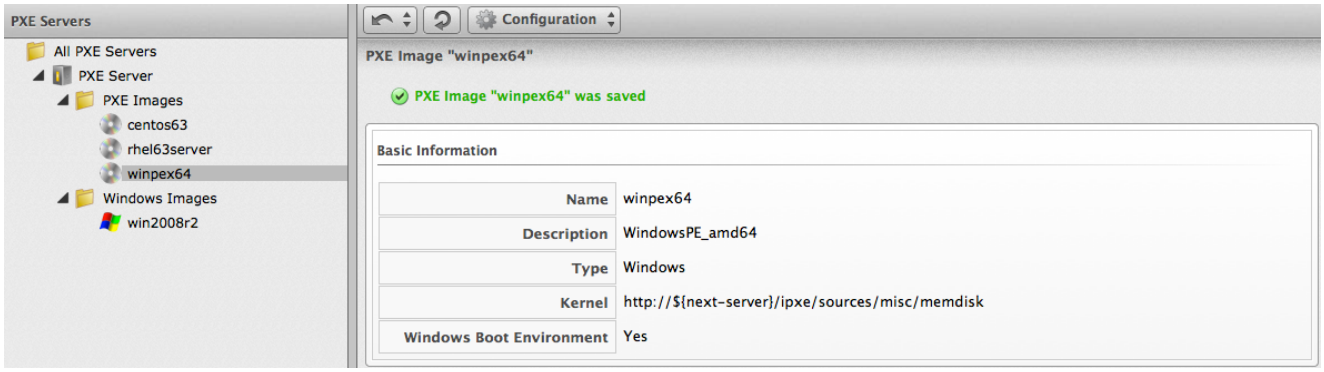
Setting the PXE Image Type for a PXE Image

The following procedure sets the image type for a chosen PXE image.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **PXE Servers** accordion and select the image that you want to set a type for.
3. Click (**Configuration**), then (**Edit this PXE Image**).
4. From the **Basic Information** area, select the correct type. If this PXE image will be used as the **Windows Boot Environment**, check **Windows Boot Environment**. At the time of this writing, only one PXE Image can be identified as the **Windows Boot Environment**. Therefore, checking one as the **Windows Boot Environment**, will remove that from any other PXE image with that check.



Click **Save**.



ISO Provisioning

Infrastructure Automation also allows ISO provisioning from Red Hat Virtualization Manager datastores. To use this feature, you will need to do the following before creating a provision request.

1. Add the **ISO Datastore**. The Red Hat Virtualization Manager system must have already been discovered or added into the VMDB. For more information, see Adding a Red Hat Enterprise Virtualization Manager Provider in *Managing Providers*.
2. Refresh the **ISO Datastore**.
3. Create a **System Image Type**.
4. Set the **ISO Image Type**.
5. **Create** a customization template.

Adding an ISO Datastore

The following procedure adds an ISO Datastore from your Red Hat Virtualization environment.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **ISO Datastores** accordion.
3. Click (**Configuration**), (**Add a new ISO Datastore**).
4. Select the Cloud or Infrastructure provider hosting the ISO Datastore.
5. Click **Add**.

The ISO datastore is added to Infrastructure Automation.

Refreshing an ISO Datastore

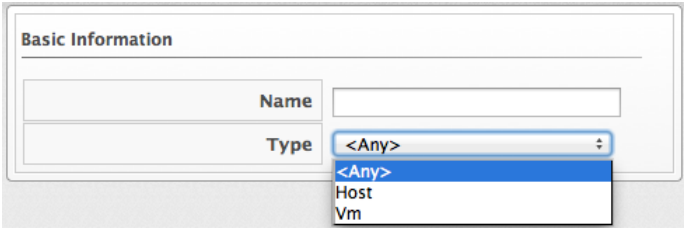
The following procedure refreshes the chosen ISO datastore and updates Infrastructure Automation with available ISOs.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **ISO Datastores** accordion, and select an ISO datastore.
3. Click (**Configuration**), then click (**Refresh Relationships**).

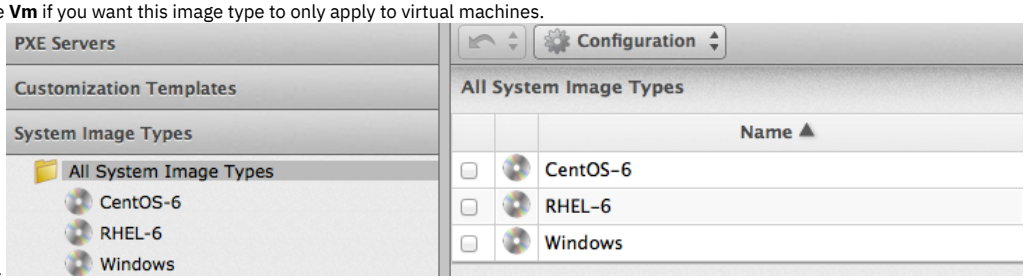
Creating System Image Types for ISO

The following procedure creates a system image type for ISO Servers.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **System Image Types** accordion.
3. Click (**Configuration**), then (**Add a new System Image Type**).
4. In **Basic Information**, type in a **Name** and select a **Type**.



   - Use **Vm** if you want this image type to only apply to virtual machines.
5. Click **Add**.



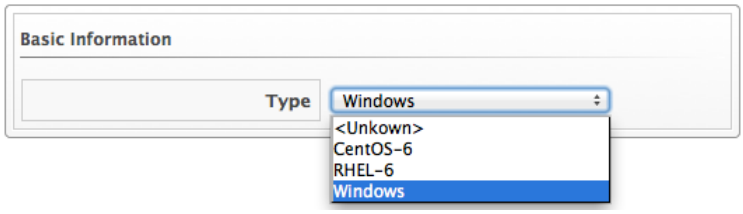
Example:

After creating the system image types, assign the types to each image on your ISO servers. To do this, you will select each image on the ISO server and identify its type.

Setting the Image Type for an ISO Image

The following procedure sets the image type for an ISO image.

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **ISO Datastores** accordion, and select the image that you want to set a type for.
3. Click (**Configuration**), then (**Edit this ISO Image**).



4. From the **Basic Information** area, select the correct **Type**.
5. Click **Save**.

Customization Templates for Virtual Machine and Instance Provisioning

Add a customization template to provide **Kickstart**, **Cloud-Init**, or **Sysprep** files for the initial loading of the operating system.

- When creating a template using Red Hat Virtualization, install the **cloud-init** package on the source virtual machine. This enables Cloud-Init to source configuration scripts when a virtual machine built on that template boots.
- See Using Cloud-Init to Automate the Configuration of Virtual Machines in the Red Hat Virtualization *Administration Guide* for more information on using Cloud-Init in a Red Hat Virtualization environment.
- See the Cloud-Init Documentation web site for example scripts.
- The **Kickstart** file must be named **ks.cfg**.
- Set the new virtual machine to power down after provisioning is complete.
- Infrastructure Automation must use the virtual machine payload feature of Red Hat Virtualization to create a floppy disk containing the data from the selected customization template.
- Customize the installer to include the data written to the floppy disk payload.
- RHEL 7.5 and above
- **`isolinux.cfg`** NDASH add **ks=cdrom** to the **append** line
- **`ks.cfg`** NDASH which must minimally include:

```
### Pre Install Scripts
%pre

# Mount the floppy drive
modprobe floppy
mkdir /tmp/floppy
mount /dev/fd0 /tmp/floppy
%end

# Include ks.cfg file from the floppy (written by CFME based on selected customization template)
%include /tmp/floppy/ks.cfg
```

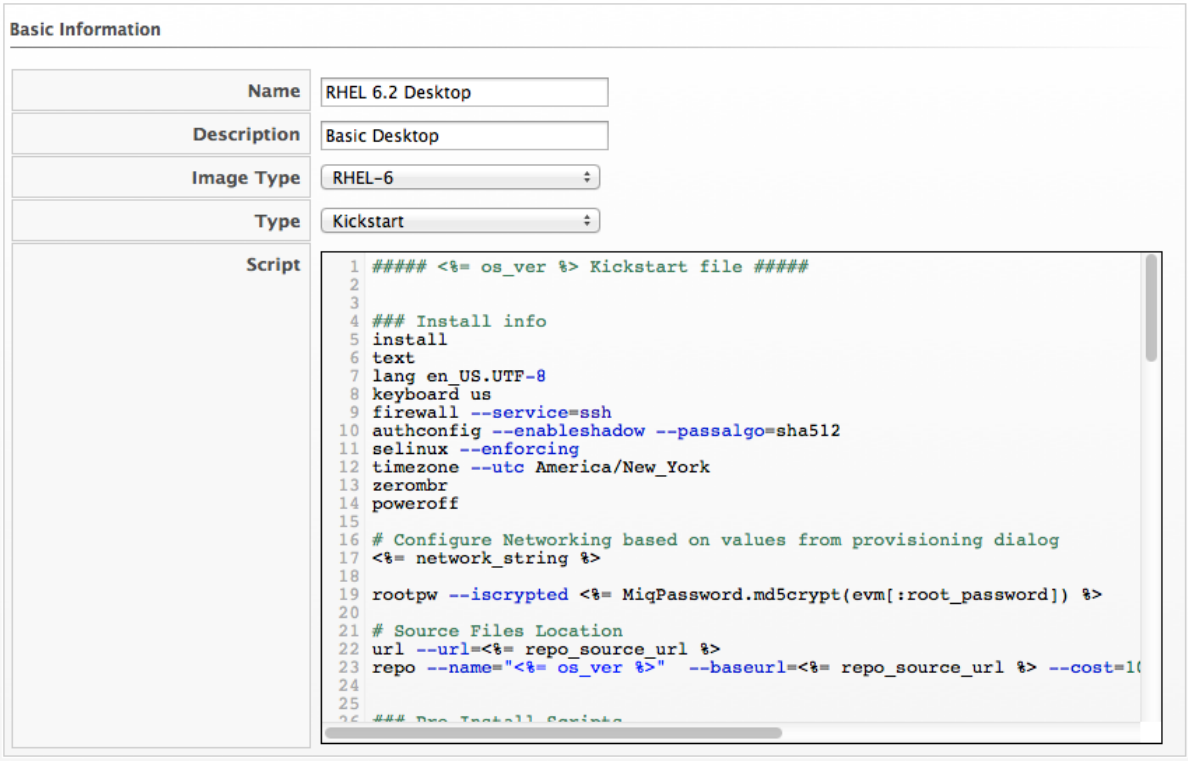
Customization Script Additions for Virtual Machine and Instance Provisioning

Customization Type	Reason to Include	Script entries		
Kickstart	Takes the values from the **Customize** tab in **Provisioning Dialog** and substitutes them into the script.	*Configure Networking based on values from provisioning dialog <% if evm[:addr_mode].first == 'static' %> <% network_string = "network –onboot yes –device=eth0 –bootproto=static –noipv6" %> <% ["ip", :ip_addr, "netmask", :subnet_mask, "gateway", :gateway, "hostname", :hostname, "nameserver", :dns_servers].each_slice(2) do	ks_key, evm_key	%> <% network_string << " –*{ks_key} # {evm[evm_key]}" unless evm[evm_key].blank ? %> <% end %> <%= network_string %> <% else %> network –device=eth0 –bootproto=dhcp <% end %>
Kickstart	Encrypts the root password from the **Customize** tab in the **Provisioning Dialog**.	rootpw –iscrypted <%= ManageIQ::Password.md5crypt(evm[:root_password]) %>		
Kickstart	Sends status of the provision back to Infrastructure Automation Server for display in the Infrastructure Automation Console.			

Customization Type	Reason to Include	Script entries		
Sysprep	Encrypts the root password from the **Customize** tab in the **Provisioning Dialog**. The value for the **AdministratorPassword** line must be inserted to use the password from the **Provision Dialog** and encrypt it.	<UserAccounts> <AdministratorPassword> <Value><%= ManageIQ::Password.sysprep_crypt(evm[:root_password]) %> </Value> <PlainText>false</PlainText> </AdministratorPassword> </UserAccounts>		

Adding a Customization Template

1. Navigate to **Compute > Infrastructure > PXE**.
2. Click the **Customization Templates** accordion.
3. Click (**Configuration**), (**Add a new Customization Template**).
4. In **Basic Information**, type in a **Name** and **Description**.



5. Select the **Image Type**. This list should include the PXE image types you created.
6. In **Type**, select **Kickstart** or **CloudInit** for Linux based systems, and **Sysprep** for Windows based system.
7. In the **Script** area, either paste the script from another source or type the script directly into the Infrastructure Automation interface.
8. Click **Add**.

Example:

The default dialogs show all possible parameters for provisioning. To limit the options shown, see Customizing Provisioning Dialogs.

Provisioning Virtual Machines

There are four types of provisioning requests available in Infrastructure Automation:

1. Provision a new virtual machine from a template
2. Clone a virtual machine
3. Publish a virtual machine to a template
4. Provision a virtual machine using cloud-init via REST API.

Provisioning a Virtual Machine from a Template

You can provision virtual machines through various methods. One method is to provision a virtual machine directly from a template stored on a provider.

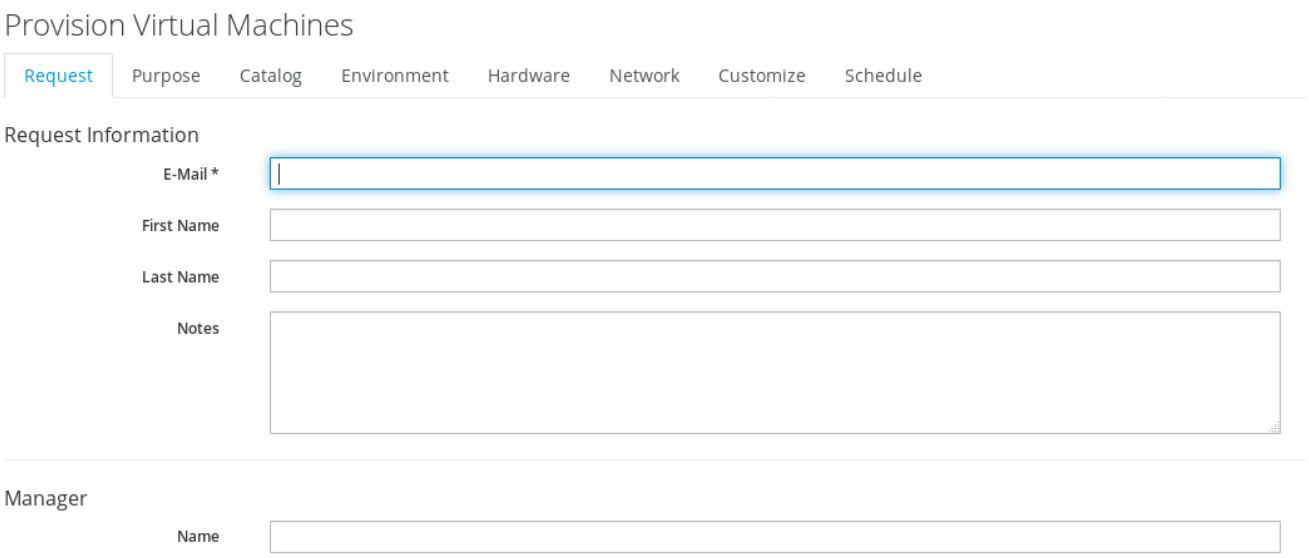
Important:

- To provision a virtual machine, you must have the “Automation Engine” role enabled.

- During virtual machine provisioning, the **Customize** tab is hidden if the template has an unknown operating system (OS) type. To make the **Customize** tab visible in the user interface, you will need to set the OS type from the provider or perform SmartState analysis on the template to detect the OS type.

To provision a virtual machine from a template:

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Click (**Lifecycle**), and then (**Provision VMs**).
3. Select a template from the list.
4. Click **Continue**.
5. On the **Request** tab, enter the request information.



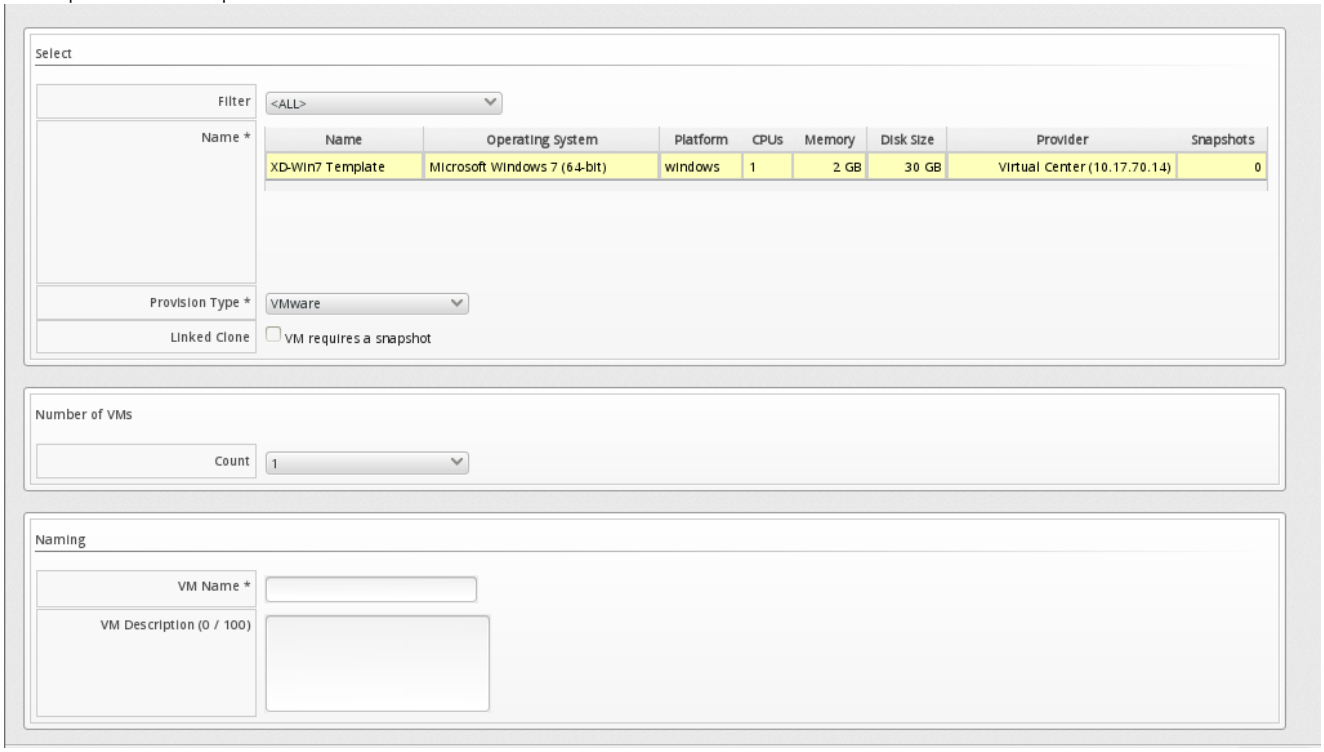
In **Request Information**, enter your name and email address. The requester will receive status emails about the provisioning request such as auto-approval, quota, provision complete, retirement, request pending approval, and request denied.

Note:

Parameters with a * next to the label are required to submit the provisioning request. To change the required parameters, see Customizing Provisioning Dialogs.

6. Click the **Purpose** tab to select the appropriate tags for the provisioned virtual machines.
7. Click the **Catalog** tab to select the template to provision from. This tab is context sensitive based on provider.

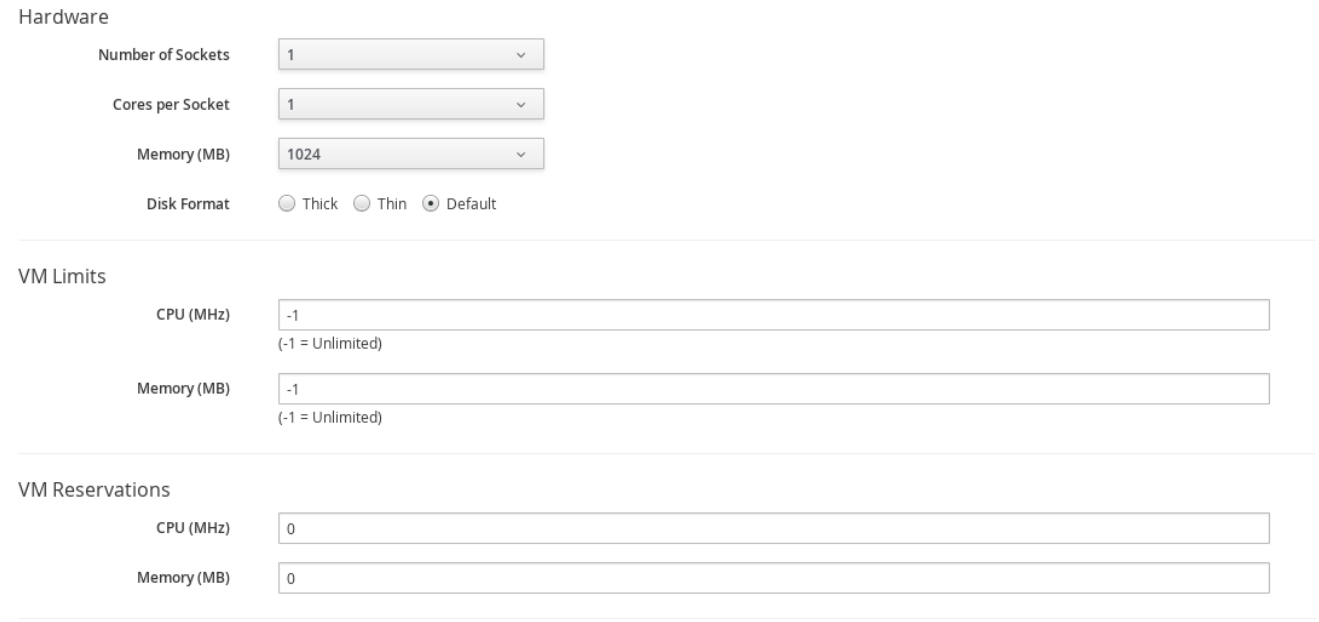
8. For templates on VMware providers:



    1. For **Provision Type**, select **VMware** or **PXE**.
        1. If **VMware** is selected, select **Linked Clone** to create a linked clone to the virtual machine instead of a full clone. Since a snapshot is required to create a linked clone, this box is only enabled if a snapshot is present. Select the snapshot you want to use for the linked clone.
        2. If **PXE** is selected, select a PXE **Server** and **Image** to use for provisioning.
    2. Under **Count**, select the number of virtual machines to create in this request.
    3. Use **Naming** to specify a virtual machine name and virtual machine description. When provisioning multiple virtual machines, a number will be appended to the virtual machine name.
9. For templates on Red Hat providers:
    1. Select the **Name** of a template to use.
    2. For **Provision Type**, select either **ISO**, **PXE**, or **Native Clone**. You must select **Native Clone** in order to use a Cloud-Init template.
        1. If **Native Clone** is selected, select **Linked Clone** to create a linked clone to the virtual machine instead of a full clone. This is equivalent to *Thin Template Provisioning* in Red Hat Virtualization. Since a snapshot is required to create a linked clone, this box is only enabled if a snapshot is present. Select the snapshot to use for the linked clone.
        2. If **ISO** is selected, select an ISO **Image** to use for provisioning.
        3. If **PXE** is selected, select a PXE **Server** and **Image** to use for provisioning.
    3. Under **Count**, select the number of virtual machines you want to create in this request.
    4. Use **Naming** to specify a **VM Name** and **VM Description**. When provisioning multiple virtual machines, a number will be appended to the **VM Name**.
10. Click the **Environment** tab to decide where you want the new virtual machines to reside.
    1. If provisioning from a template on VMware, you can either let Infrastructure Automation decide for you by checking **Choose Automatically**, or select a specific cluster, resource pool, folder, host, and datastore. VMware virtual machines can also be provisioned to a clustered datastore by selecting it under **Datastore**. Additionally, you can assign a storage profile to a VMware virtual machine under **Datastore** to configure the virtual machine to operate using a storage profile from that datastore.

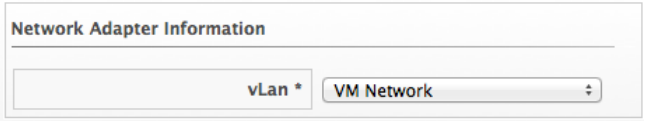
       Note, read-only datastores are excluded when provisioning a virtual machine.
    2. If provisioning from a template on Red Hat, you can either let Infrastructure Automation decide for you by checking **Choose Automatically**, or select a datacenter, cluster, host and datastore.

11. Click the **Hardware** tab to set hardware options.



    1. In **Hardware**, set the number of sockets, cores per socket, memory in MB, and disk format: thin, pre-allocated/thick or same as the provisioning template (default).
    2. For VMware provisioning, set the **VM Limits** of CPU and memory the virtual machine can use.
    3. For VMware provisioning, set the **VM Reservation** amount of CPU and memory.

12. Click **Network** to set the vLan adapter. Additional networking settings that are internal to the operating system appear on the **Customize** tab.

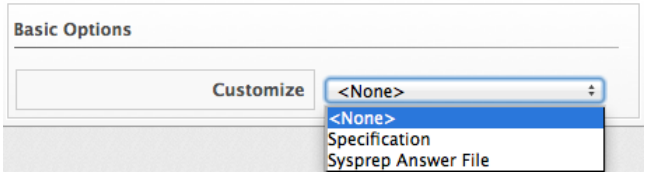


    1. In **Network Adapter Information**, select the **vLan**.

       Note:

       A VMware virtual machine can be provisioned to a DVPortgroup by selecting it from the **vLan** list. Prior to provisioning a virtual machine, the DVPortgroup must be created on a vSphere Distributed Switch (VDS) in VMware vCenter in order for Infrastructure Automation to list the DVPortgroup under **vLan**.

13. Click **Customize** to customize the operating system of the new virtual machine. These options vary based on the operating system of the template.

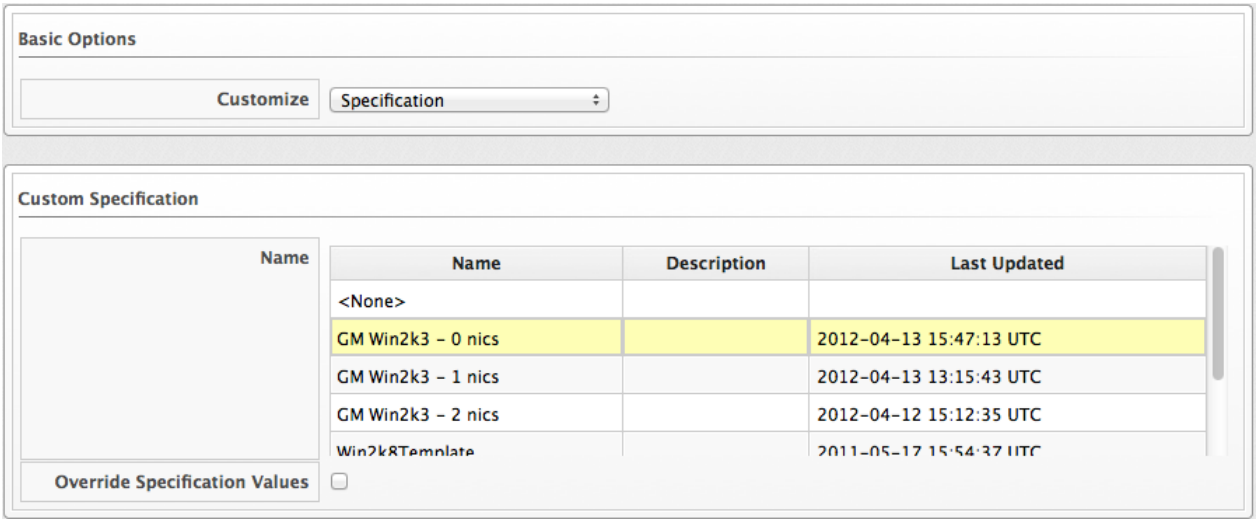


14. For Windows provisioning:

    1. To use a custom specification from the provider, click **Specification**. To select an appropriate template, choose from the list in the custom specification area. The values that are honored by Infrastructure Automation display.

       Note:

       Any values in the specification that do not show in the Infrastructure Automation console's request dialogs are not used by Infrastructure Automation. For example, for Windows operating systems, if you have any run once values in the specification, they are not used in creating the new virtual machines. Currently, for a Windows operating system, Infrastructure Automation honors the unattended GUI, identification, workgroup information, user data, windows options, and server license. If more than one network card is specified, only the first is used.



To modify the specification, select **Override Specification Values**.

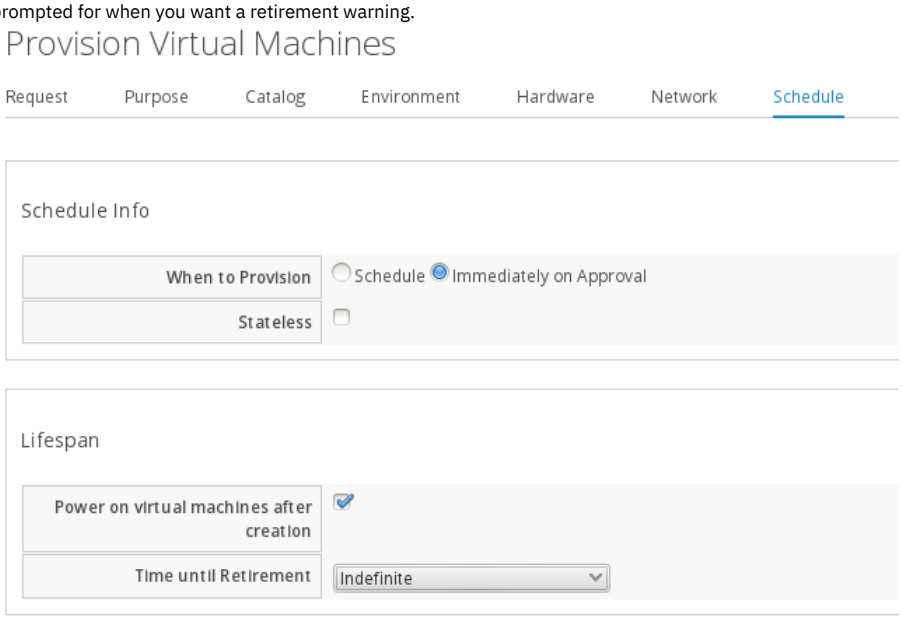
2. Select **Sysprep Answer File**, to upload a Sysprep file or use one that exists for a custom specification on the Provider where the template resides. To upload a file, click **Browse** to find the file, and then upload. To use an answer file in **Customization Specification**, click on the item. The answer file will automatically upload for viewing. You cannot make modifications to it.

15. For Linux provisioning:

    1. Under **Credentials**, enter a **Root Password** for the **root** user to access the instance.
    2. Enter a **IP Address Information** for the instance. Leave as **DHCP** for automatic IP assignment from the provider.
    3. Enter any **DNS** information for the instance if necessary.
    4. Select **Customize Template** for additional instance configuration. Select from the Kickstart or Cloud-Init customization templates stored on your appliance.

16. Click the **Schedule** tab to select when provisioning begins.

    1. In **Schedule Info**, select when to start provisioning. If you select **Schedule**, you will be prompted to enter a date and time. Select **Stateless** if you do not want the files deleted after the provision completes. A stateless provision does not write to the disk so it requires the PXE files on the next boot.
    2. In **Lifespan**, select to power on the virtual machines after they are created, and to set a retirement date. If you select a retirement period, you will be prompted for when you want a retirement warning.



17. Click **Submit**.

The provisioning request is sent for approval. For the provisioning to begin, a user with the administrator, approver, or super administrator account role must approve the request. The administrator and super administrator roles can also edit, delete, and deny the requests. You will be able to see all provisioning requests where you are either the requester or the approver.

After submission, the appliance assigns each provision request a **Request ID**. If an error occurs during the approval or provisioning process, use this ID to locate the request in the appliance logs. The Request ID consists of the region associated with the request followed by the request number. As regions define a range of one

trillion database IDs, this number can be several digits long.

Request ID Format

Request 99 in region 123 results in Request ID 123000000000099.

Provisioning a Virtual Machine using Cloud-Init via REST API

Cloud-init is a tool for automating the initial setup of virtual machines. In Infrastructure Automation, you can use cloud-init via REST API to provision a virtual machine that was created based on a template.

Note:

To use cloud-init, the template from which the virtual machine is provisioned must have cloud-init package installed, and have the **Use Cloud-Init/Sysprep** option selected.

For a virtual machine provision request via REST API, ensure the following two fields in the request's body are set correctly, otherwise cloud-init may not work.

- VLAN
- sysprep_enabled

VLAN.

The value of **VLAN** in the API request can be one of the following options:

VLAN value	Note	vNIC profile
<empty>	Must be written within the <..>	No vNIC profile
<Template>	Must be written within the <..>	Use the template's default vNIC profile
profile_name (network_name)	Must include a space between profile_name (network_name), otherwise it will fail. For example, **ovirtmgmt (ovirtmgmt)**	Set the specified vNIC profile.
vNIC profile ID		Set the specified vNIC profile, such as `3a8dce01-dd59-4a46-8a6f-823acccda79f`.

sysprep_enabled.

The value of sysprep_enabled in the API request must be in the following format.

Note:

Sysprep_enabled is written in the order: [value] - [it's meaning] - [how it appears in the Infrastructure Automation user interface virtual machine provision dialog's **Customize** tab]

For Windows template:

- **"fields"** - Sysprep Specification In the Infrastructure Automation user interface, when you navigate to the virtual machine provision dialog (**Compute > Infrastructure

  Virtual Machine**), this option located under the **Customize** tab's **Customize** drop-down list is called **Sysprep Specification**.
- **"file"** - Sysprep answer file in the Infrastructure Automation user interface, when you navigate to the virtual machine provision dialog (**Compute > Infrastructure > Virtual Machine**), this option located under the **Customize** tab's **Customize** drop-down list is called **Sysprep answer file**.

For Linux template:

- **"fields" -** Customized template and any Customized parameters will be used In the Infrastructure Automation user interface, when you navigate to the virtual machine provision dialog (**Compute > Infrastructure

  Virtual Machine**), this option located under the **Customize** tab's **Customize** drop-down list is called **Specification**.

Note:

For cloud-init to work *(that is, to have the provisioned virtual machine marked with **Use cloud-init** and the customized template as well as customized parameters, if any, will be used)* "sysprep_enabled" must be set to "fields". If you do not set it correctly, the customized template will be ignored and the provisioned virtual machine will not be marked with **Use cloud-init**; although, the template from which the virtual machine is provisioned has it marked.

For both Windows and Linux template:

- **"disabled"** - Do not customize In the Infrastructure Automation user interface, when you navigate to the virtual machine provision dialog (**Compute > Infrastructure > Virtual Machine**), this option located under the **Customize** tab's **Customize** drop-down list is called **<None>**. The customized template will be ignored, and the provisioned virtual machine will not be marked with **Use cloud-init**, even though the template from which the virtual machine was provisioned, has it marked. The default value is 'disabled'.

Note:

For an example of virtual machine provisioning request using cloud-init via REST API, see *Provision Requests* in the Infrastructure Automation API Guide.

Cloning a Virtual Machine

Virtual machines can be cloned in other providers as well.

1. Navigate to **Compute > Infrastructure > Virtual Machines**, and select the virtual machine you want to clone.

2. Click (**Lifecycle**), and then (**Clone selected item**).
3. Enter the requested information in the dialogs. Be sure to check the **Catalog** tab.
4. Schedule the request on the **Schedule** tab.
5. Click **Submit**.

Publishing a Virtual Machine to a Template (VMware Virtual Machines Only)

1. Navigate to **Compute > Infrastructure > Virtual Machines**, and select the virtual machine you want to publish as a template.
2. Click (**Lifecycle**), and then (**Publish selected VM to a Template**).
3. Enter the requested information in the dialogs. Be sure to check the **Catalog** tab.
4. Schedule the request on the **Schedule** tab.
5. Click **Submit**.

Renaming a Provisioned Virtual Machine (VMware Virtual Machines Only)

Infrastructure Automation allows you to rename a VMware virtual machine without having to reprovision it.

To rename a VMware virtual machine:

1. Navigate to **Compute > Infrastructure > Virtual Machines**, and select the VMware virtual machine you want to rename.
2. Click (**Configuration**), (**Rename selected item**).
3. In the **Basic Information** screen, provide a new **name**.
4. Click **Save**.

The renamed virtual machine will appear in the inventory view.

Provisioning Instances

Cloud instances follow the same process (Request, Approval, Deployment) as a standard virtual machine from virtualization infrastructure. First, a user makes a request for instances and specifies the image, volume or volume snapshot, tags, availability zone and hardware profile flavor. Second, the request goes through the approval phase. Finally, Infrastructure Automation executes the request.

Provisioning an EC2 Instance from an Image

1. Navigate to **Compute > Clouds > Instances**.
2. Click (**Lifecycle**), then click (**Provision Instances**).
3. Select an image from the list presented.
4. Click **Continue**.
5. On the **Request** tab, enter information about this provisioning request. In **Request Information**, type in at least a first and last name and an email address. This email is used to send the requester status emails during the provisioning process for items such as auto-approval, quota, provision complete, retirement, request pending approval, and request denied. The other information is optional. If the Infrastructure Automation Server is configured to use LDAP, you can use the **Look Up** button to populate the other fields based on the email address.

   Note:

   Parameters with a * next to the label are required to submit the provisioning request. To change the required parameters, see Customizing Provisioning Dialogs.
6. Click the **Purpose** tab to select the appropriate tags for the provisioned instance.
7. Click the **Catalog** tab for basic instance options.
   1. To change the image to use as a basis for the instance, select it from the list of images.
   2. Select the **Number of VMs** to provision.
   3. Type a **VM Name** and **VM Description**.
8. Click the **Environment** tab to select the instance's **Availability Zone**, **Virtual Private Cloud**, **Cloud Subnet**, **Security Groups**, and **Elastic IP Address**. If no specific availability zone is required, select the **Choose Automatically** checkbox.
9. Click the **Properties** tab to set provider options such as hardware flavor and security settings.
   1. Select a flavor from the **Instance Type** list.
   2. Select a **Guest Access Key Pair** for access to the instance.
   3. Select the **CloudWatch** monitoring level. Leave as **Basic** for the default EC2 monitoring.

10. Click the **Customize** tab to set additional instance options.
    1. Under **Credentials**, enter a **Root Password** for the **root** user access to the instance.
    2. Enter a **IP Address Information** for the instance. Leave as **DHCP** for automatic IP assignment from the provider.
    3. Enter any **DNS** information for the instance if necessary.
    4. Select a **Customize Template** for additional instance configuration. Select from the Cloud-Init scripts stored on your appliance.
11. Click the **Schedule** tab to set the provisioning and retirement date and time.
    1. In **Schedule Info**, choose whether the provisioning begins upon approval, or at a specific time. If you select **Schedule**, you will be prompted to enter a date and time.
    2. In **Lifespan**, select whether to power on the instances after they are created, and whether to set a retirement date. If you select a retirement period, you will be prompted for when to receive a retirement warning.
12. Click **Submit**.

The provisioning request is sent for approval. For the provisioning to begin, a user with the admin, approver, or super admin account role must approve the request. The admin and super admin roles can also edit, delete, and deny the requests. You will be able to see all provisioning requests where you are either the requester or the approver.

After submission, the appliance assigns each provision request a **Request ID**. If an error occurs during the approval or provisioning process, use this ID to locate the request in the appliance logs. The Request ID consists of the region associated with the request followed by the request number. As regions define a range of one trillion database IDs, this number can be several digits long.

Request ID Format

Request 99 in region 123 results in Request ID 123000000000099.

Provisioning an OpenStack Instance from an Image, Volume or Volume Snapshot

Create a request to provision Red Hat OpenStack Platform cloud instances from images, volumes, and volume snapshots using Infrastructure Automation. Only bootable volumes not in use will be available.

1. Browse to menu: **Compute > Clouds > Instances**
2. Click (**Lifecycle**), then click (**Provision Instances**).
3. Select an OpenStack image, volume or volume snapshot from the list presented. These files must be available on your OpenStack provider.
4. Click **Continue**.
5. On the **Request** tab, enter information about this provisioning request. In **Request Information**, type in at least an email address. This email is used to send the requester status emails during the provisioning process for items such as auto-approval, quota, provision complete, retirement, request pending approval, and request denied. The other information is optional. If the Infrastructure Automation Server is configured to use LDAP, you can use the **Look Up** button to populate the other fields based on the email address.

   Note:

   Parameters with a * next to the label are required to submit the provisioning request. To change the required parameters, see Customizing Provisioning Dialogs.
6. Click the **Purpose** tab to select the appropriate tags for the provisioned instance.
7. Click the **Catalog** tab for basic instance options.
    1. To change the source file to use as a basis for the instance, select it from the list of images, volumes, or volume snapshots.
    2. Select the **Number of Instances** to provision.
    3. Type a **Instance Name** and **Instance Description**.
8. Click the **Environment** tab to select the instance's **Cloud Tenant**, **Availabilty Zones**, **Cloud Network**, **Security Groups**, and **Public IP Address**. If no specific Cloud Tenant is required, select the **Choose Automatically** checkbox.
9. Click the **Properties** tab to set provider options such as flavors and security settings.
    1. Select a flavor from the **Instance Type** list.
    2. Select a **Guest Access Key Pair** for access to the instance. For more information about key pairs, see Managing Key Pairs.
10. Click the **Volumes** tab to provision any volumes with the instance. Volumes are useful for augmenting ephemeral storage of instances with persistent, general-purpose block storage:
    1. Fill in the **Volume Name** and **Size (gigabytes)** fields.
    2. If you want the volume to be deleted once the instance terminates (thereby making it non-persistent), check **Delete on Instance Terminate**.
    3. To provision and add multiple volumes to the instance, click **Add Volume**. Doing so will add new fields you can fill in.

       For more information about persistent storage in OpenStack, see the Red Hat OpenStack Platform *Storage Guide*.

11. Click the **Customize** tab to set additional instance options.
    1. Under **Credentials**, enter a **Root Password** for the **root** user access to the instance.
    2. Enter a **IP Address Information** for the instance. Leave as **DHCP** for automatic IP assignment from the provider.
    3. Enter any **DNS** information for the instance if necessary.
    4. Select a **Customize Template** for additional instance configuration. Select from the Cloud-Init scripts stored on your appliance.
12. Click the **Schedule** tab to set the provisioning and retirement date and time.
    1. In **Schedule Info**, choose whether the provisioning begins upon approval, or at a specific time. If you select **Schedule**, you will be prompted to enter a date and time.
    2. In **Lifespan**, select whether to power on the instances after they are created, and whether to set a retirement date. If you select a retirement period, you will be prompted for when to receive a retirement warning.
13. Click **Submit**.

The provisioning request is sent for approval. For the provisioning to begin, a user with the admin, approver, or super admin account role must approve the request. The admin and super admin roles can also edit, delete, and deny the requests. You will be able to see all provisioning requests where you are either the requester or the approver.

After submission, the appliance assigns each provision request a **Request ID**. If an error occurs during the approval or provisioning process, use this ID to locate the request in the appliance logs. The Request ID consists of the region associated with the request followed by the request number. As regions define a range of one trillion database IDs, this number can be several digits long.

Request ID Format

Request 99 in region 123 results in Request ID 123000000000099.

Provisioning an IBM Power Systems Virtual Servers Instance from an Image

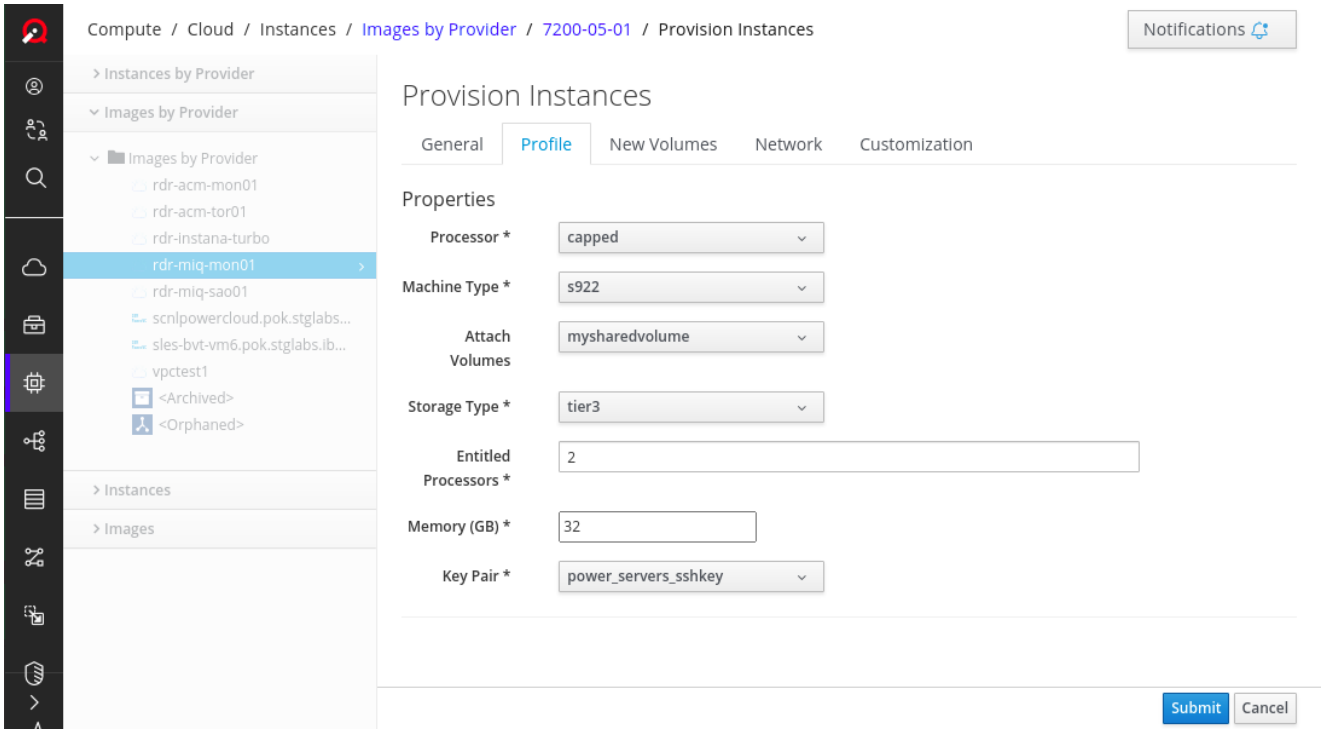
For an overview of IBM Power Systems Virtual Servers, including terminology used in Infrastructure Automation, see Getting started with IBM Power Systems Virtual Servers

Create a request to provision IBM Power Systems Virtual Servers instances from images.

1. Browse to menu: **Compute > Clouds > Instances**.
2. Click **Lifecycle**, then click (**Provision Instances**).
3. Select an image from the list presented. Each image is associated with a single Power Systems Virtual Servers provider instance. If there are multiple provider instances registered be sure to select an image associated with the desired instance.

   Notes:

   - Images must already be imported into a Power Systems Virtual Servers provider instance before becoming available in the Infrastructure Automation inventory. This can be done using the IBM Cloud web interface, CLI, or API. For detailed instructions see IBM Cloud PowerVS docs: Importing a boot image and IBM Cloud CLI: ibmcloud pi image-create.
   - SAP HANA on IBM Power Systems Virtual Servers requires special SLES for SAP HANA images. These boot images are added to a Power Systems Virtual Servers provider instance automatically when provisioning an SAP HANA instance through the IBM Cloud web interface. Once the SLES for SAP HANA image has been added, subsequent instances can be provisioned using Infrastructure Automation.
4. Click **Continue**.
5. Click the **General** tab for general instance options.
    1. Select the **Number of Instances** to provision.
    2. Type an **Instance Name**.
6. Click the **Profile** tab for hardware profile options.



1. Select the **Processor** type.

   Note: See Power Systems Virtual Servers documentation for more information.

2. Select a **Machine Type**.

   Note: When provisioning with an SAP HANA image, the **Machine Type** field is used to select the SAP profile. The SAP profile includes the number of entitled processors and memory (these fields will be automatically set and can only be changed by selected a different SAP profile).

3. Optionally, select **Attach Volumes** items.

   Note: A VM cannot have disks from different storage types. Only existing volumes matching the **Storage Type** will be available for selection.

4. Select a **Storage Type**.

   Note: See Power Systems Virtual Servers documentation for more information on volume types.

5. Enter number of **Entitled Processors**.

   Notes:

   - If 'dedicated' **Processor** is selected, **Entitled Processors** must be allocated in whole numbers.
   - If 'shared' or 'uncapped' **Processor** is selected, **Entitled Processors** must be allocated in a multiple of '0.25'.
   - When provisioning with an SAP HANA image this field is set by the profile (**Machine Type**) selection.

6. Enter **Memory (GB)**.

   Note: When provisioning with an SAP HANA image this field is set by the profile (**Machine Type**) selection.

7. Optionally, select a **Key Pair**.

7. Click the **New Volumes** tab to optionally create new volumes to be attached to the VM.

   All new volumes will be the **Storage Type** selected in the **Profile** tab. For each new volume:

   1. Type a volume **Name**.
   2. Enter **Size (GB)**.
   3. Select a **Sharable**.

8. Click the **Network** tab to select networks.

   1. Select an existing network from **Network Interface**. If 'None' is selected then **Attach to New Public Network** must be selected.
   2. Select **Attach to New Public Network** to create a new public network to attach to the VM.

9. Click the **Customization** tab for additional options.

   1. Use the **Upload File** section to upload a Cloud-init file.

2. In the **Other** section, select **Migratable** and **VM pinning**

   Note: Learn more about VM pinning.

3. In the **IP Address Information** section, optionally enter an IP address to request for the selected **Network Interface** in the **Network** tab.

10. Click **Submit**.

The provisioning request is sent for approval. For the provisioning to begin, a user with the admin, approver, or super admin account role must approve the request. The admin and super admin roles can also edit, delete, and deny the requests. You will be able to see all provisioning requests where you are either the requester or the approver.

After submission, the appliance assigns each provision request a **Request ID**. If an error occurs during the approval or provisioning process, use this ID to locate the request in the appliance logs. The Request ID consists of the region associated with the request followed by the request number. As regions define a range of one trillion database IDs, this number can be several digits long.

Request ID Format

Request 99 in region 123 results in Request ID 123000000000099.

Customizing Provisioning Dialogs

The default set of provisioning dialogs shows all possible options. However, Infrastructure Automation also provides the ability to customize which tabs and fields are shown. You can decide what fields are required to submit the provisioning request or set default values.

For each type of provisioning, there is a dialog that can be created to adjust what options are presented. While samples are provided containing all possible fields for provisioning, you can remove what fields are shown but cannot add new fields or tabs.

Edit the dialogs to:

1. Hide or show provisioning tabs.
2. Hide or show fields. If you hide an attribute, the default will be used, unless you specify otherwise.
3. Set default values for a field.
4. Specify if a field is required to submit the request.
5. Create custom dialogs for specific users.

Adding a Provision Dialog for All Users

1. Login to the Infrastructure Automation console for the Infrastructure Automation server where you want to change the dialog.
2. Navigate to **Automation > Embedded Automate > Customization**.
3. Click the **Provisioning Dialogs** accordion.
4. Click the type of dialog you want to create: **Host Provision**, **VM Provision** or **VM Migrate**.
5. Select one of the default dialogs.
6. Click **Configuration**, and then click **(Copy this Dialog)**.
7. Type a new **Name** and **Description** for the dialog.
8. In the **Content** field,
   - To remove a tab from display, change its display value to ignore. By choosing ignore, you not only hide the tab, but also skip any fields on that tab that were required. To show the tab, change the display value to show.
   - To hide a field, change its `:display:` value from `:edit` to `:hide`. To display fields of most data types, use `:edit`. To display a button, use `:show`. To set a default value for a field, add `:default: defaultvalue` to the list of parameters for the field. Set the `:required:` parameter to either `true` or `false` based on your needs.

     Note:

     If you set `:required:` to `true`, the field must have a value for the provision request to be submitted.
9. Click **Add**.

If you are using **Provisioning Profiles**, you can specify a specific file that holds the customizations. To do this, you must create an instance mapping to this file in the Infrastructure Automation **Applications/provisioning/profile/VM provisioning by group** class. By default, if you are using provisioning profiles and the group does not have a defined instance, the appropriate default dialog file will be used based on the type of provisioning selected.

Creating a Custom Provision Dialog

1. Navigate to **Automation > Embedded Automate > Customization**.
2. Click on the **Provisioning Dialogs** accordion.
3. Click on the type of dialog you want to create, **Host Provision**, **VM Provision** or **VM Migrate**.
4. Select one of the default dialogs.

5. Click (**Configuration**), and then (**Copy this Dialog**).
6. Rename the dialog as shown in the examples below.

Type of Provision	Dialog Name
Provision Virtual Machine from a template	miq_provision_dialogs_groupname_template Example: miq_provision_dialogs_ EvmGroup-user_self_service _template
Clone a Virtual Machine	miq_provision_dialogs_groupname_clone_to_vm Example: miq_provision_dialogs_ EvmGroup-user_self_service _clone_to_vm
Publish a Virtual Machine to a template	miq_provision_dialogs_groupname_clone_to_template Example: miq_provision_dialogs_ EvmGroup-user_self_service _clone_to_template

7. Make any changes you need.
8. In the **Content** field,
   - To remove a tab from display, change its display value to ignore. By choosing ignore, you not only hide the tab, but also skip any fields on that tab that were required. To show the tab, change the display value to show.
   - To hide a field, change its **:display:** value from **:edit** to **:hide**. To ensure the field does not get turned back on by a workflow model, use **:display_override: :hide**. To display fields of most data types, use **:edit**. To display a button, use **:show**. To set a default value for a field, add **:default:**
     defaultvalue to the list of parameters for the field. Set the **:required:** parameter to either **true** or **false** based on your needs.

     Note:

     If you set **:required:** to **true**, the field must have a value for the provision request to be submitted.
9. Click **Add**.

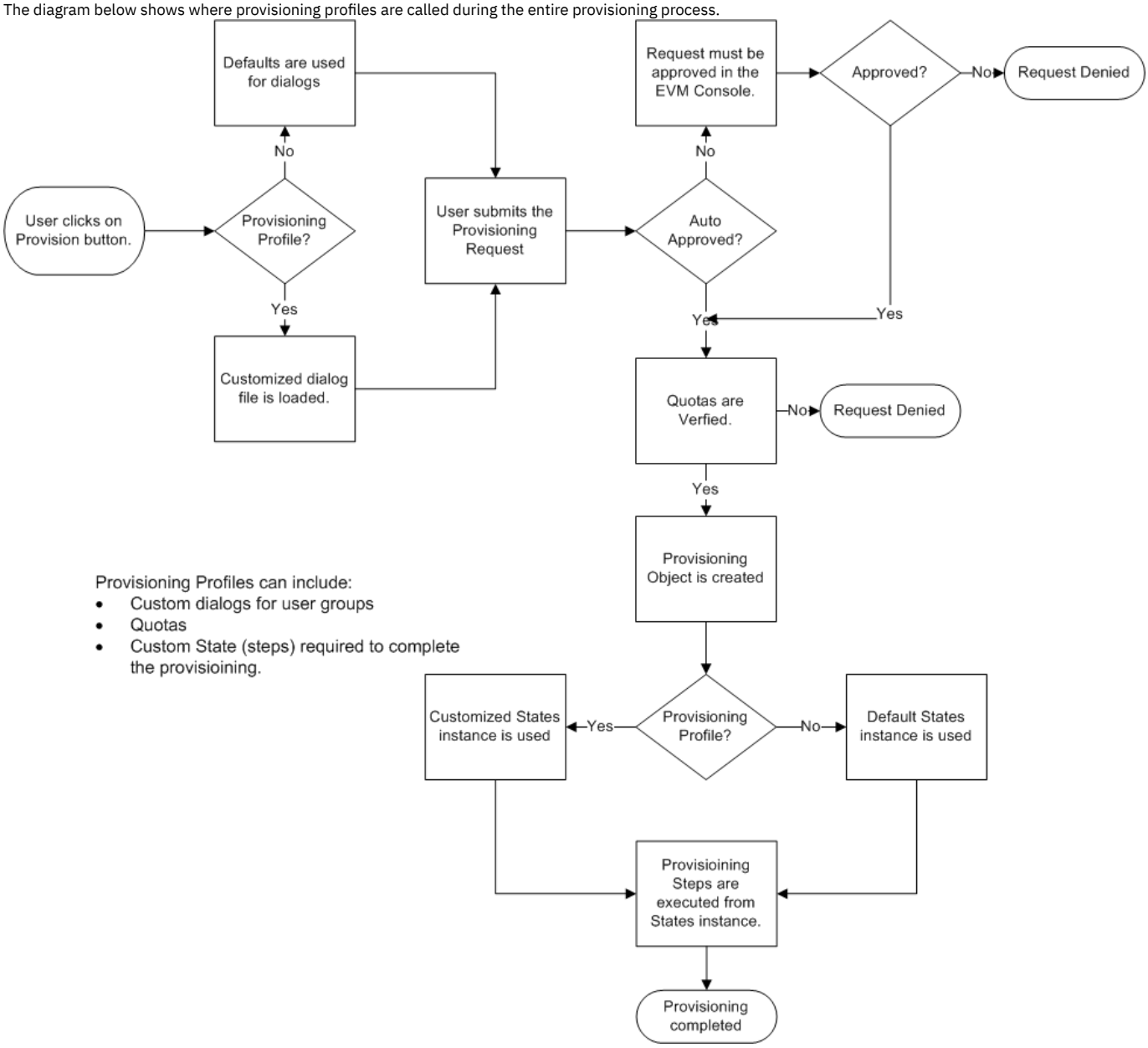
Enter the name of the new dialog into the dialog name field in the appropriate Infrastructure Automation **Applications/provisioning/profile instance**. This dialog can now be referred to in an instance in the Provisioning Profiles class so that it can be used for groups of users.

Provisioning Profiles

Provisioning profiles can be used to customize the dialogs and the state machine (steps used to provision the machine). Profiles can be created for LDAP or Infrastructure Automation groups. To use provisioning profiles:

- Create a **Provisioning Profile** instance for the LDAP or Infrastructure Automation group. If no instance exists, then default settings will be used.
- If customizing dialogs, create a custom dialog file, and specify the name of that file in the provisioning profile instance. If customizing the states for provisioning, create a state instance and set the name of the state instance in the provisioning profile instance.

The diagram below shows where provisioning profiles are called during the entire provisioning process.



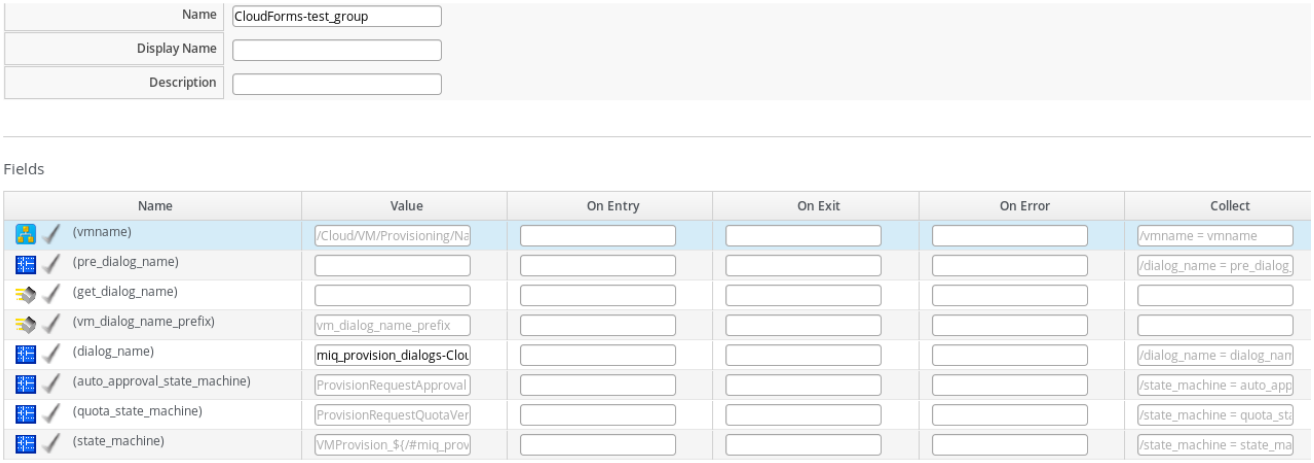
Creating a Provisioning Profile Instance

1. Navigate to **Automation > Embedded Automate > Explorer**.
2. Using the tree located in the accordion, click **DOMAIN > Cloud > VM > Provisioning > Profile**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.

   This example uses the **Cloud** Namespace, but can also use the **Infrastructure** namespace.
3. Click (**Configuration**), (**Add a New Instance**).
4. Make the name of the tag identical to the name of the LDAP or Infrastructure Automation group you are creating the instance for, replacing spaces in the group name with underscores. For example, change **Infrastructure Automation-test group** to **Infrastructure Automation-test_group**.



5. In the dialog name field, enter the name of the customized dialog file. This file must reside on the Infrastructure Automation appliance in the **`/var/www/miq/vmdb/db/fixtures`** directory. Red Hat recommends naming the file in the format **`miq_provision_dialogs-groupname.rb`** and copying this file to all Infrastructure Automation appliances. For instructions on creating a custom dialog file, see Customizing Provisioning Dialogs.

   Note:

   Be sure that the custom dialog file exists. If it does not, an error will appear when the user clicks on the **Provisioning** button in the Infrastructure Automation console.

6. Click **Add**.

Setting Provisioning Scope Tags

Some non-default placement methods, for example the **redhat_best_placement_with_scope** or **vmware_best_fit_with_scope** methods, may require you to set **Provisioning Scope** tags for a host and a datastore.

To enable these resources for all groups, set the scope to **All**. To limit access to a select group, create a tag in the **Provisioning Scope** category with the exact name of the user group and set this tag on the desired resources. See Tags in *General Configuration* for information on creating tags.

To set the scope for a datastore:

1. Navigate to **Compute > Infrastructure > Datastores**.
2. Select the datastore to set the provisioning scope for.
3. Click (**Policy**), and then (**Edit Tags**).
4. From the **Select a customer tag to assign** drop down, select **Provisioning Scope** and then a value for the tag from the next drop down menu.
5. Click **Save**.

Managing Key Pairs

Key pairs allow you to manage SSH access between a user and provisioned instance. For more information about key pairs in OpenStack, see Manage Key Pairs in the *Instances and Images Guide*.

To manage key pairs, browse to menu: **Compute > Clouds > Key Pairs**. From there, you can view a list of available key pairs. Click on a key pair to view its details.

To create a new key pair:

1. Browse to menu: **Compute > Clouds > Key Pairs**.
2. Click **Configuration**, (**Add a new Key Pair**).
3. Enter a **Name** for the key pair.
4. If you want to use a public key, copy its contents into the **Public Key (optional)** field.

   Note: You must have a public key value for the IBM POWER VS provider to create the key pair. An error message is returned if you fail to add a public key value and can be viewed in **Settings > Tasks**.

5. Select which cloud provider on which to create the key pair. The key pair will then be available for use by instances in that provider.
6. Click **Add**.

Working with Requests

Provisioning Request Approval Methods

In this chapter, you will learn about the different approval methods. The request can be approved manually in the Infrastructure Automation console, set for automatic approval by setting options in the **Automate Explorer**, or by using an external method.



When using an external method, the approval actually takes place on the external system and is sent directly for execution. This chapter discusses how to view and edit requests in the Infrastructure Automation Console, how to approve a request, and how to set automatic approval parameters.

Working with Provisioning Requests

After a provisioning request is sent, if you have proper authority, you can copy, edit, delete, approve, or deny a request.

After submission, the appliance assigns each provision request a **Request ID**. If an error occurs during the approval or provisioning process, use this ID to locate the request in the appliance logs. The **Request ID** consists of the region associated with the request followed by the request number. As regions define a range of one trillion database IDs, this number can be several digits long.

Request ID Format

Request 99 in region 123 results in **Request ID 123000000000099**.

Reloading the Status of Provisioning Requests

1. Navigate to **Services > Requests**.
2. Click (**Refreshes the current display**).

Approving a Provisioning Request

After a user creates provisioning request, administrators have the ability to approve the request and allow Infrastructure Automation to complete virtual machine or instance creation.

1. Navigate to **Services > Requests**.
2. Click on the request you want to approve.
3. Click (**Approve this request**).
4. Type in a **Reason** for the approval.

Denying a Provisioning Request

1. Navigate to **Services > Requests**.
2. Click on the request you want to deny.
3. Click (**Deny this request**).
4. Type in a **Reason** for the denial.

Copying a Provisioning Request

1. Navigate to **Services > Requests**.
2. Click on the request you want to copy.
3. Click (**Copy original provision request**).
4. Make changes to the request.
5. Click **Submit**.

If the logged in user is not same as the requester or the request has been already approved or denied, you cannot edit or delete the request.

Editing a Provisioning Request

1. Navigate to **Services > Requests**.
2. Click on the request you want to edit.
3. Click (**Edit the original provision request**).
4. Make changes to the request.
5. Click **Submit**.

Deleting a Provisioning Request

1. Navigate to **Services > Requests**.
2. Click on the request you want to delete.
3. Click (**Delete this request**).
4. Click **OK** to confirm.

Automatically Approving Requests

You can set thresholds for automatic approval of provisioning requests and, therefore, remove the requirement to manually approve the request. You can do this either as a global default or on a per template basis.

Enabling Global Defaults for Automatic Approval

To enable a global set of default approval values, edit the defaults instance by navigating to **Automation > Embedded Automate > Explorer**, then **DOMAIN > Cloud|Infrastructure > VM > Provisioning > StateMachines > ProvisionRequestApproval** in the accordion menu. The parameters in this instance are used by the methods in that same class. By default, the maximum number of virtual machines or instances that can be automatically approved for provisioning is 1. To skip the check for the maximum number of virtual machines, set this field to 0. Set this field to -1 to force manual approval. At a minimum, you must change this parameter for all others to be validated.

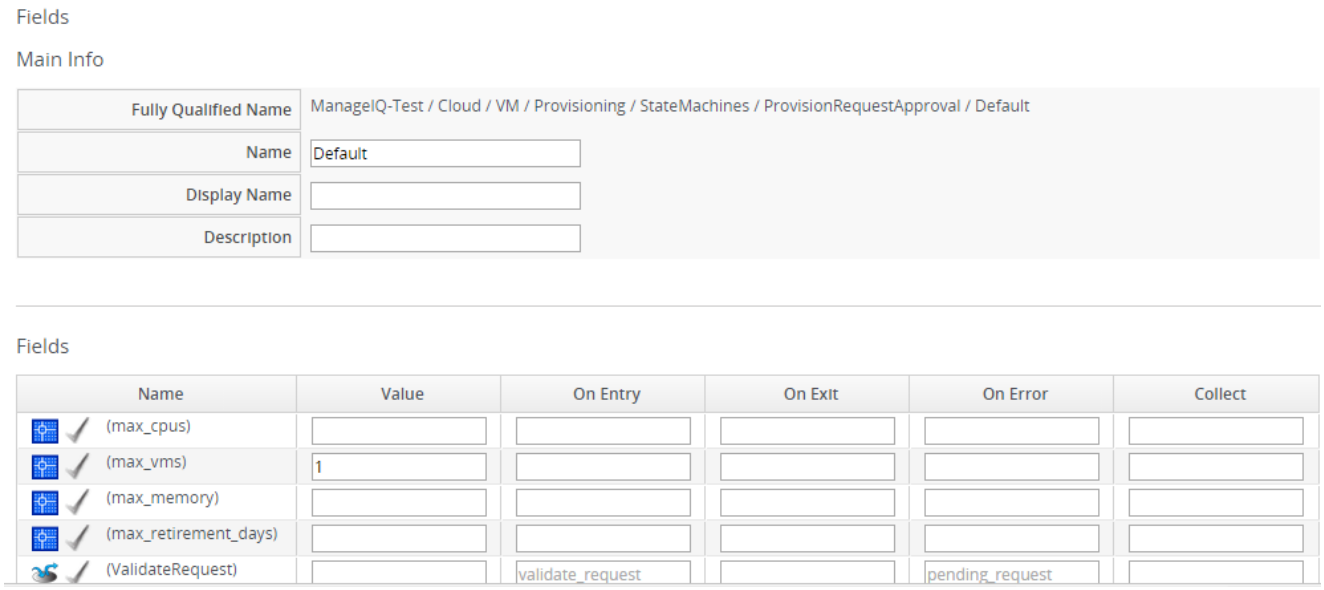
1. Navigate to **Automation > Embedded Automate > Explorer**.
2. From the tree in the accordion menu, select **DOMAIN > Cloud > VM

   Provisioning > StateMachines > ProvisionRequestApproval Class**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.

   This example uses the **Cloud** Namespace but can also use the **Infrastructure** namespace.
3. Click (**Configuration**), then (**Edit this instance**).



   Note:

   Do not change any values other than those listed below. Doing so may prevent the automatic approval process from running.

   - Use **max_cpus** to set the number of CPUs allowed to approve automatically the provisioning request.
   - Use **max_vms** to set the maximum number of virtual machines or instances that are allowed to be provisioned automatically approve the request. If this is set to blank, no requests will be automatically approved.
   - Use **max_memory** to set the maximum memory allowed to approve automatically the provisioning request.
   - Use **max_retirement_days** to set the maximum number of days until the virtual machine or instance is retired to automatically approve this request.
   - If a value is blank or **0**, the parameter is ignored.
4. Click **Save**.

The thresholds for automatic approval are set. The next time a provision request is created these thresholds will be checked. If the requirements are met, the provisioning request will be approved with no user intervention.

Template Specific Approval Defaults

Infrastructure Automation provides tags that can be used to set default automatic approval values on a per template or image basis. These values **supersede** those in the **Automate** model. Use these tags to eliminate the need for manual approval for all provisioning requests. To enable automatic approval, assign the tags directly to templates or images.

Category Display Name (Name)	Use (Sample values)
Auto Approve Max CPU (prov_max_cpus)	Sets the maximum number of CPUs that can be automatically approved in a single provisioning request. Sample Values: 1, 2, 3, 4, 5
Auto Approve Max Memory (prov_max_memory)	Sets the maximum number of memory that can be automatically approved in a single provisioning request. Sample Values: 1, 2, 4, 8 (in GB)
Auto Approve Max Retirement Days (prov_max_retirement_days)	Sets the maximum number of days until retirement that can be automatically approved in a single provisioning request. Sample Values: 30, 60, 90, 180 (in days)
Auto Approve Max VM (prov_max_vms)	Sets the maximum number of virtual machines or instances that can be automatically approved in a single provisioning request. Sample Values: 1, 2, 3, 4, 5

Assigning Tags to a Template for Auto Approval

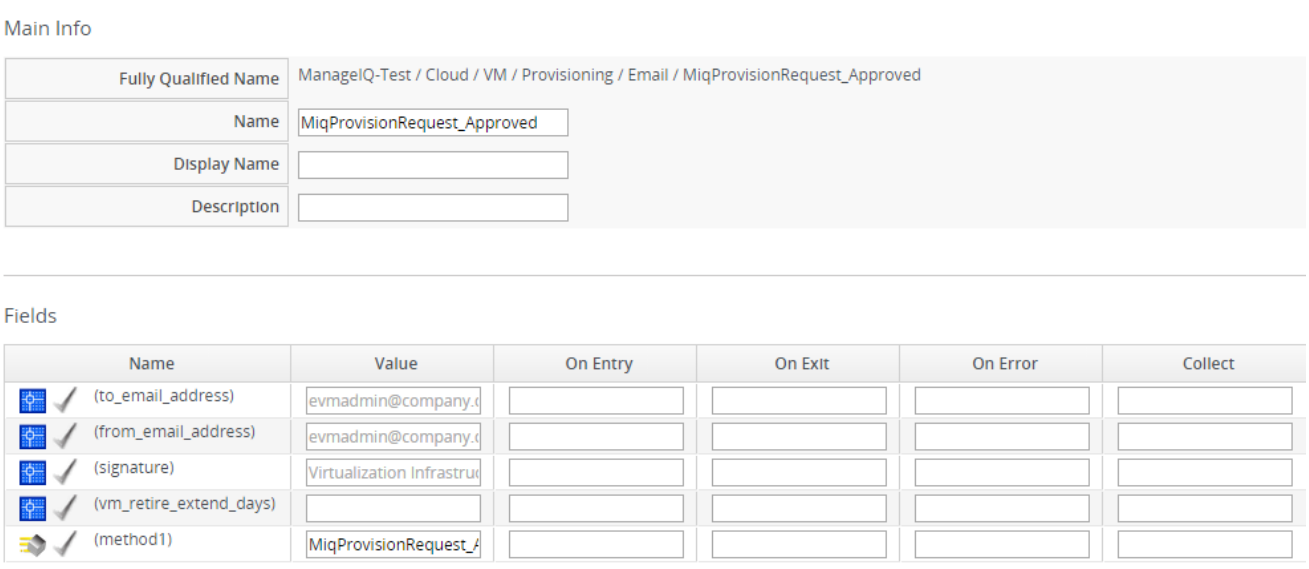
1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Click the **Templates** accordion, and select the templates that you want to tag.
3. Click (**Policy**), and then (**Edit Tags**).
4. Select a customer tag from the first dropdown, and then a value for the tag.

The thresholds for automatic approval for a specific template are set. The next time a provision request is created for this template these thresholds will be checked. If the requirements are met, the provisioning request will be approved with no user intervention.

Setting Provisioning Notification Email Addresses

Infrastructure Automation contains a set of Automate instances for provisioning. These Automate instances also include email fields to set the sender and recipient of provisioning notifications, such as requests. These fields are set to **evmadmin@company.com** as a default.

1. Navigate to **Automation > Embedded Automate > Explorer**.
2. Select an instance within the chosen class.
3. Click (**Configuration**), then (**Edit this instance**).
4. Type the desired email addresses in the **to_email_address** and **from_email_address** fields.
5. Click **Save**.



Fulfilling Requests

Fulfilling a Request

After a request has been approved, Infrastructure Automation then goes through the steps required to complete the request. The steps followed for a regular provision from a virtual machine to a virtual machine (not to a template) are found by navigating to **Automation > Embedded Automate > Explorer**, then listed under **DOMAIN > Cloud|Infrastructure > VM > Provisioning > StateMachines > VMProvision_VM > Provision VM from Template (template)**. The value for each state shows where the instance resides in the **Datastore** accordion. The default set of execution steps is shown below. For more information on state machines, see State Machines.

Default Execution Steps in States Instance

Step	Description
Customize Request	Apply customizations.
Acquire IP Address	Integrates with IPAM (IP Address Management) to get an IP Address.
Acquire MAC Address	Integrates with IPAM to get a MAC Address.
Register DNS	Integrates with IPAM to register with DNS.
Register CMDB	Integrates with CMDB (Configuration Management Database) to register with the CMDB.
Register AD	Integrates with IPAM to register with active directory.
Placement	Determine placement of virtual machine or instance.
PreProvision	Pre-provisioning steps.
Provision	Create the virtual machine or instance.
CheckProvisioned	Check that the new virtual machine or instance is in the VMDB.
PostProvision	Post-provisioning steps.
Register DHCP	Integrate with IPAM to register the IP address with DHCP Server.
Activate CMDB	Integrate with IPAM to activate the virtual machine or instance in the CMDB.
Email owner	Send email to owner that the virtual machine or instance has been provisioned.

Quotas

Quotas allow you to establish maximum usage thresholds for an user, group, or tenant for provisioned virtual machines or instances and are integrated into provisioning profiles. These maximums are checked after the approval but before the actual provision request is started. The quota is set for the tenant or group as a whole.

Applying User or Group Quotas

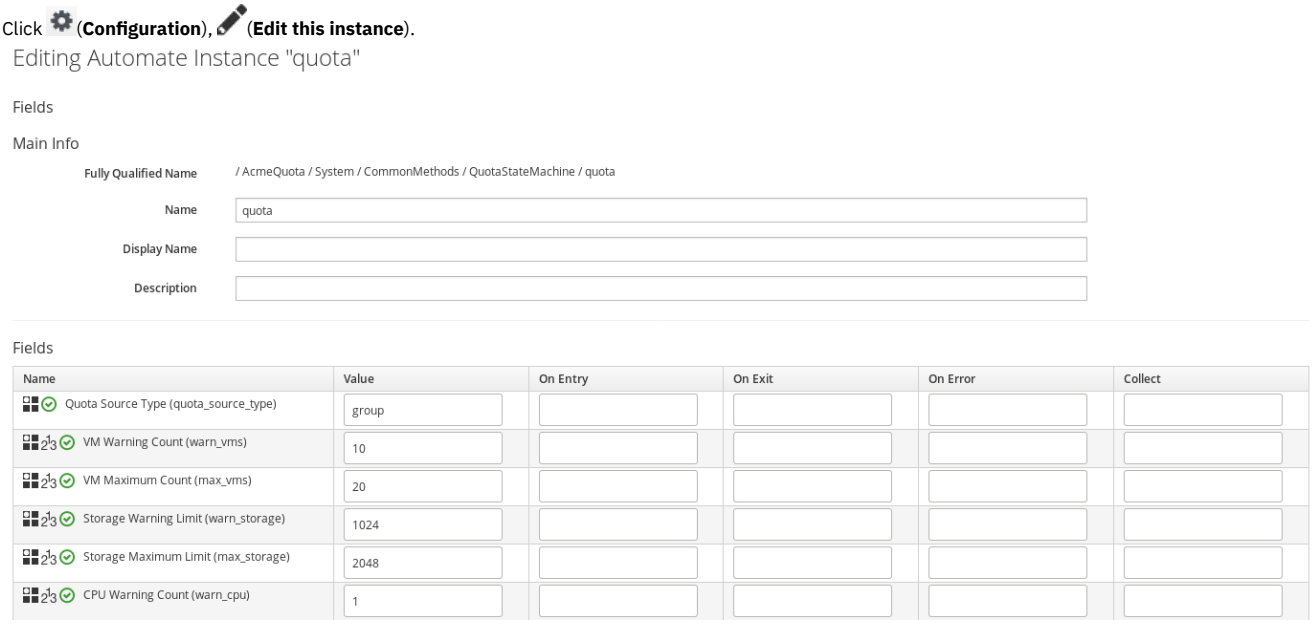
1. Log in as a user with administrator or super administrator rights to the Infrastructure Automation console.
2. Navigate to **Automation > Embedded Automate > Explorer**.
3. Copy the **ManageIQ > System > CommonMethods > QuotaStateMachine

   quota** instance to a custom DOMAIN.
4. From the accordion menu, click **DOMAIN > System > CommonMethods

   QuotaStateMachine > quota**.

   Note:

   By default, quotas are applied to tenants and do not require any change in **Automation > Embedded Automate > Explorer**.
5. Click (**Configuration**), (**Edit this instance**).

   1. Set the value for **Quota Source Type** to *user* or *group*.

      Important:

      A user creating a provisioning request must have an email address saved in their profile, or provisioning may fail. See Creating a User in *General Configuration* for details on configuring users.
   2. Set the values for **VM Warning Count**, **VM Maximum Count**, **Storage Warning Limit**, **Storage Maximum Limit**, **CPU Warning Count**, **CPU Maximum Count**, **Memory Warning Limit**, or **Memory Maximum Limit** to be the maximums for a specific user or group.
6. Click **Save**.

Using Tags for Owner and Group Quotas

Infrastructure Automation provides tags for enforcing quotas for the owners of virtual machines or instances. Ownership of a virtual machine or instance can be set either during the provisioning process or by using the **Configuration Set Ownership** button. If a virtual machine or instance has an owner, the value is displayed in the **Lifecycle** section of the virtual machine or instance summary page.

Quota tags can be assigned directly to **either** a group or owner **not** to a configuration item. The table below shows the tags for use in quotas.

Category Display Name (Name)	Use
Quota Max CPU (quota_max_cpu)	Sets the maximum number of CPUs summed over all virtual machines and instances owned by the group or user. Sample Values: 1, 2, 3, 4, 5, 10, 20, 30, 40, 50
Quota Max Memory (quota_max_memory)	Sets the maximum memory summed over all virtual machines and instances owned by the group or user. Sample Values: 1024, 2048, 4096, 8192, 10240, 20480, 40960, 81920 (in MB)
Quota Max Storage (quota_max_storage)	Sets the maximum storage summed over virtual machines and instances owned by the group or user. Sample Values: 10, 100, 1000, 20, 200, 40, 400 (in GB)

Applying a Tag to a User or User Group

1. Click  (**Settings**) > **Application Settings**.
2. Click the **Access Control** accordion, and select the user or group that you want to tag.
3. Click  (**Policy**), then click (**Edit Tags**).
4. Select the appropriate customer tag to assign, then the value.
5. Click **Save**.

Note:

When quotas are applied by both automate instance and tagging, the tagged values will have higher precedence.

State Machines

The automate state machine processes an ordered list of states. It can ensure the successful completion of a step before the next step is run, permit steps to be retried, allow setting a maximum time to retry the state before exiting, and number of retries before exiting the state. Before each state is executed, the `On_Entry` method is executed and after the state ends the `On_Exit` or `On_Error` method is executed based on how the state ends.

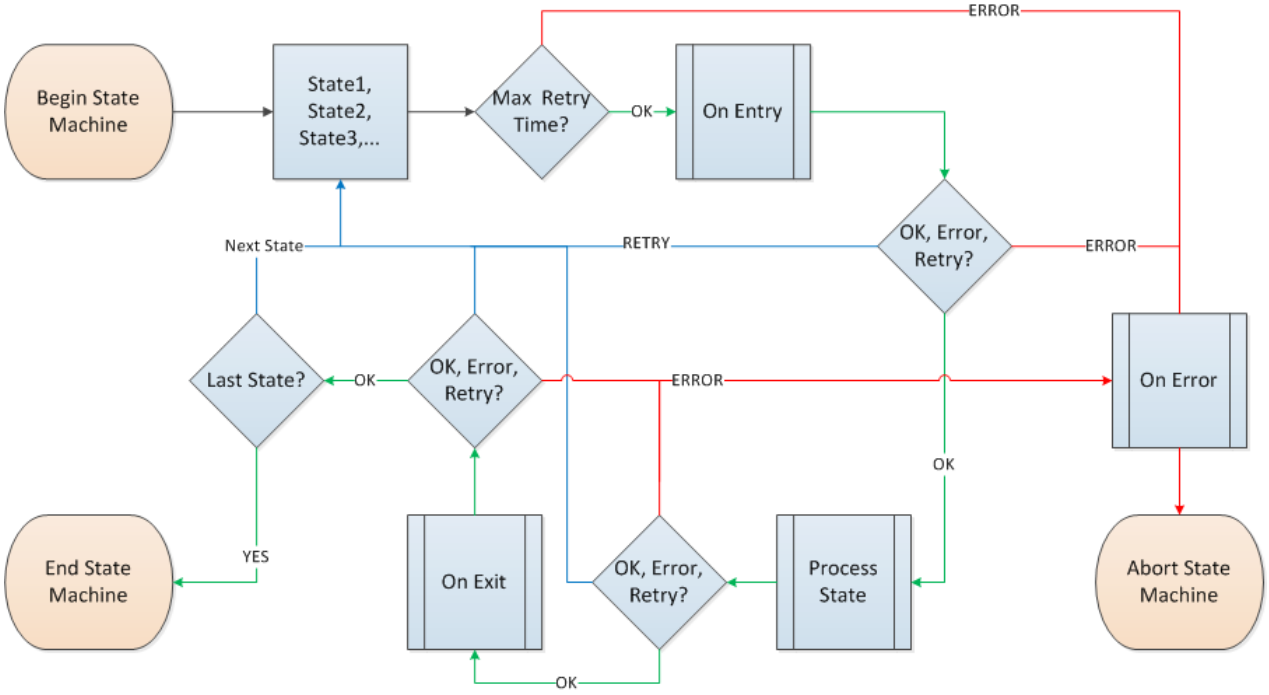
The following components make up a Infrastructure Automation automate state machine:

Component	Description
On_Entry	Method to run when entering the state. It enables you to execute an automate method to do some pre-processing before the state of the state machine is processed.
On_Exit	Method to run when exiting the state.
On_Error	Method to run if an error is encountered when running the state. It enables you to execute an automate method to do some final processing before the state machine finally exits (MIQ_ABORT) due to the error.
Default Value	Runs after the On_Entry method completes (The actual state being processed).
Max Retries	Maximum number of times to retry the state before exiting.
Max Time	Maximum time in seconds to retry the state before exiting.

In the diagram below, you can see how these components combine to create a state machine workflow.

Note:

The retry logic, `On_Entry` and `On_Error` are distinct cases in the program flow.



Code snippet demonstrating the state machine retry logic:

```
# Get current provisioning status
task = $evm.root['service_template_provision_task']
task_status = task['status']
result = task.status

Then check the result to see how it should proceed:

case result
when 'error'
  $evm.root['ae_result'] = 'error'
.....
when 'retry'
  $evm.root['ae_result'] = 'retry'
  $evm.root['ae_retry_interval'] = '1.minute'
when 'ok'
  $evm.root['ae_result'] = 'ok'
end

When the result is "retry", it sets:
  $evm.root['ae_result'] = 'retry'
  $evm.root['ae_retry_interval'] = '1.minute'
```

The following image shows a simple state machine pertaining to approving a provision request. This instance can be found in **Datastore > ManageIQ > Infrastructure > VM > Provisioning > StateMachines > ProvisioningRequestApproval > Default**.

Name	Value	On Entry	On Exit	On Error	Collect	Max Retries	Max Time	Message
max_vms	1							create
ValidateRequest		validate_request		pending_request		100		create
ApproveRequest		approve_request				100		create

1. The attribute **max_vms** has a value of 1. State machine processing can use the attributes of the state machine instance to make logic decisions. In this case, the **validate_request** method, which is processed during the **On_Entry** portion of the **ValidateRequest** state, evaluates the **max_vms** attribute. If the number of virtual machines requested is less than the **max_vms** value, the request can be auto-approved.
2. **ValidateRequest** is the first state to be executed.
3. **ApproveRequest** is the next state to be executed.

Note:

Grayed out items reflect values that are set in the class schema. These values can be overwritten on a per instance basis.

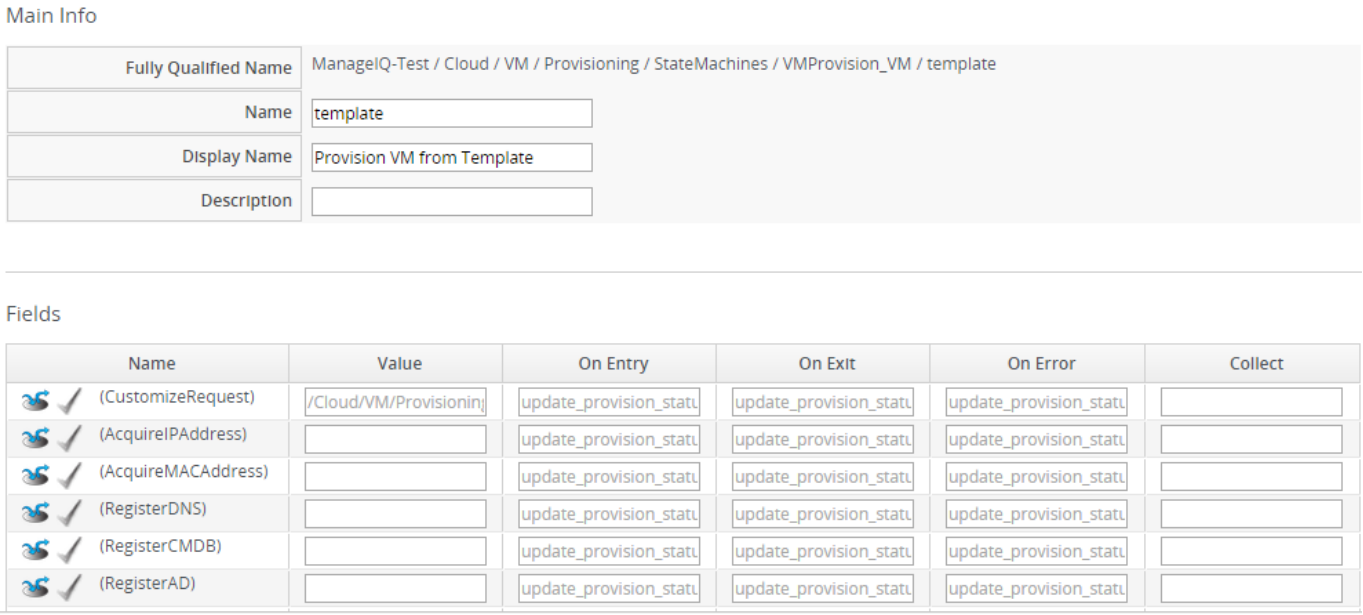
Customizing Provisioning States

The steps followed when provisioning a virtual machine or cloud instance are completed based on instances from the **DOMAIN > Cloud|Infrastructure > VM > Provisioning > StateMachines > VMProvision_VM** class.

Editing the Default State Instance

1. Navigate to **Automation > Embedded Automate > Explorer**.
2. Click **Datastore** accordion. Select **Configuration** > Add a New Domain**. You must fill the Name and Description values and click **Add** to create a custom domain.
3. From the accordion menu, navigate to **Cloud > VM > Provisioning > StateMachines > VMProvision_VM**.
4. Click on the drop-down menu, **Configuration > Copy this class**.
5. To add the class to the custom domain, specify the custom domain in the **To Domain** list and click Copy.
6. From the Datastore accordion, navigate to **custom domain > Cloud > VM > Provisioning > StateMachines > VMProvision_VM**.
    1. Click **Schema** tab.
    2. Select **Configuration > Edit selected Schema**.
7. You can remove, update, or insert a line in the Editing Class Schema panel.

Note: Depending on your environment, you can leave the values blank in a provisioning process. For example, if you are not integrating with IPAM or a CMDB, then you can leave the values blank.



1. On the Datastore tree, navigate to **custom domain > Cloud > VM > Provisioning > StateMachines > VMProvision_VM**. Click **Schema** tab and select Configuration > Edit sequence.
2. Resequence the schema in the Edit of Class Schema Sequence panel.

Viewing the Status of a Provisioning Request

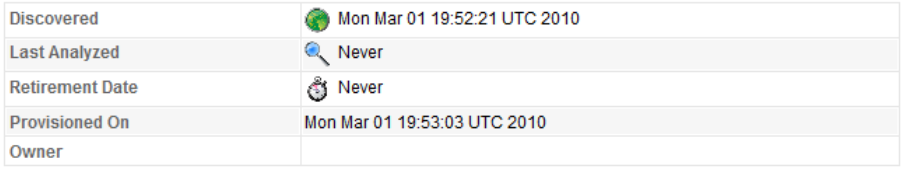
After a request has been approved, the various stages of fulfillment are executed. You can see the progress of the provisioning process by viewing its status.

1. Navigate to **Services > Requests**. The list of requests is shown.
2. Click on a specific request for more information. Once the provisioning begins, if the request was supposed to create more than one virtual machine or instance, a field will appear called **Provisioned VMs**. Click on the number that appears next to it for information on each of the individual provisions.

Viewing a Provisioned Virtual Machine or Instance

When a virtual machine or instance is created as a result of a provisioning request, its summary screen will show when it was provisioned in the **Lifecycle** area of the respective summary.

1. From **Services > Workloads**, click the virtual machine or instance that you want to view.



Viewing a Virtual Machine or Instance Summary

From **Services > Workloads**, click the virtual machine or instance that you want to view.

You can now request the log table from **Services > Request** to display logs related to a particular request.

Note: The table appears only if there are any logs to display. This table is sorted by the time, severity, and message.

Catalogs and Services

Through the use of catalogs, Infrastructure Automation provides support for multi-tier service provisioning to deploy layered workloads across hybrid environments. You can create customized dialogs that will give consumers of the services the ability to input just a few parameters and provision the entire service. The following table lists the terminology associated with catalogs that you will use within the Infrastructure Automation user interface for service provisioning.

Type	Information
Catalog Bundle	A group of templates.
Catalog Item	A single template.
Template	A template is a copy of a preconfigured virtual machine, designed to capture the installed software and software configurations, as well as the hardware configuration of the original virtual machine.
Dialog Tabs	Part of a service dialog.
Element	An item on a tab in a dialog. It can be a button, check box, drop down list, radio button, tag control, text area box, or a text box.
Provisioning Dialogs	Dialogs created for host provisioning, virtual machine migration, or virtual machine provisioning. The dialog name must be added to the appropriate provision instance to be processed.
Service Catalog	A catalog item or catalog bundle that is available for provisioning.
Service Dialogs	Made up of fully customizable tabs, items, and values for use with service provisioning.

Terminology

Generic Objects

Generic Objects are object-like entities, defined at runtime, that have unique names and user-defined attributes and relationships. Residing in the Automate Engine datastore, generic objects are designed to manage objects other than those related to private infrastructure, and public or private cloud providers.

Using automate requests, services and catalog items, generic objects can be directly accessed or passed, during any step, as a parameter to a service state machine. As a result, generic objects can be used to quickly add the capability to provision and collect data on resources not supported by Infrastructure Automation.

Viewing Generic Objects Classes

View a list of generic objects and click through to see detailed summary information for each object.

1. Navigate to **Automate > Generic Objects**.
2. Click on a generic object class in the table to view its summary information.

Creating Generic Objects Classes

Model a new resource by creating a generic object class and adding it to your Infrastructure Automation inventory. Each generic object class can have attributes, associations, and methods. Once created, generic object classes are visible to users of the Self Service user interface at the resource level.

Create a generic object class using the following steps:

1. Browse to menu: **Automation > Automate > Generic Objects**.
2. Click **Configuration**, then click (**Add a New Generic Object Class**).
3. Provide a **Name** and **Description** for the new object class.
4. In the **Attributes** field, enter a **Name** and choose a **Type** from the drop-down list. Click the button to add attributes.
5. Enter a **Name** and select a **Class** for the object class's **Associations**. Click the button to create additional associations.
6. Provide a **Name** for the **Methods**. Click the button to add methods.
7. Click **Add**.

Editing Generic Object Classes

Edit existing generic object classes using the following steps:

1. Browse to menu: **Automation > Automate > Generic Objects**.
2. Click on a generic object class in the list view.
3. Click **Configuration**, then click (**Edit this Generic Object Class**).
4. Make required changes to the generic object class fields.
5. Click **Save**.

Removing Generic Objects Classes

Remove generic object classes from your inventory using the following steps:

1. Browse to menu: **Automation > Automate > Generic Objects**.

2. Check the generic objects classes from the table to remove.
3. Click **Configuration**, then click (**Remove selected Generic Object Classes from Inventory**).
4. Click **OK** to confirm.

Exporting Generic Objects

Export generic objects from Infrastructure Automation to create a shared library.

Exporting a generic object requires the object has been created in Infrastructure Automation.

Procedure

1. ssh to the Infrastructure Automation appliance as **root**.
2. Browse to **/var/www/miq/vmdb**:

   ```
   # cd /var/www/miq/vmdb
   ```

3. Create a temporary directory to store the generic object definitions:

   ```
   # mkdir tmp/generic_object_definitions
   ```

4. Export the generic object definitions using the following **bin/rake** command:

   ```
   # bin/rake evm:export:generic_object_definitions -- --directory tmp/generic_object_definitions
   ```

Locate the exported yaml file containing the generic object properties in the temporary folder.

Importing Generic Objects

Import generic objects to Infrastructure Automation from a shared library.

Procedure

1. ssh to the Infrastructure Automation appliance as **root**.
2. Browse to **/var/www/miq/vmdb**:

   ```
   # cd /var/www/miq/vmdb
   ```

3. Create a temporary directory to store the generic object definitions:

   ```
   # mkdir tmp/generic_object_definitions
   ```

4. Copy the yaml file containing the generic object definitions to the temporary folder:

   ```
   # cp generic_objects.yaml tmp/generic_object_definitions
   ```

5. Import the generic object definitions to the Infrastructure Automation appliance using the following **bin/rake** command:

   ```
   # bin/rake evm:import:generic_object_definitions -- --source tmp/generic_object_definitions
   ```

Confirm the generic objects have imported successfully by Viewing Generic Objects Classes in the Infrastructure Automation user interface.

Service Dialogs

When provisioning a service, input will be needed from the requester. Service dialogs are used to take input from the user. This input is connected to a method in the Automate model that defines how the user's input is translated into the provision request. Before creating a service dialog, be sure to plan what items you need the user to input.

Adding a Service Dialog

- When creating a service dialog for use with Ansible playbook catalog items, variable elements must use the prefix **param_** when assigning the value. For example, a new variable labeled **key1** should have its value set as **param_key1**. - Using Ansible playbooks to populate dynamic dialog fields is not recommended due to delay times caused by the overhead of interaction between systems. - If you add the playbook automate method to a service dialog, only users with admin privileges can run the dialog.
Infrastructure Automation includes a drag-and-drop service dialog editor to create service dialogs. The editor, with its drag-and-drop feature, provides a visual representation of the components that comprise a service dialog. You can easily design your service dialog utilizing dialog tabs, sections (previously referred to as boxes), and elements.

When users access a service, the majority of options available to them are preset and cannot be altered. The requirements for the service determine the options and fields that need to be present in the dialog for user input. A service dialog exposes some of those options to the user so that even if they are ordering a basic Red Hat Enterprise Linux 7 machine, for example, they can at least choose the amount of memory, virtual CPUs, or other options available to the instance they order. In cases where certain fields must be unique, such as the name of virtual machines in Red Hat Virtualization, users must enter their own unique name for the virtual machine they choose or the operation will fail, so this field must be exposed.

A service dialog contains three components:

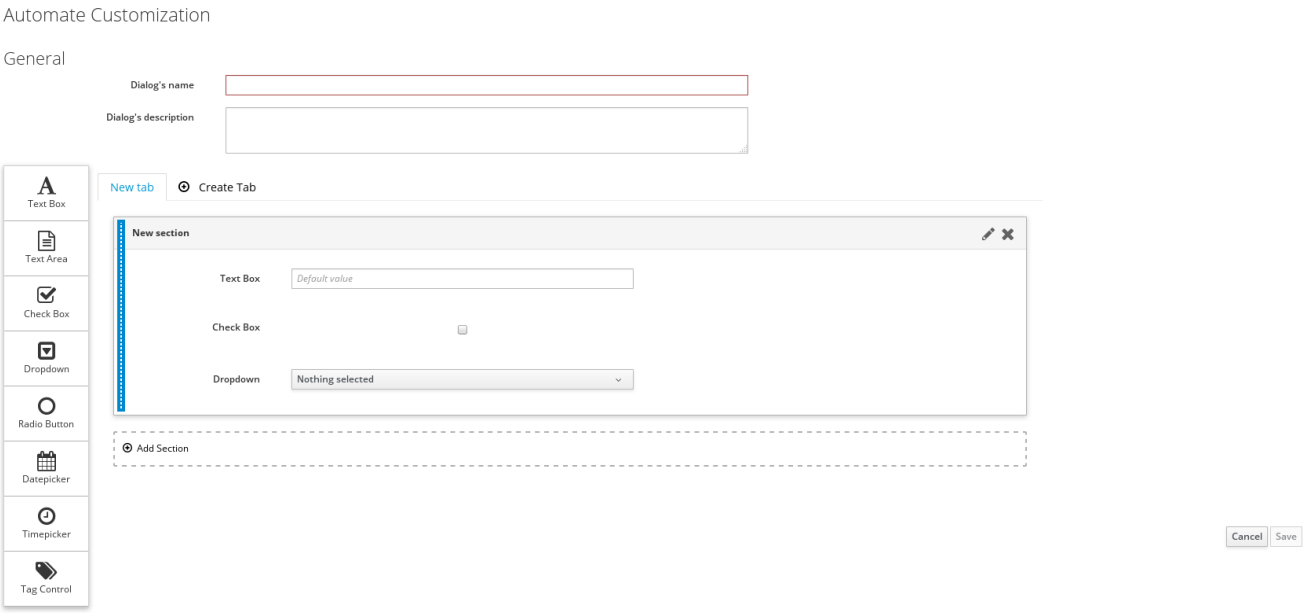
- One or more **Tabs**.
- Inside the tabs, one or more **Sections**. Note that in the previous method of creating service dialogs using the Infrastructure Automation user interface, **Sections** were referred to as **Boxes**.

- Inside the sections, one or more **Elements**. Elements are controls that accept input. Elements contain methods, like check boxes, drop-down lists or text fields, to fill in the options on the provisioning dialog.

Important:

The names of the elements must correspond to the options used in the provisioning dialog.

1. Browse to menu: **Automation > Automate > Customization**.
2. Click the **Service Dialogs** accordion.
3. Click **Configuration**, and then **Add a new Dialog**.



4. Enter basic details under **General**:
    1. Enter the **Dialog's name** and **Dialog's description**.
5. Add a new tab to the dialog:
    1. Click **Create Tab**. Then, click the icon on the new tab to edit tab information.
    2. Enter a **Label**.
    3. Optional: Enter a description for the tab in **Description**.
    4. Click **Save**.
6. Add a new section to the tab:
    1. Click **Add Section**. Then, click the icon on the upper-right to edit section details.
    2. Enter a **Label**.
    3. Optional: Enter a description for the section in **Description**.
    4. Click **Save**.
7. Add elements to the section:
    1. From the list of elements on the left, click an element you want to add, then drag-and-drop it inside the section. Then, click the icon next to the element to edit its field details.

Element Types	Additional Info
Text Area	Provides text area for users to enter text. You can enter the default text in **Default Value**, or leave it as blank.
Text Box	Text box is similar to a text area with the added option to enable **Protected** so the text is shown as asterisks(*), instead of plain text.
Check Box	Enable **Default Value** if you want the box checked by default.
Drop Down	Use drop down to create list entries either manually or using automate methods. Enable **Dynamic** to create lists using automate methods; use **Entry Point** to select an automate instance. Enable **Show Refresh Button** to allow users to refresh list options manually.
Radio Button	Similar to a drop down but displays options using radio buttons.
Datepicker	Use this to enable users to pick a date by clicking the calendar icon.
Timepicker	use this to enable users to pick a date and time.

Element Types	Additional Info
Tag Control	Select a **Category** of tags you want assigned to virtual machines associated with the service dialog. Enable **Single Select** if only one tag can be selected.

2. Enter a **Label**, **Name**, and **Description** for the element.

   Element names must correspond to the options used in the provisioning dialog. **Name** must use only alphanumeric characters and underscores without spaces. It is also used to retrieve the value of this element in the method used with the dialog and must start with **dialog_service_type**.
3. Optional: Add additional information in **Help** to assist the user to complete the fields in the service dialog. This field is useful for explaining unfamiliar terminology or providing configuration tips. This information is presented when you hover over the [!] exclamation mark in the Service Dialog while ordering a Service Catalog later.
4. Set other options as required.
5. Click **Save**.

8. Optional: Repeat the above step to add more elements to the existing section, or create and add elements to a new section as required.
9. Optional: Repeat the step to add a new tab to the dialog, and subsequent steps to add sections and elements to it as required.
10. Click **Save** to create the dialog.

The service dialog is now created, and added to the **Service Dialogs** accordion.

Creating a Service Dialog from a Container Template

Complete the following procedure to create a Service Dialog from a Container Template.

1. Navigate to **Compute > Containers > Container Templates** and select the template for provisioning.
2. Click (**Configuration**), then (**Create Service Dialog from Container Template**).
3. Enter a name for the dialog in **Service Dialog Name**.
4. Click **Save**.

You can use this service dialog when creating a catalog item for container template provisioning; see Creating an OpenShift Template Catalog Item.

Importing Service Dialogs

You can share service dialogs between appliances using the export and import features.

1. Navigate to **Automation > Embedded Automate > Customization**.
2. In the **Import/Export** accordion, click **Service Dialog Import/Export**.
3. In the **Import** area, click **Browse** to select an import file.
4. Click **Upload**.

Exporting Service Dialogs

You can share service dialogs between appliances using the export and import features.

1. Navigate to **Automation > Embedded Automate > Customization**.
2. In the **Import/Export** accordion, click **Service Dialog Import/Export**.
3. In the **Export** area, select the service dialogs that you want to export.
4. Click **Export**.

Methods

Creating a Method to Associate with the Dialog

You will need to create a method that connects the values in the dialog with the provisioning request. The method should be created in the `DOMAIN/Service/Provisioning/StateMachines/ServiceProvision_Template` class of the **Automate** model.

Note:

DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.

A method is provided below that was created for the following scenario:

- You want to provision a three-tiered service that contains catalog items of web, app and DB. Each of these virtual machines (or cloud instances) has been tagged under the **Service** category with the appropriate value. Then, added as a catalog item and combined into a catalog bundle.
- The **Service Dialog** captures the selection of small, medium or large application in a dropdown called **service_type**. When referring to a value captured in an element in a dialog, the name of the element should be prefixed with **dialog_**. For example, **service_type** becomes **dialog_service_type** when used in the method.

- The method will set the memory sizes for each of the catalog items based on the **service_type** selection.

```
#            Automate Method
#
$evm.log("info", "Automate Method ConfigureChildDialog Started")
#
#            Method Code Goes here
#
$evm.log("info", "==========================================")
$evm.log("info", "Listing ROOT Attributes:")
$evm.root.attributes.sort.each { |k, v| $evm.log("info", "\t#{k}: #{v}")}
$evm.log("info", "==========================================")

stp_task = $evm.root["service_template_provision_task"]
$evm.log("info", "==========================================")
$evm.log("info", "Listing task Attributes:")
stp_task.attributes.sort.each { |k, v| $evm.log("info", "\t#{k}: #{v}")}
$evm.log("info", "==========================================")

#############################################################
#### This is how the method would look for dialog variables
#############################################################
dialog_service_type = $evm.root['dialog_service_type']
$evm.log("info","User selected Dialog option = [#{dialog_service_type}]")

stp_miq_request_task = stp_task.miq_request_task
#$evm.log("info","(parent) miq_request_task:  = [#{stp_miq_request_task}]")

#############################################################
#### This is how you get the catalog items for the catalog bundle
#############################################################

stp_miq_request_tasks = stp_task.miq_request_tasks
#$evm.log("info","(children) miq_request_tasks count:  = [#{stp_miq_request_tasks.count}]")

#############################################################
#### By going through the children, you can set the dialog variable for each of the children (we based our values on the
childrens service tags)
#############################################################

stp_miq_request_tasks.each do |t|

  $evm.log("info"," Setting dialog for: #{t.description}")
  service = t.source
  service_resource = t.service_resource
  #$evm.log("info"," Child service resource name: #{service_resource.resource_name}")
  #$evm.log("info"," Child service resource description: #{service_resource.resource_description}")

  service_tag_array = service.tags(:app_tier)
  service_tag = service_tag_array.first.to_s

  memory_size = nil

#############################################################
#### The dialog_service_type is the attribute set on the service dialog
#### We use the service_tag to decide what child gets what dialog
#############################################################

  case dialog_service_type
  when "Small"
    case service_tag
    when "app"
      memory_size = 1024
    when "web"
      memory_size = 1024
    when "db"
      memory_size = 4096
    else
      $evm.log("info","Unknown Dialog type")
    end
  when "Large"
    case service_tag
    when "app"
      memory_size = 4096
    when "web"
      memory_size = 4096
    when "db"
      memory_size = 8192
    else
      $evm.log("info","Unknown Dialog type")
    end
  else
    $evm.log("info","Unknown Dialog type - setting Dialog options here")
  end

#############################################################
#### set_dialog_option sets the dialog for the child
#############################################################

  t.set_dialog_option('memory',memory_size) unless memory_size.nil?
  $evm.log("info","Set dialog for selection: [#{dialog_service_type}]  Service_Tier: [#{service_tag}] Memory size: [#
```

```
{memory_size}]")

end
#
#
#
$evm.log("info", "Automate Method ConfigureChildDialog Ended")
exit MIQ_OK
```

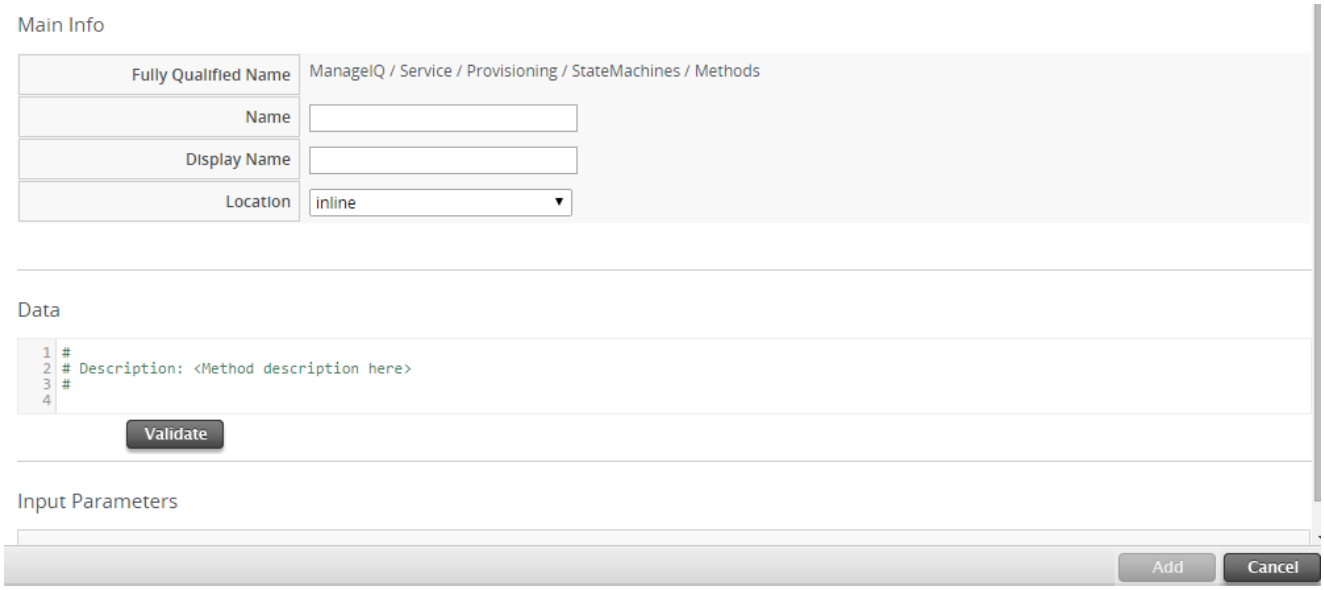
Creating a Method in the Service Class

Service methods have been split based on purpose.

1. Navigate to **Automation > Embedded Automate > Explorer**.
2. Service Class is located at **DOMAIN > Service > Provisioning > StateMachines > Methods** and **DOMAIN > Service > Retirement > StateMachines > Methods**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.
3. Click the **Methods** tab.
4. Click (**Configuration**), then (**Add a New Method**).
5. Enter a **Name** and **Display Name**.
6. In the **Data** field, enter the method contents.
7. Click **Validate** and wait for your data entry to be successfully validated.
8. Click **Add**.



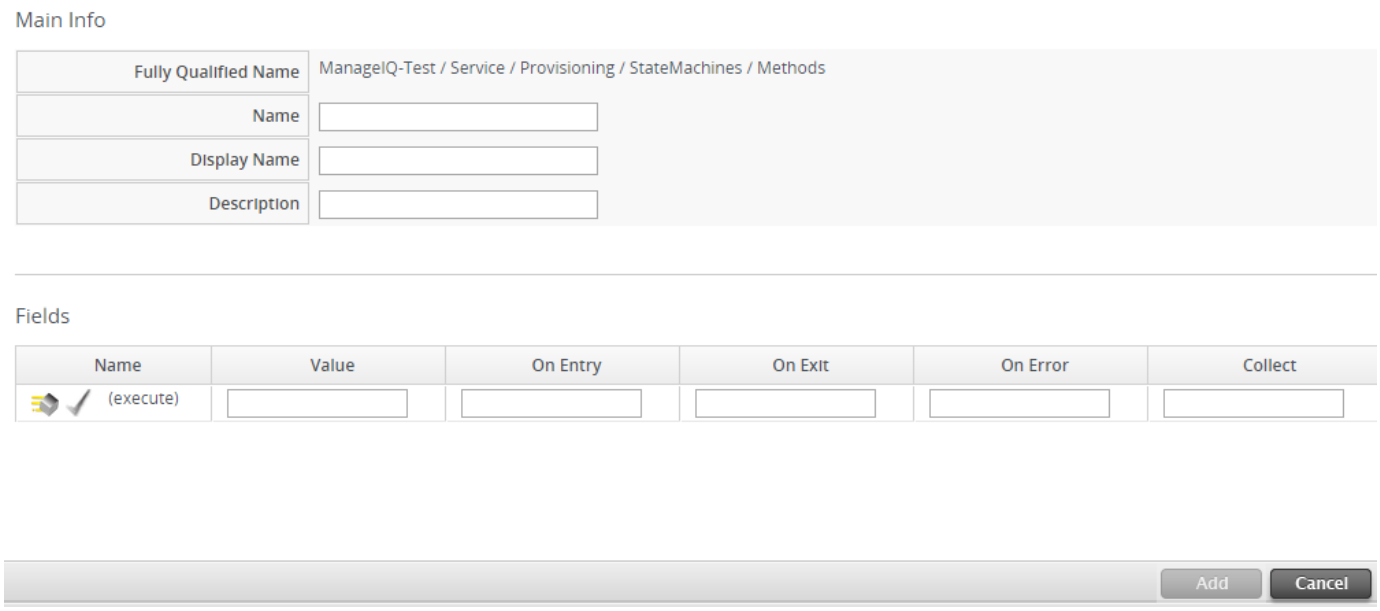
Creating an Instance in the Service Class

1. Navigate to **Automation > Embedded Automate > Explorer**.
2. Service Class is located at **DOMAIN > Service > Provisioning > StateMachines > Methods** and **DOMAIN > Service > Retirement > StateMachines > Methods**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.
3. Click the **Instances** tab.
4. Click (**Configuration**), then (**Add a new Instance**).
5. Enter a **Name** and **Display Name**.
6. In the **Fields** area, enter the method's name in **Value**.
7. Click **Add**.

The instance is created so that it can be called from the **ServiceProvision** class.



Note:

After the method has been created, it must be mapped to an instance in the **`DOMAIN/Service/Service/Provisioning/StateMachines`** class. The name of the instance must be specified as the **Entry Point**. This method must be called before the provision job begins.

Associating a Method with an Automate Instance

Service methods have been split based on purpose.

1. Navigate to **Automation > Embedded Automate > Explorer**.
2. From the accordion menu, click the required service method.
3. Service Class is located at **DOMAIN > Service > Provisioning > StateMachines > Methods** and **DOMAIN > Service > Retirement > StateMachines > Methods**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.
4. Either create a new instance or select the **clone_to_service** instance.
5. Click (**Configuration**), then (**Edit Selected Instance**).
6. In the **configurechilddialog** value, put the path to the method.
7. Click **Save** or **Add** if you are adding this to a new instance.

Provisioning Dialog

The following example shows how to create a Provisioning Dialog that modifies the parameters for provisioning a small, medium, or large service.

The **`provision-vm-small-med-large.asl`** workflow runs for every provisioned VM and is responsible for modifying the VM size. This workflow is the Provisioning service entry point.

You need to set the Provisioning service entry point to workflow script. For example, this workflow is run for every provisioned Virtual Machine (VM), and modifies the VM size.

Use the following sections and follow them in order to provision a VM service, create a Provisioning Dialog and modify the parameters.

Create a workflow (optional)

Note: If you want to use the existing **`provision-vm-small-med-large.asl`** workflow script without any changes, you can skip this section and proceed to **`Add a Workflow Repository`** section.

1. Fork **`https://github.com/ManageIQ/workflows-examples`** repository or create your own repository.
2. Clone the repo locally by using the clone command:

   ```
   git clone https://github.com/<owner>/workflows-examples
   ```

   Where **`<owner>`** is the owner of the repository.
3. Create a new branch:

   ```
   git checkout -b sm-md-lg
   ```

4. Create your own script or edit the existing script that is provided. The script used in this example is `provision-vm-small-med-large.asl`.
5. Push code changes to your branch.

Add a Workflow Repository

Use the following steps to add a Workflow Repository.

1. Click **Automation** > **Embedded Workflows** > **Repositories**.
2. Click **Configuration** > **Add new Repository**.
3. Provide the **Name** and **URL** for the repository:
   - **Name**: Local name to identify this repository. For example, `Example Workflows`.
   - **Url**: Git repository URL. For example, `https://github.com/ManageIQ/workflows-examples`.
   - **SCM Branch**: Git repository branch name. For example, leave this field blank if you want to use the existing `provision-vm-small-med-large.asl` script from the master branch, or use a branch name like `sm-md-lg` if you have edited or created your own workflow script.
4. Click **Save**.

Create a Provisioning Dialog

Use the following steps to create a Provisioning Dialog.

1. Click **Automation** > **Embedded Automate** > **Customization**.
2. Click **Service Dialogs** > **All Dialogs**.
3. Click **Configuration** > **Add a new Dialog**.
4. Provide a **Name** and **Description** for the dialog:
   - **Name**: Name used in the Service Catalog. For example, `sm-med-lg dialog`.
   - **Description**: Description for the dialog. For example, `Dialog choosing vm size`.
5. Add a text box named **vm_name**.
6. Add a dropdown named **size**.
7. Click the edit pencil icon for the **size** dropdown.
8. Under **Options** > **Entries** enter the following values:
   - Enter the following for the first row:
     - For **Name** field, type `Large`.
     - For **Value** field, type `large`.
   - Enter the following for the second row:
     - For **Name** field, type `Medium`.
     - For **Value** field, type `medium`.
   - Enter the following for the third row:
     - For **Name** field, type `Small`.
     - For **Value** field, type `small`.
9. Click **Save**.

Create a Service Catalog Item

Use the following steps to create a Service Catalog Item.

1. Click **Service** > **Catalogs**.
2. Click **Catalogs** > **All Catalogs**.
3. Click **Configuration** > **Add a New Catalog**.
4. Provide a **Name** and **Description**:
   - **Name**: Name displayed in the catalog. For example, `Provision VM`.
   - **Description**: Description of the catalog. For example, `Provisioning VM by size`.
5. Click **Save**.
6. Click **Catalog Items** > **All Catalog Items** > **Provisioning**.
7. Click **Configuration** > **Add a New Catalog Item**.
8. Provide a value for `Catalog Item Type`:
   - **Catalog Item Type**: The functionality of this catalog item. Choose **VMware**.
9. Under the **Basic Information** tab, provide the values for the following fields:
   - **Name**: Name displayed in the catalog. For example, `Provision VM with Size`.
   - **Description**: Description of the catalog. For example, `Provision a VM choosing size`.
   - **Display in Catalog**: Check to display this catalog item.
   - **Catalog**: Organization concept. For example, `My Company/Provisioning`.
   - **Dialog**: The dialog created previously. For example, `sm-med-lg dialog`.
   - **Provisioning Entry Point**:
     - For **endpoint type**, choose **Embedded Workflow**.
     - For the **endpoint value** choose the workflow script. For example, `provision-vm-small-med-large.asl`.
10. Under the **Request Info** > **Catalog** tab enter the following values:
    - **VM Name**: Name that is overwritten by the **VM name** in the dialog. Enter anything here.
11. Under the **Environment** tab, choose an appropriate **Host** and **Datastore**:
    - **Host**: Host that runs the VM.
    - **Datastore**: Disk that stores the VM.
12. Under the **Networking** tab, choose an appropriate **Network Adapter**.
13. Click **Add**.

Provisioning a Virtual Machine

1. Click **Services** > **Catalogs** > **Service Catalogs**.
2. Click **Order** under the newly created service, **Vm with Size**.

3. Provide the following fields:
    - **VM name**: Name for the newly created VM. For example `demo-1`.
    - **Size**: The desired VM size. You can choose `Small`, `Medium` or `Large`. For example, choose **Small**.
4. Click **Submit**

Iterate

If the provisioning did not go as planned use the following steps:

1. Edit the workflow script.
2. Push code changes to your branch.
3. Refresh the Workflow repository.
4. You can follow the workflows link to view the version of the script is stored on the server.
5. If you changed the script name, edit the Service Catalog Item that you previously created.
6. Provision the Virtual Machine.

Catalogs

Catalogs are used to create groups of virtual machines or instances for provisioning. For example, a complete package of a database server, desktop with specialized software already on it, and a firewall. You will need to complete the following steps to create and provision a service catalog.

1. Create **Catalog Items** for each virtual machine or instance that will be part of the service.
2. Create a **Service** dialog. For example, create a dropdown with three options small, medium, and large.
3. Create a method for the Service Dialog. This method defines what each of the options means to each of the individual virtual machines or cloud instances for the service. This method is called from a service provisioning instance in the Automate model.
4. Create an instance in the `DOMAIN/Service/Provisioning/StateMachines/ServiceProvision_Template` class that calls the method.

    Note:

    DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.
5. Associate method with Automate instance.
6. Create a **Catalog Bundle**, adding each of the catalog items to it. Select the **Service Dialog** you created. Use the instance created in the `DOMAIN/Service/Provisioning/StateMachines/ServiceProvision_Template` class as the **Entry Point**. Check **Display in Catalog** box.
7. Provision a service.

Creating a Catalog Bundle

1. Navigate to **Services > Catalogs**.
2. Click the **Catalog Items** accordion.
3. Click (**Configuration**), and then (**Add a New Catalog Bundle**).
4. Enter a name and description for the bundle. Select **Display in Catalog**.
5. Select the required **Catalog**.
6. Select the required **Dialog**.
7. Select chargeback currency.
8. Enter **Price / Month**.
9. Select the provisioning and retirement entry points.
10. Select **Additional Tenants**.
11. Click on the **Resources** tab, then select the catalog item you want to add to the bundle from the **Add a Resource** dropdown.
12. Click **Add**.

A catalog bundle is created and visible in the **Service Catalog** accordion.

Note:

You should also create and specify an Entry Point in the `DOMAIN/Service/Provisioning/StateMachines/Methods/CatalogBundle` class for each catalog item that is part of a bundle. If you do not, then the pre and post provision processing will occur for each item in the bundle in addition to processing for the **Catalog Bundle**. To set the entry point, go into each **Catalog Item** and check **Display in Catalog**. Then, you will see the **Entry Point** field.

Copying a Catalog Bundle

1. Navigate to **Services > Catalogs**.
2. Click the **Catalog Items** accordion.
3. Select the catalog bundle you want to copy from **All Catalog Items**.

4. Click (**Configuration**), and then (**Copy Selected Item**).
5. Enter a name for the copy of catalog bundle you are creating. Note that this name must not already be in use.
6. Click **Add**.

A copy of the catalog bundle is saved. You can now select this new copy of the catalog bundle and edit as required by navigating to (**Configuration**), then clicking (**Edit this Item**).

Creating a Catalog Item

Create a catalog item for each virtual machine or cloud instance that will be part of the service.

1. Browse to menu: **Services > Catalogs**.
2. Click the **Catalog Items** accordion.
3. Click **Configuration**, and then (**Add a New Catalog Item**).
4. Select the **Catalog Item Type** you are adding. This list only shows items related to providers available or options activated in the appliance (for example, **Ansible Playbook** is available as a **Catalog Item Type** option if the Embedded Ansible server role is enabled on the appliance).
5. Enter a name and description for the bundle. Select **Display in Catalog**.
6. Select a **Catalog**.
7. Select a **Dialog**.

   Note:

   You can only choose from the catalogs and dialogs you have already created. If you haven't done so, leave the values blank and edit later.
8. Select a **Zone**.
9. Select the chargeback currency.
10. Enter **Price / Month**.
11. Select the catalog item's **Subtype**.
12. Select the provisioning and retirement entry points, that is the Automate instance to run upon provisioning and retirement.

    Note:

    The entry point must be a State Machine since the **Provisioning Entry Point** list is filtered to only show State Machine class instances. No other entry points will be available from the **Provisioning Entry Point** field.
13. Select **Additional Tenants**.
14. Click **Add**.

Copying a Catalog Item

Note:

When copying a catalog item for reuse, you must click **Display in Catalog** in the copied catalog item for the item to appear in the catalog.

1. Navigate to **Services > Catalogs**.
2. Click the **Catalog Items** accordion.
3. Select the catalog item you want to copy from **All Catalog Items**.
4. Click (**Configuration**), and then (**Copy Selected Item**).
5. Enter a name for the copy of catalog item you are creating. Note that this name must not already be in use.
6. Click **Add**.

A copy of the catalog item is saved. You can now select this new copy of the catalog item and edit as required by navigating to (**Configuration**), then clicking (**Edit this Item**).

Creating a Generic Catalog Item

Create generic catalog items for services non-specific to virtualization or cloud environments. This catalog item type can serve a wide array of needs, from creating a vLAN across a network to accessing virtual machine IP addresses and adding them to a load balancer pool.

1. Browse to menu: **Service > Catalogs**.
2. Click the **Catalog Items** accordion.
3. Click **Configuration**, and then (**Add a New Catalog Item**).

4. Select **Generic** from the Catalog Item Type list.
5. In the **Basic Info** subtab:
    1. Type a **Name/Description**.
    2. Check **Display in Catalog** to display the item in the catalog. A **Dialog** will be required if you select **Display in Catalog**.
    3. Choose a **Catalog** to which to add the new item.
    4. Select a **Dialog** from the available options.
    5. Choose a **Subtype** from the list menu.
    6. Add **Entry Point(NS/Cls/Inst)** options.
        1. **Provisioning Entry Point (Domain/NS/Cls/Inst)** requires you to select an Automate instance to run upon provisioning.
        2. **Retirement Entry Point (Domain/NS/Cls/Inst)** requires you to select an Automate instance to run upon retirement.

            Note:

            The entry point must be a State Machine since the **Provisioning Entry Point** list is filtered to only show State Machine class instances. No other entry points will be available from the **Provisioning Entry Point** field.
6. In the **Details** subtab, write a **Long Description** for the catalog item.
7. Click **Add**.

Creating an Ansible Playbook Service Catalog Item

Create a catalog item that uses an Ansible Playbook to back it.

Note:

- Before creating an Ansible service, at least one repository, one playbook, and one credential must exist in the Infrastructure Automation inventory. Check your inventory and add the appropriate resources before creating an Ansible service. For more information, see Automation Management Providers in *Managing Providers*.
- Debugging verbosity is available for Ansible playbook catalog items. Selecting a higher verbosity value provides more detailed output as the playbook executes. **0 (Normal)** is the default value. **1 (Verbose)** will yield return data while a value of **3 (Debug)** provides connection attempt and task invocation details. Higher levels, such as **4 (Connection)** can be useful for debugging SSH connections. Use **5 (WinRM Debug)** when debugging WinRM connections.
- Using Ansible playbooks to populate dynamic dialog fields is not recommended due to delay times caused by the overhead of interaction between systems.
- Only users with administrator privileges can run a service dialog based on a playbook.

1. Browse to menu: **Services < Catalogs**.
2. In the **Catalog Items** accordion, click on the **All Catalog Items**.
3. Click **Configuration**, then (**Add a New Catalog Item**).
4. Select **Ansible Playbook** from the **Catalog Item Type** drop-down list.
5. Type a **Name** and **Description** for the new service catalog item.
6. Click **Display in Catalog**.
7. Select the appropriate **Catalog** from the drop-down list.
8. In the **Provisioning** tab, set parameters for your catalog item to use by configuring a Playbook to back your service item:
    1. Choose a **Repository** from the drop-down list.
    2. Select the **Ansible Playbook** to use.
    3. Assign the appropriate **Machine Credentials** from the drop-down list.
    4. Add **Cloud** or **Network Credentials** from the drop-down lists.
    5. Choose the **Host** against which to run the service item.
    6. Set the **Max TTL** in minutes. The Time To Live (TTL) field allows you to set the maximum execution time for the playbook to run.
    7. Use the **Escalate Privilege** toggle switch to enable user privilege escalation if called for in credentials during the playbook run.
    8. Choose a **Verbosity** value to set the debug level for playbook execution.
    9. Add key value pairs for **Variables** and their corresponding **Default Values**.
    10. In the **Dialog** options, choose an existing dialog from the **Use Existing** drop-down list or select **Create New** to add a new dialog.
9. In the **Retirement** tab, set parameters for your catalog item to use by selecting values for the following:

1. Choose a **Repository** from the drop-down list.
2. Select the **Ansible Playbook** to use.
3. Assign the appropriate **Machine Credentials** from the drop-down list.
4. Add **Cloud** or **Network Credentials** from the drop-down lists.
5. Choose the **Host** against which to run the service item.
6. Set the **Max TTL** in minutes. The Time To Live (TTL) field allows you to set the maximum execution time for the playbook to run.
7. Use the **Escalate Privilege** toggle switch to enable user privilege escalation if called for in credentials during the playbook run.
8. Choose a **Verbosity** value to set the debug level for playbook execution.
9. Add key value pairs for **Variables** and their corresponding **Default Values**.
10. In the **Dialog** options, choose an existing dialog from the **Use Existing** drop-down list or select **Create New** to add a new dialog.

10. Click **Add**.

Creating an Ansible Tower Service Catalog Item

Create a service catalog item from an Ansible Tower template you can use to execute an Ansible Tower playbook in Infrastructure Automation.

You must first create the job or workflow template in Ansible Tower. The job or workflow templates are automatically discovered by Infrastructure Automation when refreshing your Ansible Tower provider's inventory.
First, create a catalog:

1. Browse to menu: **Services > Catalogs**.
2. Click **Configuration**, then ⊕ (**Add a New Catalog**)
3. Enter a **Name** and **Description** for the catalog.
4. Click **Add**.

Then, create an Ansible Tower service catalog item:

1. Browse to menu: **Automation < Ansible Tower < Explorer**, then click on the **Templates** accordion menu.
2. Click **Ansible Tower Templates** and select an Ansible Tower job or workflow template.
3. Click **Configuration**, then ⊕ (**Create Service Dialog from this Template**).
4. Enter a **Service Dialog Name** (for example, *ansible_tower_job*)and click **Save**.
5. Browse to menu: **Services > Catalogs**. Click **Catalog Items**.
6. Click **Configuration**, then ⊕ (**Add a New Catalog Item**) to create a new catalog item with the following details, at minimum:
   - For **Catalog Item type**, select **Ansible Tower**.
   - Enter a **Name** for the service catalog item.
   - Select **Display in Catalog**.
   - In **Catalog**, select the catalog you created previously.
   - In **Dialog**, select the service dialog you created previously (in this example, *ansible_tower_job*). To ask the user to enter extra information when running the task, **Service Dialog** must be selected. A dialog is required if **Display in Catalog** is chosen.
   - In **Provider**, select your Ansible Tower provider. This brings up the **Ansible Tower Template** option and configures the **Provisioning Entry Point State Machine** automatically.
   - Add configuration information for **Reconfigure Entry Point** and **Retirement Entry Point** as applicable.
   - Select your desired **Ansible Tower Template** from the list. Generally, this is the Ansible Tower job or workflow template previously used to create the service dialog.
7. Click **Add**. The catalog item you created will appear in the **All Service Catalog Items** list.

Creating an Amazon Service Catalog Item

Use the following procedure to create an Amazon catalog item. Once created, the catalog item and service dialog combine with all of the options in the provisioning dialog. Users can then order Red Hat Enterprise Linux instances from the **Service Catalog** in the Infrastructure Automation Service user interface.

1. Browse to menu: **Services > Catalogs**, then click on the **Catalog Items** accordion.
2. Click **Configuration**, then ⊕**Add a New Catalog Item**.
3. Select **Amazon** from the **Catalog Item Type** list.

4. Enter the basic details in the **Basic Info** tab:
    1. Enter a **Name** and **Description** for the new service catalog item.
    2. Select **Display in Catalog**.
    3. Select the appropriate catalog from the **Catalog** list.
    4. Select the appropriate service dialog from the **Dialog** list.
5. Click the **Request Info** tab to enter request details:
    1. On the **Catalog** tab, select your Amazon AWS image name from the **Name** list, and the number of instances from the **Count** list. The **VM Name** will be overwritten during the provisioning process, but you can enter it as *changeme* for now.
    2. On the **Properties** tab, select *T2 Micro* from the **Instance Type** list, and *Basic* or *Advanced* for **CloudWatch**. If you plan to access the instance, select a **Guest Access Key Pair**, too.
    3. On the **Customize** tab, set the **Root Password** under **Credentials**, then select the *Basic root pass template* as a script for cloud-init under **Customize Template**.
6. Click **Add**.

Creating an Azure Service Catalog Item

Use the following procedure to create an Azure catalog item.

1. Browse to menu: **Services > Catalogs**, then click on the **Catalog Items** accordion.
2. Click **Configuration**, then (**Add a New Catalog Item**).
3. Select **Azure** from the **Catalog Item Type** list.
4. Enter the basic details in the **Basic Info** tab:
    1. Enter a **Name** and **Description** for the new service catalog item.
    2. Select **Display in Catalog**.
    3. Select the appropriate catalog from the **Catalog** list.
    4. Select the appropriate service dialog from the **Dialog** list.
5. Click the **Request Info** tab to enter request details:
    1. On the **Catalog** tab, select your Azure image name from the **Name** list, and the number of instances from the **Count** list. The **VM Name** will be overwritten during the provisioning process, but you can enter it as *changeme* for now.
    2. Select appropriate **Environment** settings that are known to work for your Azure environment.
    3. On the **Customize** tab, set the **Username** and **Password** under **Credentials**, then select the appropriate script under **Customize Template**.
6. Click **Add**.

Creating an OpenShift Template Catalog Item

Note:

Before adding a new catalog item for container template provisioning, create a service dialog from a container template. See Creating a Service Dialog from Container Template for details.

Complete the following procedure to create an OpenShift Template catalog item.

1. Browse to menu: **Service > Catalogs**, then click on the **Catalog Items** accordion.
2. Click **Configuration**, then (**Add a New Catalog Item**).
3. Select **OpenShift Template** from the **Catalog Item Type** list.
4. Enter a **Name** and **Description** for the new service catalog item. Select **Display in Catalog**.
5. Select the appropriate catalog from the **Catalog** list.
6. From the **Dialog list**, select the service dialog you have created from a container template.
7. Select your provider from the **Provider** list.
8. Set the **Provisioning Entry Point**.
9. Click **Add**.

Creating an Orchestration Catalog Item

Use the following procedure to create an Orchestration catalog item.

1. Browse to menu: **Services > Catalogs** and select **Catalog Items** in the accordion menu.
2. Click **Configuration**, then click ➕ **Add a New Catalog Item**. The **Adding a new Service Catalog Item** window is displayed.
3. Select **Orchestration** from the **Catalog Item Type** list.
4. Enter the basic details in the **Basic Info**:
    1. Enter a **Name** and **Description** for the new service catalog item.
    2. Select **Display in Catalog** box.
    3. Select the appropriate catalog from the **Catalog** list.
    4. Select the appropriate dialog from the **Dialog** list.
    5. Select the **Orchestration Template** from the list.
5. Click **Add**.

Creating a VMware Content Library OVF Template Catalog Item

Use the following procedure to create a VMware Content Library OVF Template catalog item.

Note:

Before you create the catalog item, make sure you have added a VMware vCenter Provider, and create a Service Dialog. See Adding a VMware vCenter Provider and Creating a Service Dialog for more information.

1. Browse to menu: **Services > Catalogs** and select **Catalog Items** in the accordion menu.
2. Click **Configuration**, then click ➕ **Add a New Catalog Item**. The **Adding a new Service Catalog Item** window is displayed.
3. Select **VMware Content Library OVF Template** from the **Catalog Item Type** list.
4. Enter the values in the **Basic Info**:
    1. Enter a **Name** and **Description** for the new service catalog item.
    2. Select **Display in Catalog** box.
    3. Select the appropriate catalog from the **Catalog** list.
    4. Select the appropriate dialog from the **Dialog** list.
    5. Select the **OVF Template** that you want to use for the deployment.
    6. Enter a **VM Name**.
    7. Select the **Accept EULA** box.
    8. Select the **Resource Pool**.
    9. Optional: Select the **Folder** and the **Host**.
5. Click **Add**.

Provisioning a Service

1. Browse to menu: **Services > Catalogs**.
2. Click the **Service Catalogs** accordion, and select the service to provision.
3. Click **Order**.
4. On the **Order Service** screen, select the box for **Do you want to continue?** and click **Submit**.

The parameters are passed to the children based on the method tied to the choices made in the dialog.

Orchestration Stacks

Cloud orchestration is a service that allows you to create, update, and manage cloud resources and their software components as a single unit and then deploy them in an automated, repeatable way through a template. Templates use a human-readable syntax and can be defined in text files, thereby allowing users to check them into version control. Templates allow you to easily deploy and reconfigure infrastructure for applications within your cloud. A user can author the stack templates, or can upload them from other sources.

Infrastructure Automation supports adding Amazon CloudFormation, OpenStack Heat, Microsoft Azure, VNF, and VMware vApp template type, and provides the ability to:

- Inventory stacks and elements of each type into the Infrastructure Automation VMDB.
- Model the relationships of instances to their stacks, inclusive of the user interface. For example, selecting an instance within a region that is within a stack, the user interface shows this on the standard instance view.

- Model the stack and its elements in the user interface.

Note:

When importing a template into Infrastructure Automation, the selected elements are converted according to their type. For example, lists convert to list boxes, and single items convert to text boxes.

Creating an Orchestration Template

Complete the following procedure to add an orchestration template.

1. Navigate to **Services > Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Click **Configuration**, then click **Create a new Orchestration Template**.
3. Enter a **Name** and **Description** for your template.
4. Select the template type from the **Template Type** list. The default is Amazon CloudFormation.
5. Select **Draft** to create a draft template.
6. Add your template in the area below for the selected **Template Type**.
7. Click **Add**.

Editing Orchestration Templates

Complete the following procedure to edit orchestration templates.

1. Navigate to **Services > Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Select the orchestration template you want to edit from the **All Orchestration Templates** list.
3. Click **Configuration**, then click **Edit this Orchestration Template**.
4. Edit the template as needed.

   Note:

   You can only edit the name and description of a read-only template as there can be stacks associated with the template.
5. Click **Save**.

Copying Orchestration Templates

Complete the following procedure to copy an orchestration template to create a new template.

1. Navigate to **Services > Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Click **Configuration**, then click **Copy this Orchestration Template**.
3. Change the **Description** and the actual content of the template as required. Infrastructure Automation automatically prefixes *Copy of* to the old template **Name**.

   Note:

   To create a copy of an orchestration template into a new template, the old and new template content must differ.
4. Click **Add**.

Deleting Orchestration Templates

Complete the following procedure to delete orchestration templates.

1. Navigate to **Services > Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Select the orchestration template you want to delete from the **All Orchestration Templates** list.
3. Click **Configuration**, then click **Remove this Orchestration Template from Inventory**.
4. Click **OK**.

Note:

Read-only templates cannot be deleted.

Retirement

Retiring Virtual Machines

Retiring Virtual Machines and Instances

When a virtual machine or instance is no longer required, it can be retired. Once a virtual machine or instance reaches its retirement date, it is immediately shut down and not allowed to restart. If an attempt to restart is made, Infrastructure Automation will shut down the virtual machine or instance.

There are three built-in policies involved with retirement:

- If the virtual machine or instance reaches its retirement date, it will be stopped even if it is running.
- If a retired virtual machine or instance is requested to start through Infrastructure Automation, the virtual machine or instance will not be allowed to start.
- If a provider starts a retired virtual machine or instance outside of Infrastructure Automation, the virtual machine or instance will be stopped.

Infrastructure Automation provides a number of ways to retire a virtual machine or instance:

- By using the allocated buttons in the Infrastructure Automation console.
- When creating a provision request, a retirement date can be set up.

Using the Console to Retire a Virtual Machine

Through the Infrastructure Automation console, you can retire a virtual machine on a specific date or immediately.

Retiring a Virtual Machine Immediately

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Select the virtual machine or instance that you want to retire.
3. Click (**Lifecycle**), then (**Retire this VM/Instance**).

The virtual machine or instance is immediately stopped, and will be shut down if an attempt is made to restart it.

Setting a Retirement Date and Time for a Virtual Machine or Instance

You can schedule virtual machine retirement by specifying a date and time, or by selecting a relative time a number of months, weeks, days or hours ahead of the present time.

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Select the virtual machine or instance that you want to set a retirement date for.
3. Click (**Lifecycle**), then (**Set Retirement Dates**).
4. From **Enter Retirement Date as**, select **Specific Date and Time** or **Time Delay from Now** to schedule retirement.
    1. To choose a **Specific Date and Time**, click the **Retirement Date and Time** field to open the calendar.
        1. Select a retirement date using the calendar control.
        2. Click then select a retirement time (in UTC) using the arrows.
    2. To retire the virtual machine using a relative time, select **Time Delay from Now**.
        1. From **Time Delay**, specify a retirement time any number of months, weeks, days, or hours in the future using the arrows.
5. Select a **Retirement Warning** if desired.
6. Click **Save**.

The scheduled retirement date and time display in the virtual machine summary screen.

Removing a Retirement Date for a Virtual Machine or Instance

1. Navigate to **Compute > Infrastructure > Virtual Machines**.
2. Select the virtual machine or instance that you want to remove the retirement date from.
3. Click (**Lifecycle**), then click (**Set Retirement Date**).
4. Save a blank date to remove the retirement dates.

Setting Retirement in a Provision Request

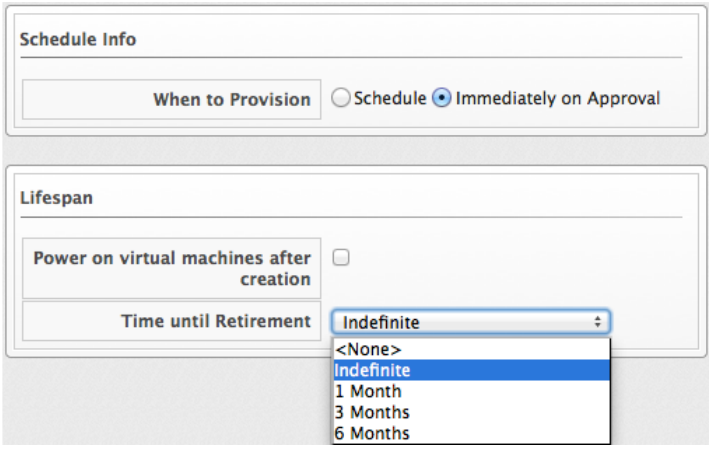
If you are using Infrastructure Automation to provision, you can set when you want retirement in the provision request. To see how to create a request, see Provisioning Requests. A warning email will be sent to the owner before the retirement.

Scheduling Retirement in a Provision Request

When provisioning a cloud instance or virtual machine, a multi-tabbed screen appears where you can set up your provision requests.

1. Click the **Schedule** tab to set when to provision your request and the lifespan of the virtual machine or instance.
2. In **Lifespan**, you can choose to power on the virtual machines or instances after creation and set the **Time until Retirement**. If you select the time until retirement, you will select **Retirement Warning** accordingly.

3. Click **Submit**.



Extending Retirement Dates

Infrastructure Automation **Automate** includes a method to extend the retirement of a virtual machine or instance by 14 days. This section describes how to create a button that invokes this method and how to edit the method to change the number of days.

Creating a Custom Button to Extend Retirement

1. Navigate to **Automation > Embedded Automate > Customization**.
2. Click the **Buttons** accordion.
3. From the **Object Types** tree, select **VM and Instance**.
4. Navigate to the button group to which you want to add this button. (If you do not have a button group, add one and then create the button.)
5. Click (**Configuration**), then (**Add a new Button**).
6. Type in a button text and button hover text, and select the image you want to use.
7. In **Object Details**, select **Request** from the **/System/Process/** dropdown. By default, the message is `create`. Do not change it.
8. In **Request**, type `vm_retire_extend`.
9. Click **Add**.

Changing the Number of Days to Extend Retirement

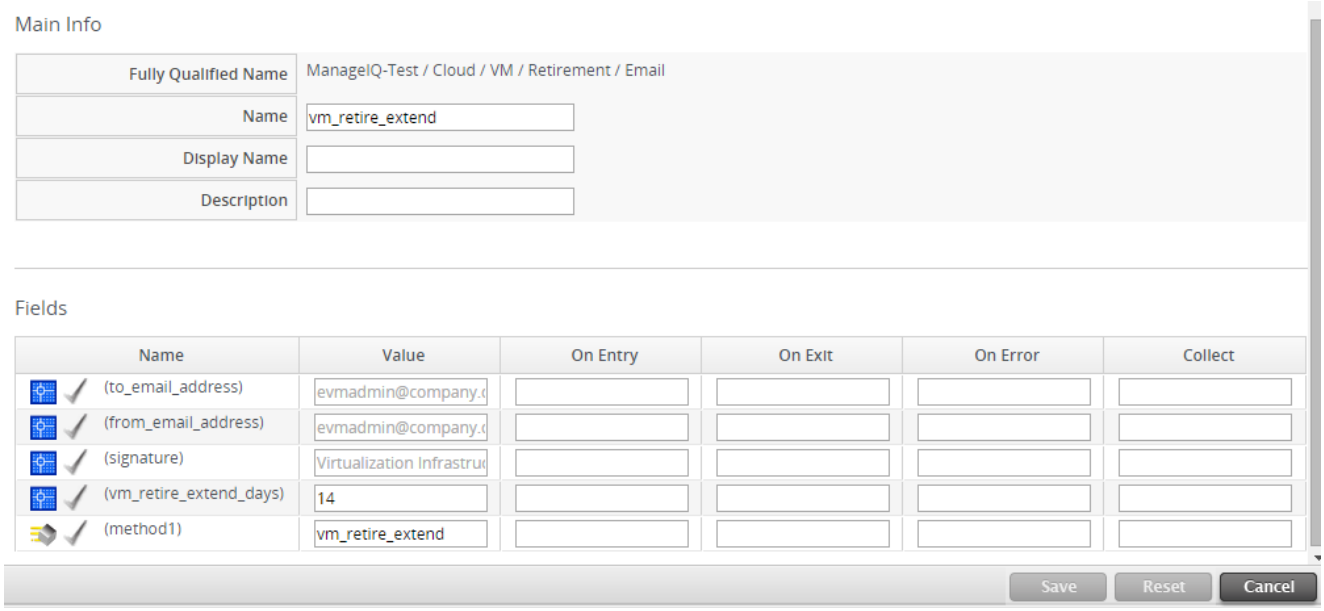
1. Navigate to **Automation > Embedded Automate > Explorer**.
2. Click **DOMAIN > Cloud > VM > Retirement > Email > System > Request**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.

   This example uses the **Cloud** Namespace, but you can also use the **Infrastructure** namespace.
3. Click (**Configuration**), then (**Edit this Instance**).
4. In the Value field, change the **vm_retire_extend_days** attribute to the new value.

5. Click **Save**.



- Creating a Service in for Virtual Machine Provisioning
  - Overview
  - Examining the Provisioning Dialog
  - Creating a Service Dialog
  - Creating a Catalog Item
  - Provisioning a Virtual Machine Using the Self Service User Interface

Creating a Service in for Virtual Machine Provisioning

Overview

Infrastructure Automation provides the ability to define services, which can include one or more virtual machines or instances, and can be deployed across hybrid environments. Services can be made available to users through the Self Service user interface, allowing users to order predefined IT services without IT operations getting involved.

The following three items are required to make a service available to users from the Infrastructure Automation Self Service user interface:

1. **Service Dialog**: A service dialog is where you allow users to configure virtual machine or instance options. A service dialog determines which options users get to change. You can choose the options presented to the user, from setting the service name to changing all provisioning dialog options.
2. **Service Catalog**: A service catalog is used to group service dialogs together.
3. **Service Catalog Item**: A service catalog item is the actual service which combines a service dialog with a provisioning dialog.

Examining the Provisioning Dialog

Understanding provisioning dialogs is important when working with services in Infrastructure Automation. Every provider in Infrastructure Automation comes with a sample provisioning dialog covering the options specific to that provider. The process begins with the provisioning dialog, which presents the basic configuration options for a virtual machine or instance. The options presented are dependent on the provider you are using. For instance, a cloud provider might have flavors of instances, whereas an infrastructure provider might allow you to set the memory size or number of CPUs on a virtual machine.

To see the sample provisioning dialog for Red Hat Virtualization:

1. Browse to menu: **Automation > Automate > Customization**.
2. Expand the **Provisioning Dialogs** accordion and select **Sample RedHat VM Provisioning Dialog** under **VM Provision**.

This is a textual representation of the dialog you will get when you provision a virtual machine.

Creating a Service Dialog

Using a service dialog you can determine which options users get to change. When users access a service, the majority of options available to them are preset and cannot be altered. A service dialog exposes some of those options to the user so that even if they are ordering a basic RHEL 7 machine, for example, they can at least choose the amount of memory, virtual CPUs, or other options available to the instance they order. In cases where certain fields must be unique, such as the name of virtual machines in Red Hat Virtualization, users must enter their own unique name for the virtual machine they choose or the operation will fail, so this field must be exposed.

Note:

A service requires an assigned service dialog, regardless of exposing details to user. A simple service dialog with a **Submit** and **Cancel** button is required at minimum.

A service dialog contains three components:

- One or more **Tabs**.
- Inside the tabs, one or more **Sections**. Note that in the previous method of creating a service dialog using the Infrastructure Automation user interface, **Sections** were referred to as **Boxes**.
- Inside the sections, one or more **Elements**. Elements are controls that accept input. Elements contain methods, like check boxes, drop-down lists or text fields, to fill in the options on the provisioning dialog.

The names of the elements must correspond to the options used in the provisioning dialog.
The following procedure outlines how to create a service dialog for your sample service:

1. Browse to menu: **Automation > Automate > Customization**.
2. Click the **Service Dialogs** accordion.
3. Click **Configuration**, and then (**Add a new Dialog**).
4. Enter basic information for your dialog under **General**:
    1. Enter `Provision RHV VM` in **Dialog's name**, and enter a description for the dialog in **Dialog's description**.
5. Add a new tab to the dialog:
    1. Click **Create Tab**. Then, click the icon on the new tab to edit tab information.
    2. Enter `General` in **Label**.
    3. Optional: Enter a description for the tab in **Description**.
    4. Click **Save**.
6. Add a new section to the tab:
    1. Click **Add Section**. Then, click the icon on the upper-right to edit section details.
    2. Enter `Service and VM Name` in **Label**.
    3. Optional: Enter a description for the section in **Description**.
    4. Click **Save**.
7. Add a **Text Box** element to the section for the service name:
    1. From the list of elements on the left, click the **Text Box** element, then drag-and-drop it inside the section. Then, click the icon next to the element to edit its field details.
    2. Enter `Service Name` in **Label**.
    3. Enter `service_name` in **Name**.
    4. Click **Options**, then select **String** from the **Value Type** list.
    5. Click **Save**.
8. Add a **Text Box** element to the section for the virtual machine name:
    1. From the list of elements on the left, click the **Text Box** element, then drag-and-drop it inside the section. Then, click the icon next to the element to edit its field details.
    2. Enter `VM Name` in **Label**.
    3. Enter `vm_name` in **Name**.
    4. Click **Options**, then select **String** from the **Value Type** list.
    5. Click **Save**.
9. Add a **Text Box** element to the section for the virtual machine description:
    1. From the list of elements on the left, click the **Text Box** element, then drag-and-drop it inside the section. Then, click the icon next to the element to edit its field details.
    2. Enter `VM Description` in **Label**.
    3. Enter `vm_description` in **Name**.
    4. Click **Options**, then select **String** from the **Value Type** list.

5. Click **Save**.

10. Add a new section to the tab:
    1. Click **Add Section**. Then, click the icon on the upper-right to edit section details.
    2. Enter `VM Characteristics` in **Label**.
    3. Optional: Enter a description for the section in **Description**.
    4. Click **Save**.
11. Add a **Dropdown** element to the section for the number of CPUs:
    1. From the list of elements on the left, click the **Dropdown** element, then drag-and-drop it inside the section. Then, click the icon next to the element to edit its field details.
    2. Enter `Number of CPUs` in **Label**.
    3. Enter `option_0_cores_per_socket` in **Name**.
    4. Click **Options**. Next to **Entries**, click + to add individual entries for the values `1`, `2`, and `4`, and their description.
    5. Click **Save**.
12. Add a **Dropdown** element to the section for the virtual machine memory:
    1. From the list of elements on the left, click the **Dropdown** element, then drag-and-drop it inside the section.
    2. Click the icon next to the element to edit its field details.
    3. Enter `VM Memory` in **Label**.
    4. Enter `option_0_vm_memory` in **Name**.
    5. Click **Options**. Next to **Entries**, click + to add individual entries for the values `1024`, `2048`, and `4096` and their description.
    6. Click **Save**.
13. Click **Save** to save the dialog.

Your service dialog is now added in the **Service Dialogs** accordion.

Creating a Catalog Item

Create a service catalog item to represent that service in a catalog.

1. Browse to menu: **Services > Catalog**.
2. Click the **Catalog Items** accordion.
3. Click **Configuration**, and then (**Add a new Catalog Item**).
4. Select **RHEV** from the **Catalog Item Type** list.
5. Enter the basic details in the Basic Info tab:
    1. Enter a name for the catalog item in **Name**.
    2. Enter a description for the catalog item in **Description**.
    3. Select **Display in Catalog**.
    4. Select the catalog you created from the **Catalog** list.
    5. Select the service dialog you created from the **Dialog** list.
6. Click the **Details** tab:
    1. Enter a description of the catalog item using HTML tags.
7. Click the **Request Info** tab to enter the request details. Enter the same data you did in the traditional provisioning dialog.
    1. In the **Catalog** tab, select the template on which to base the virtual machine from the **Selected VM** area.
    2. Enter `changeme` as the default name of the virtual machine in **VM Name**.
    3. In the Network tab, select **rhevm** from the **vLan** list.
8. Click **Add**.

Note:

In Infrastructure Automation, service catalog items can be more complex than a single virtual machine and are provisioned through a series of automation workflows. The service dialog you created will be used to gather information from the user requesting the virtual machine, hiding all the complexity of the details needed to actually deploy any of the components.

You can now provision a virtual machine through Red Hat Virtualization using the Self Service user interface.

Provisioning a Virtual Machine Using the Self Service User Interface

1. Log in to the Self Service User Interface at https://<hostname/ui/service>.
2. Click the **Service Catalog** tab.
3. Click the service you created.
4. In the **Service and VM Name**:
    1. Enter the service name in **Service Name**.
    2. Enter the virtual machine name in **VM Name**.
    3. Enter the virtual machine description in **VM Description**.
5. In **VM Characteristics**, enter **Number of CPUs** and **VM Memory**.
6. Click **Add to Shopping Cart** and access the shopping cart by clicking its icon on the upper-right corner of the screen.
7. Click **Order**.

You have ordered a new provisioning request. You can follow its progress and see additional details of the order request by selecting it from the **Orders** tab.

Also, see the Self Service User Interface Guide for more information about using the Self Service user interface.

Summary.

As part of the above sample service, you created a provisioning dialog that defines the options that can be set on a virtual machine. Then, you created a service dialog which allows us to expose certain options to be set by the user. For our example, the service name, virtual machine name, virtual machine description, and virtual machine characteristics are configurable. Additionally, you created a service catalog and finally a catalog item. The catalog item combines the service dialog with all of the options in the provisioning dialog. Lastly, you provisioned a Red Hat Virtualization virtual machine using the Self Service user interface.

- Service Provisioning Using an Orchestration Template
    - Overview
    - Creating an Orchestration Template
    - Creating a Service Dialog from an Orchestration Template
    - Creating an Orchestration Catalog Item
    - Creating a Catalog
    - Ordering a Service
    - Orchestration Stacks

Service Provisioning Using an Orchestration Template

Overview

Cloud orchestration is a service that allows you to create, update, and manage cloud resources and their software components as a single unit and then deploy them in an automated, repeatable way through a template. The deployed virtual machine instances and associated collection of resources are referred to as stack. Infrastructure Automation supports a number of orchestration template types, including Amazon CloudFormation and OpenStack Heat, that make deploying complex services easier in the cloud.

In the following example workflow, you will create an orchestration template, and add it as a catalog item to a service catalog. You can then create an orchestration stack from the template, and launch the stack from the catalog using a service dialog.

Creating an Orchestration Template

Complete the following procedure to create a new orchestration template.

1. Navigate to **Services > Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Click **Configuration**, then click **Create a new Orchestration Template**.
3. Enter a **Name** and **Description** for your template.
4. Select **Amazon CloudFormation** from the **Template Type** list.
5. Select **Draft** to create a draft template.
6. Add your template in the area below for the selected **Template Type**. You can author your own stack template, or you can copy and paste from an existing text file.

7. Click **Add**.

Next, create a service dialog from the orchestration template.

Creating a Service Dialog from an Orchestration Template

Complete the following procedure to create a service dialog based on the input parameters defined in the orchestration template.

1. Navigate to **Services > Catalogs** and click **Orchestration Templates** in the accordion menu.
2. Expand **All Orchestration Templates**, then click the orchestration template you created in Creating an Orchestration Template to create a service dialog from it.
3. Click **Configuration**, then click **Create Service Dialog from Orchestration Template**.
4. Enter a name for the service dialog in **Service Dialog Name**.
5. Click **Save**.

Creating an Orchestration Catalog Item

Complete the following procedure to create a new catalog item from the orchestration template you created in Creating an Orchestration Template.

1. Navigate to **Services > Catalogs** and select **Catalog Items** in the accordion menu.
2. Click **Configuration**, then click **Add a New Catalog Item**.
3. Select **Orchestration** from the **Catalog Item Type** list.
4. Enter the basic details in the **Basic Info**:
    1. Enter a **Name** and **Description** for the new service catalog item.
    2. Select **Display in Catalog** box.
    3. Select the appropriate catalog from the **Catalog** list.
    4. Select the appropriate dialog from the **Dialog** list.
    5. Select the orchestration template you created in Creating an Orchestration Template from the **Orchestration Template** list.
    6. Select **Amazon** from the **Provider** list.
5. Click **Add**.

Next, assign the catalog item to a new service catalog.

Creating a Catalog

Complete the following procedure to create a new catalog, and assign the catalog item you created in Creating an Orchestration Template.

1. Navigate to **Services > Catalogs** and select **Catalogs** in the accordion menu.
2. Click **Configuration**, then click **Add a New Catalog**.
3. In **Basic Info**, add **Name** and **Description** for the new catalog.
4. Under **Assign Catalog Items**, select the catalog item you created in Creating an Orchestration Template from the **Unassigned** box, then move it to the **Selected** box.
5. Click **Add**.

Ordering a Service

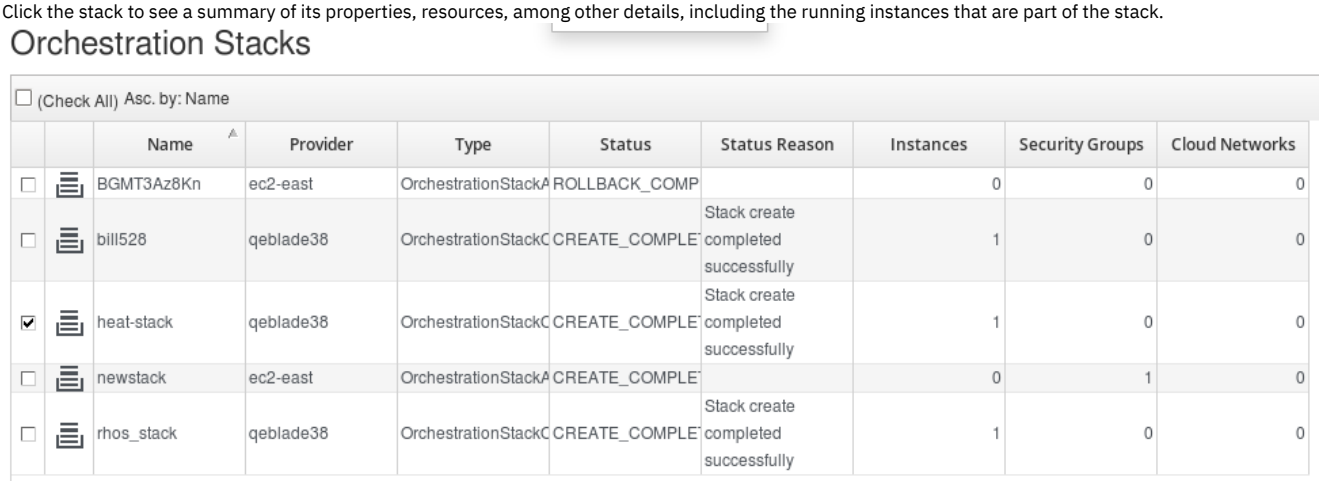
Complete the following procedure to order catalog items from the service catalog.

1. Navigate to **Services > Catalogs** and select **Service Catalogs** in the accordion menu.
2. From **All Services**, select the catalog and the services (catalog items) that you want to order from your catalog, then click **Order**. You will see the **Order Service** window with **Options** and **Parameters**.
3. Enter a name for the stack in **Stack Name**.
4. From the **On Failure** list, select what to do if stack creation fails. The default is **Rollback**.
5. Optional: Enter **Timeout** in minutes.
6. Set the remaining options as needed that will vary depending on the dialog.
7. Click **Submit**.

You have now submitted your service provisioning request. Once a request has been approved, the various stages of fulfillment will be processed. You can monitor the request status and other details in **Services > Requests**.

Orchestration Stacks

1. Once the status of the provisioning request in **Services > Requests** is **Finished**, click **Compute > Clouds > Stacks** to see the deployed stack.
2. Click the stack to see a summary of its properties, resources, among other details, including the running instances that are part of the stack.



You have now deployed instances and associated collection of resources (referred to as a stack) using an orchestration template.

Managing controls

Learn more about managing controls for Infrastructure Automation.

- Policies and Profiles Guide
- Monitoring, Alerts, and Reporting
- Chargeback

- Policies and Profiles Guide
  - Policies
    - Control Policies
      - Creating Control Policies
      - Editing Basic Information, Scope, and Notes for a Policy
      - Copying a Policy
      - Deleting a Policy
      - Creating a New Policy Condition
      - Editing Policy Condition Assignments
      - Editing Policy Event Assignments
      - Assigning an Action to an Event
    - Compliance Policies
      - Creating a Compliance Policy
      - Creating a Compliance Condition to Check Host File Contents
      - Checking for Compliance
        - Scheduling a Compliance Check
        - Checking a Virtual Machine for Compliance from the Summary Screen
        - Checking a Host for Compliance from the Summary Screen
        - Checking a Replicator for Compliance from the Summary Screen
        - Checking a Pod for Compliance from the Summary Screen
        - Checking a Container Node for Compliance from the Summary Screen
        - Checking a Container Image for Compliance from the Summary Screen
  - Conditions
    - Creating a Condition
    - Editing a Condition
    - Copying a Condition
    - Deleting a Condition
  - Actions
    - Custom Actions
      - Creating an Assign Profile to Analysis Task Action
      - Creating a Snapshot Action
      - Deleting Snapshots by Age
      - Evaluating an Alert

        - Creating an Inherit Tag Action
        - Creating a CPU Reconfigure Action
        - Creating a Memory Reconfigure Action
        - Creating a Remove Tag Action
        - Creating an Ansible Playbook Run Action
        - Creating an E-mail Action
        - Creating an SNMP Action
        - Creating a Set Custom Attribute Action
        - Creating a Tag Action
      - Editing an Action
      - Deleting an Action
    - Policy Profiles
      - Creating Policy Profiles
      - Deleting a Policy Profile
      - Simulating Policy
        - Simulating Policy Profiles on Virtual Machines
      - Assigning Policy Profiles
        - Assigning Policy Profiles to an Infrastructure Provider
        - Removing Policy Profiles from an Infrastructure Provider
        - Assigning Policy Profiles to a Cluster
        - Removing Policy Profiles from a Cluster
        - Assigning Policy Profiles to a Host
        - Removing Policy Profiles from a Host
        - Assigning Policy Profiles to a Virtual Machine
        - Removing Policy Profiles from a Virtual Machine
        - Assigning Policy Profiles to a Resource Pool
        - Removing Policy Profiles from a Resource Pool
        - Assigning Policy Profiles to a Cloud Provider
        - Removing Policy Profiles from a Cloud Provider
        - Assigning Policy Profiles to a Network Provider
        - Removing Policy Profiles from a Network Provider
        - Assigning Policy Profiles to a Container Provider
        - Removing Policy Profiles from a Container Provider
        - Assigning Policy Profiles to a Replicator
        - Removing Policy Profiles from a Replicator
        - Assigning Policy Profiles to a Pod
        - Removing Policy Profiles from a Pod
        - Assigning Policy Profiles to a Container Node
        - Removing Policy Profiles from a Container Node
        - Assigning Policy Profiles to a Container Image
        - Removing Policy Profiles from a Container Image
        - Assigning Policy Profiles to an Instance
        - Removing Policy Profiles from an Instance
      - Disabling a Policy in a Policy Profile
      - Viewing Policy Simulation - Resultant Set of Policy (RSOP)
      - Exporting and Importing Analysis Profiles
- Appendix
- Events

Policies and Profiles Guide

Policies

Policies are used to manage your virtual environment. There are two types of policies available: compliance and control. Compliance policies are used to harden your virtual infrastructure, making sure that your security requirements are adhered to. Control policies are used to check for a specific condition and perform an action based on the outcome. For example:

- Prevent virtual machines from running without an administrator account.
- Prevent virtual machines from starting if certain patches are not applied.
- Configure the behavior of a production virtual machine to only start if it is running on a production host.
- Force a SmartState Analysis when a host is added or removed from a cluster.

Control Policies

A control policy is a combination of an event, a condition, and an action. This combination provides management capabilities in your virtual environment.

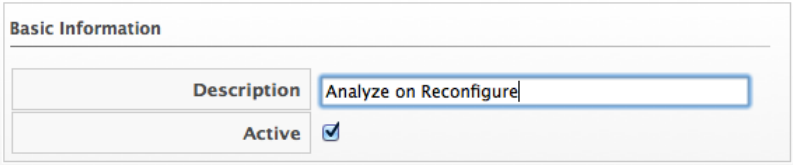
- An event is a trigger to check a condition.
- A condition is a test that is triggered by an event.
- An action is an execution that occurs if a condition is met.

Creating Control Policies

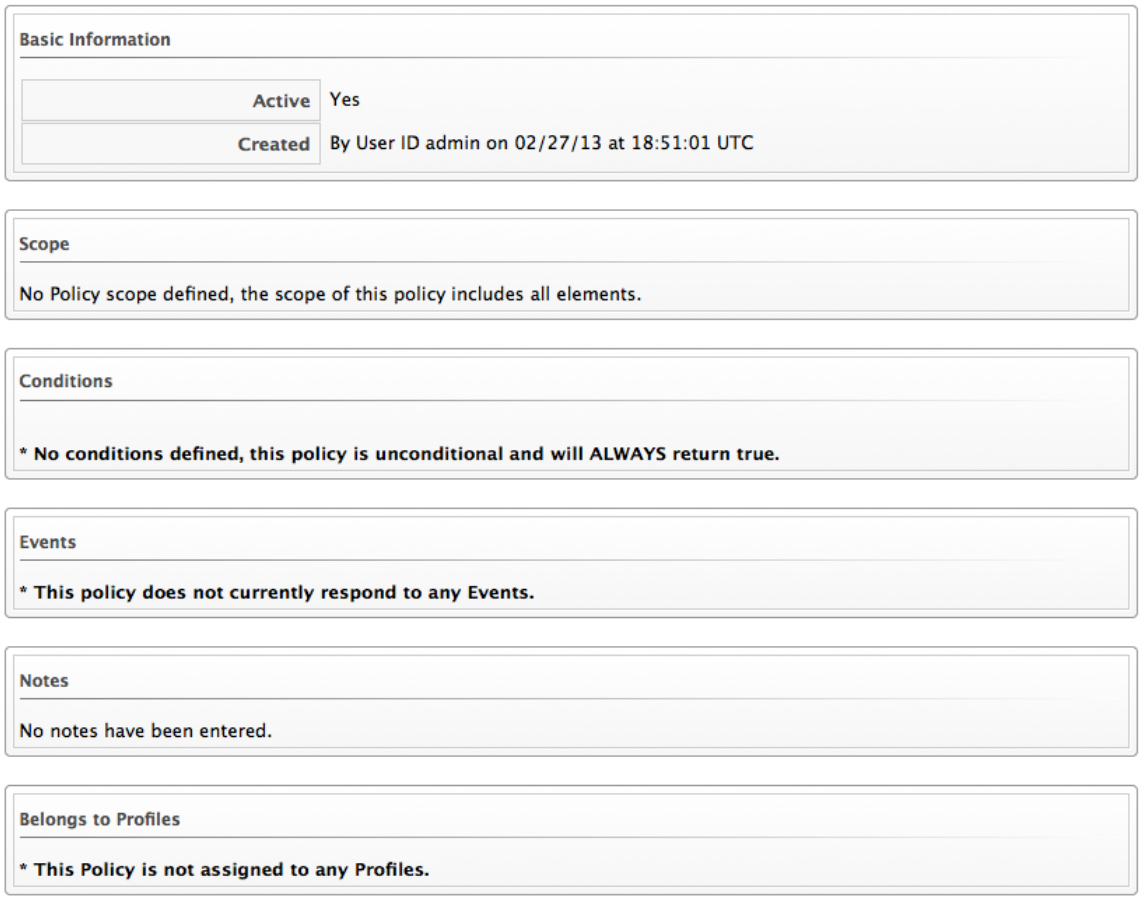
Create control policies by combining an event, a condition, and an action. Plan carefully the purpose of your policy before you create the policy. You can also use a scope expression that is tested immediately when the policy is triggered by an event. If the item is out of scope, then the policy does not continue on to the conditions, and none of the associated actions run.

This procedure describes how to create a control policy, its underlying conditions, and assign its events and actions in one process. Conditions and custom actions can be created separately as well. Those procedures are described in later sections in conditions and actions. Also, a description of all events is provided in events.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select **Control Policies**.
3. Select either **Host Control Policies** or **VM Control Policies** or **Replicator Control Policies** or **Pod Control Policies** or **Container Node Control Policies** or **Container Image Control Policies**.
4. Click **Configuration**, **Add a New Host / VM / Replicator / Pod / Node / Image Control Policy**.
5. Type in a **Description**.



6. Uncheck **Active** if you do not want this policy to be processed even when assigned to a resource.
7. You can enter a **Scope** here (You can also create a scope as part of a condition, or not use one at all). If the host or virtual machine is not included in the scope, no actions are run.
8. In the **Notes** area, add a detailed explanation of the policy.
9. Click **Add**. You are brought to the page where you add conditions and events to your new policy.

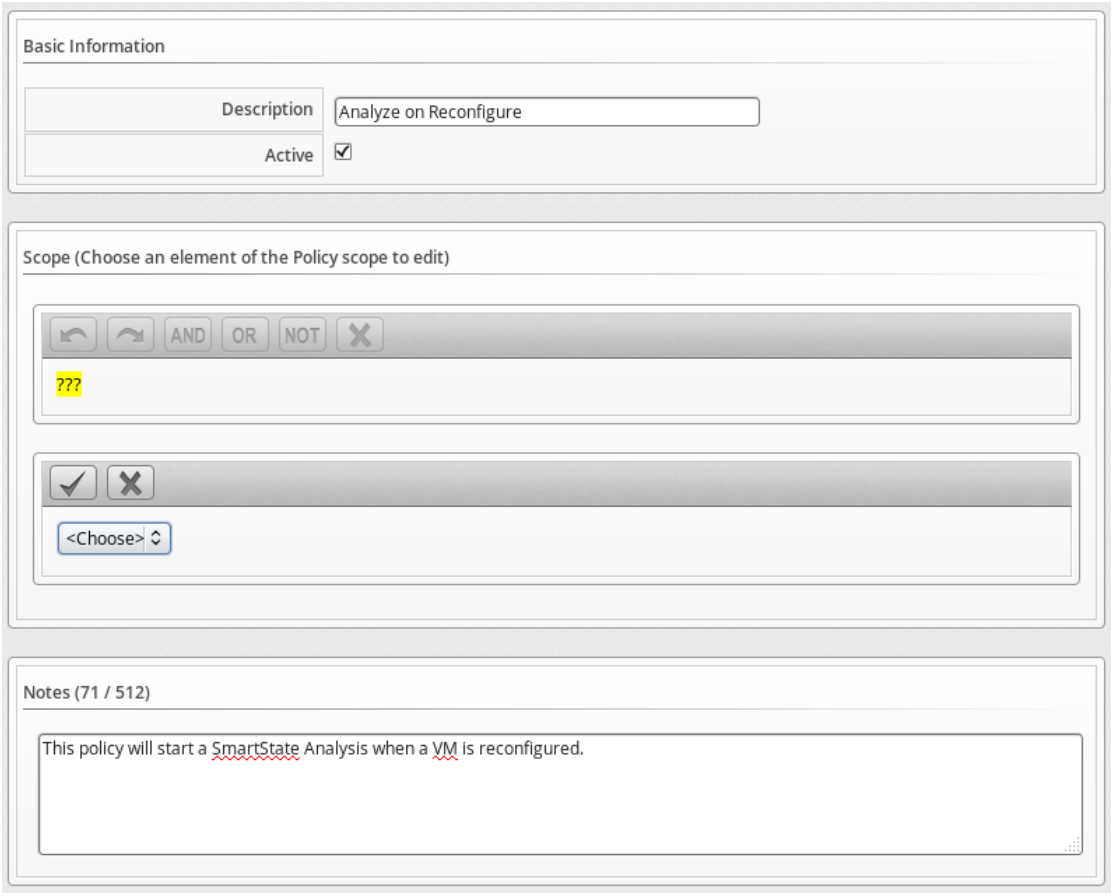


10. Click **Configuration** to associate conditions, events, and actions with the policy.

Editing Basic Information, Scope, and Notes for a Policy

As your enterprise's needs change, you can change the name of a policy or its scope. If the items that are being evaluated are out of scope, policy processing stops and no actions run.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select the policy to edit.
3. Click **Configuration**, **Edit Basic Info, Scope, and Notes**.
4. In the **Scope** area, create a general condition based on a simple attribute. Or, click an existing expression to edit it. Based on what you choose, different options appear. Configuring a **Scope** is optional for a policy.



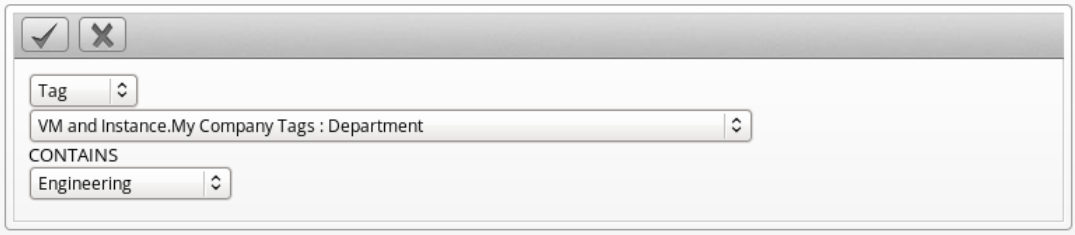
- Click **Field** to create criteria based on field values.



- Click **Count of** to create criteria based on the count of something, such as the number of snapshots for a virtual machine, or the number of virtual machines on a host.



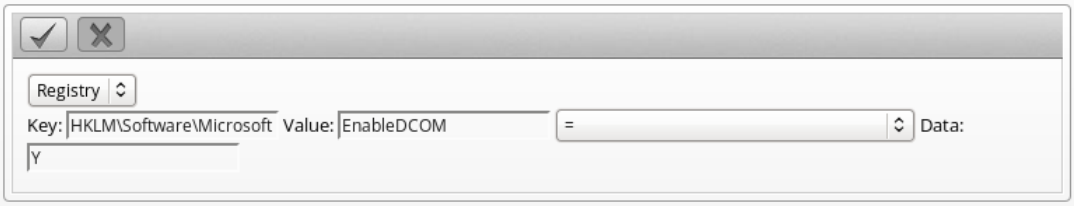
- Click **Tag** to create criteria based on tags that are assigned to your resources. For example, you can check the power state of a virtual machine or see whether it is tagged as production.



- Click **Find** to seek a particular value, and then check a property. For example, finding the **Admin** account and checking that it is enabled. Use the following check commands:
  - **Check Any**: The result is true if one or more of the find results satisfy the check condition.
  - **Check All**: All of the find results must match for a true result.
  - **Check Count**: If the result satisfies the expression in check count, the result is true.



- Click **Registry** to create criteria based on registry values. For example, you can check whether DCOM is enabled on a Windows System. Note that this applies only to Windows operating systems. Registry will only be available if you are editing a VM Control Policy.

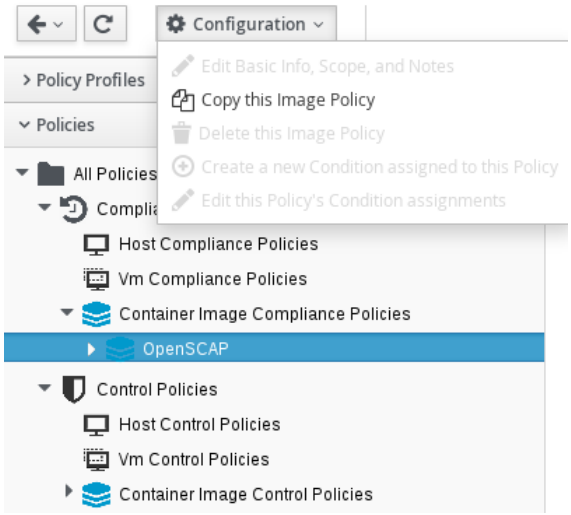


5. Click **Commit Expression Element Changes** to add the scope.
6. In the **Notes** area, make the required changes.
7. Click **Save**.

Copying a Policy

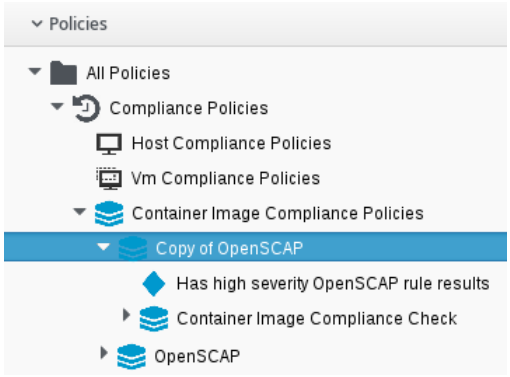
You can copy a policy if its contents are similar to a new one that you want to create, then change the condition or event associated with it. This enables you to make new policies efficiently.

1. Navigate to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select the policy that you want to copy.
3. Click **Configuration**, and an option to copy the policy appears; for example, **Copy this Image Policy**.



4. Click **OK** to confirm.

The new policy is created with a prefix of **Copy of** in its description, and it can be viewed in the Policy accordion.



Deleting a Policy

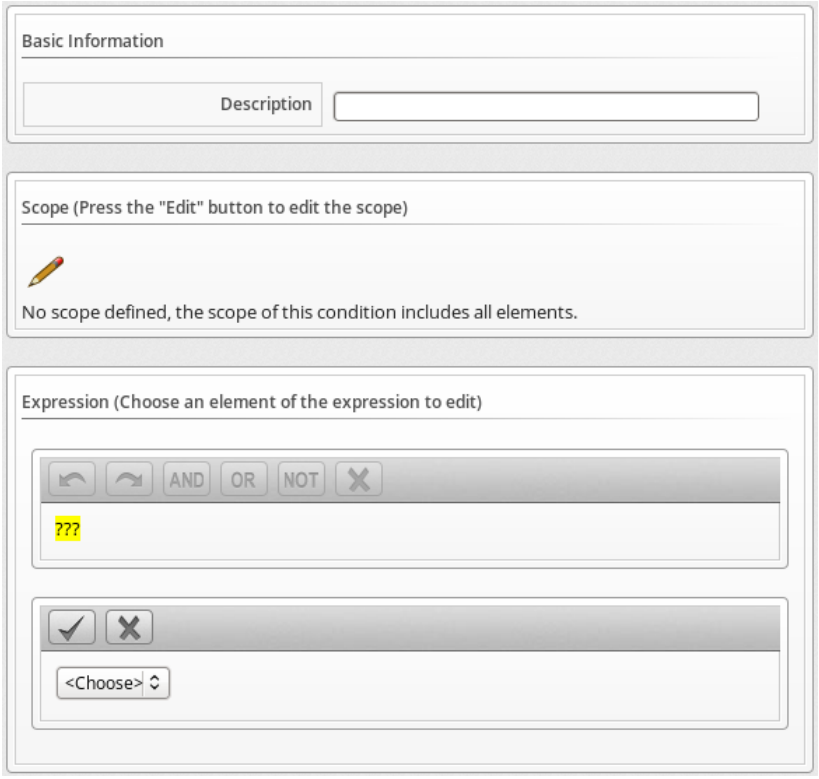
You can remove policies that you no longer need. You can remove only the policies that aren't assigned to a policy profile.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select the policy that you want to remove.
3. Click **Configuration**, **Delete this Host/VM and Instance/Replicator/Pod/Node/Image Policy**.
4. Click **OK** to confirm.

Creating a New Policy Condition

If you have not created a condition to use with this policy, you can create one directly from inside the policy. A condition can contain two elements: a scope, and an expression. The expression is mandatory, but the scope is optional. A scope is a general attribute that is quickly checked before evaluating a more complex expression. You can create a scope at either the policy or condition level.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select the policy that you want to create a new condition for.
3. Click **Configuration**, **Create a new Condition assigned to this Policy**.
4. Type in a **Description** for the condition. It must be unique to all the conditions.



5. Click **Edit this Scope** in the **Scope** area to create a general expression based on a simple attribute, such as operating system version. Based on what you choose, different options display. **Scope** is optional.

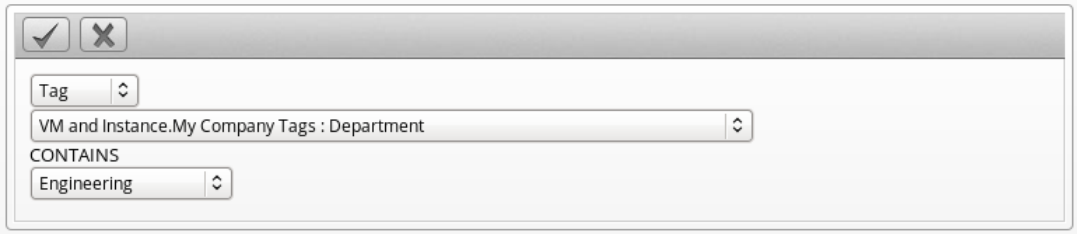
   - Click **Field** to create criteria based on field values.



   - Click **Count of** to create criteria based on the count of something, such as the number of snapshots for a virtual machine, or the number of virtual machines on a host.



   - Click **Tag** to create criteria based on tags that are assigned to your resources. For example, you can check the power state of a virtual machine or see whether it is tagged as production.

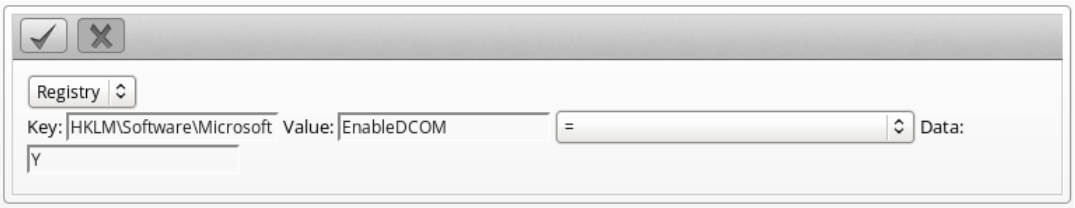


   - Click **Find** to seek a particular value, and then check a property. For example, finding the Admin account and checking that it is enabled. Use the following check commands:

- **Check Any**: The result is true if one or more of the find results satisfy the check condition.
- **Check All**: All of the find results must match for a true result.
- **Check Count**: If the result satisfies the expression in check count, the result is true.



- Click **Registry** to create criteria based on registry values. For example, you can check whether DCOM is enabled on a Windows System. This applies only to Windows operating systems. Registry is only available if you are creating a VM Control Policy.



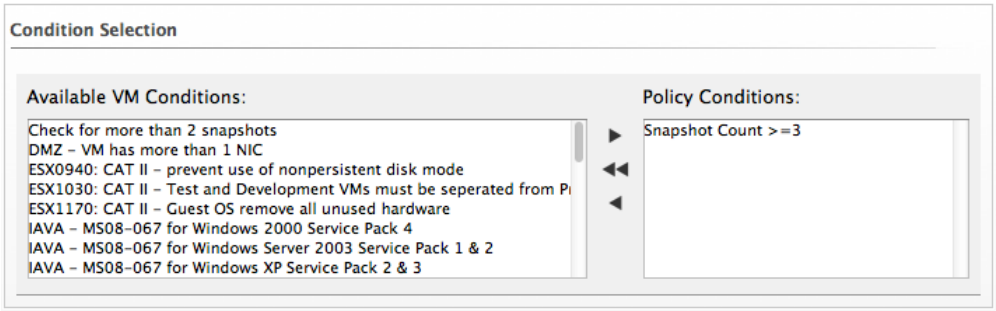
6. Click **Commit expression element changes** to add the scope.
7. Click **Edit this Expression** in the **Expression** area. Based on what you choose, options display according to the choices that are presented in the **Scope** area.
8. Click **Commit Expression Element Changes** to add the expression.
9. In **Notes**, type in a detailed explanation of the condition.
10. Click **Add**.

The condition is created and is assigned directly to the policy. **Note**: The condition can be assigned to other policies.

Editing Policy Condition Assignments

Use this procedure to use a condition that was created either separately or as part of another policy. You can also remove a condition from a policy that no longer applies.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select the policy that you want to assign conditions to.
3. Click **Configuration**, **Edit this Policy's Condition assignments**.
4. From the **Condition Selection** area, you can assign conditions to the policy, remove all conditions from the policy, or remove specific conditions from the policy.



- To add one or several conditions, select all the conditions that you want to apply from the **Available Conditions** box. Use `Ctrl` to add multiple conditions to a policy. Then, click **Move selected Conditions into this Policy**.
- Click **Remove all Conditions from this Policy** to unassign any conditions from this policy.
- To remove one or some conditions, select all the conditions that you want to remove from the **Policy Conditions** box. Use `Ctrl` to select multiple conditions. Then, click **Remove selected Conditions from this Policy**.

5. Click **Save**.

Editing Policy Event Assignments

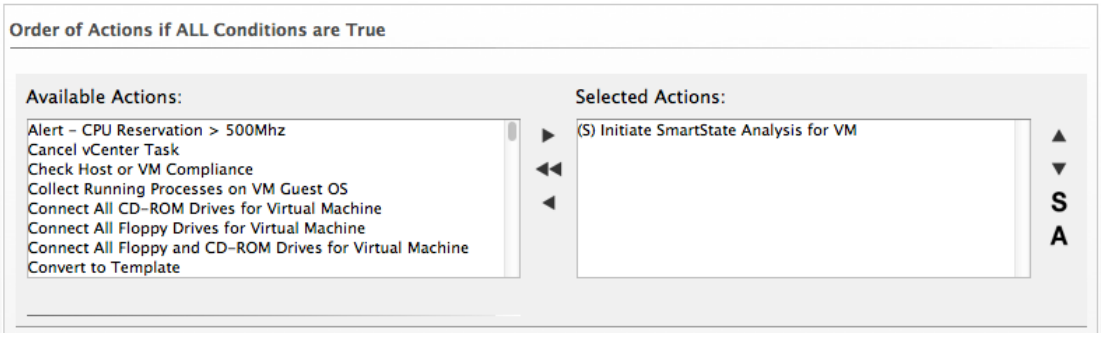
The policy evaluates its scopes and conditions when specified events occur in your virtual infrastructure. This procedure helps you to select those events and the actions that occur based on the evaluation of the scopes and conditions for the policy.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion and select the control policy that you want to assign events to.
3. Click **Configuration**, **Edit this Policy's Event assignments**.
4. Check all the events that you want to assign to this policy.
5. Click **Save**.

Assigning an Action to an Event

This procedure describes how to assign an action to an event.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, and select the policy that you want to assign actions to.
3. From the **Events** area, click the description of the event you want to assign an action to.
4. Click **Configuration**, **Edit Actions for this Policy Event**.
5. Select all the appropriate actions from the **Available Actions** box, inside the **Order of Actions if ALL Conditions are True**. These are the actions that take place if the resources meet the Condition of the Policy.



Note:

Each selected action can be executed synchronously or asynchronously; synchronous actions will not start until the previous synchronous action is completed, and asynchronous action allows the next action to start whether the first action has completed or not. Also, at least one Infrastructure Automation server in the Infrastructure Automation zone must have the notifier server role that is enabled for the trap to be sent.

6. Click the add button , then:
   - Click the action, then click **A** **Set selected Actions to Asynchronous** to make it asynchronous.
   - Click the action, then click **S** **Set selected Actions to Synchronous** to make it synchronous. If creating a synchronous action, use the up and down arrows to identify in what order you want the actions to run.
7. Select all the actions from the appropriate **Available Actions** box, inside of the **Order of Actions if ANY Conditions are False**. These are the actions that take place if the resources do not meet the condition of the policy.
8. Click **Save**.

Compliance Policies

Compliance policies are designed to secure your environment by checking conditions that you create. These conditions can include the same conditions that you would use in a control policy, and most of the procedures are the same. However, a compliance policy automatically assigns the mark as a compliant action when the entity type (virtual machine or host, for example) to which the policy applies passes all of the conditions. If any of the conditions are not met, then the virtual machine or host is marked as non-compliant. The compliance status is shown in the summary screen for the entity type and on the compare and drift screens.

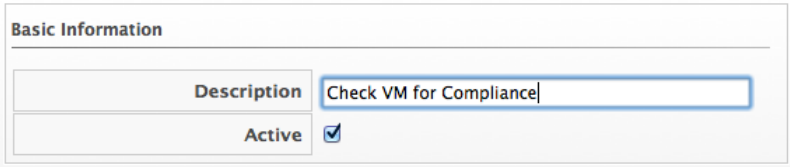
Creating a Compliance Policy

Create compliance policies by assigning or creating a condition. Infrastructure Automation automatically assigns the events and actions to the compliance policy as opposed to a control policy where you must define this yourself. The entity type (VM or host, for example) compliance check event is assigned to the compliance policy. A compliance policy runs the mark as compliant action when the virtual machine or host passes all of the conditions. If any of the conditions are not met, then the virtual machine or host is marked as non-compliant.

Carefully plan the purpose of your policy before creating it. You can also use a scope expression that is tested immediately when the compliance check event triggers the policy. If the item is out of scope, then the policy does not continue on to the conditions, and none of the associated actions run.

1. Browse to menu: **Control > Explorer**.

2. Click the **Policies** accordion, and select **Compliance Policies**.
3. Select either **Host Compliance Policies** or **VM Compliance Policies** or **Replicator Compliance Policies** or **Pod Compliance Policies** or **Container Node Compliance Policies** or **Container Image Compliance Policies**.
4. Click **Configuration**, **Add a new Compliance Policy**.
5. Type in a **Description** for the policy.



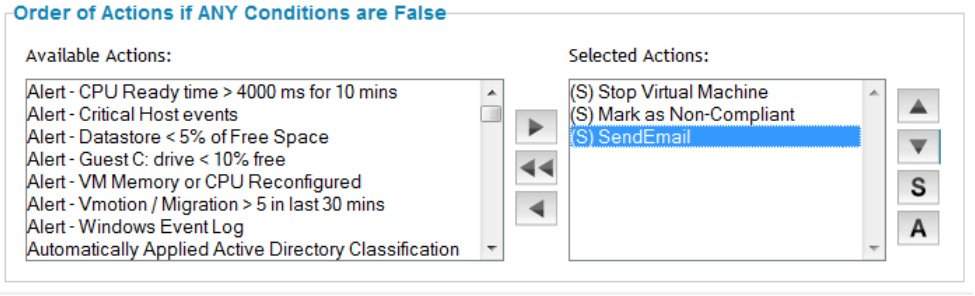
6. Uncheck **Active** if you do not want this policy to be processed even when assigned to a resource.
7. You can enter a scope here. (You can also create a scope as part of a condition, or not use one at all.) If the host or virtual machine is not included in the scope, no actions run.
8. In the **Notes** area, add a detailed explanation of the policy.
9. Click **Add**.

You must add one or several conditions:

- You can create a new condition by clicking **Configuration**, **Create a new Condition assigned to this Policy**, as described in Creating a New Policy Condition.
- You can use an existing condition by clicking **Configuration**, **Edit this Policy's Condition assignments**, as described in Editing Policy Condition Assignments.

By default, if any of the conditions are false, the virtual machine is marked as non-compliant. To add other actions, such as sending an email if the virtual machine fails the compliance test:

1. Click the **Compliance Check** event under the policy (exact name depends on entity type, for example **VM Compliance Check**.
2. Click **Configuration**, **Edit Actions for this Policy Event**.
3. Select **Stop Virtual Machine** and **Send Email** from the **Available Actions** area in **Order of Actions if ANY conditions are False**. (**Mark as Non-Compliant** should already be selected.)



4. Click **Move selected Actions into this Event**.
5. Click **Add**.

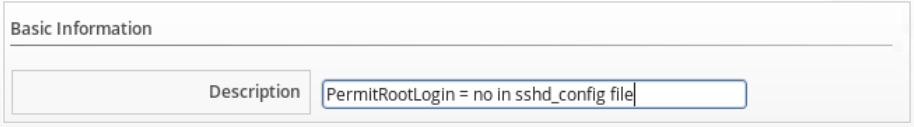
You can now make this part of a policy profile. After assigning the policy profile to the virtual machine, you can check it for its compliance status either on a schedule or on demand.

Creating a Compliance Condition to Check Host File Contents

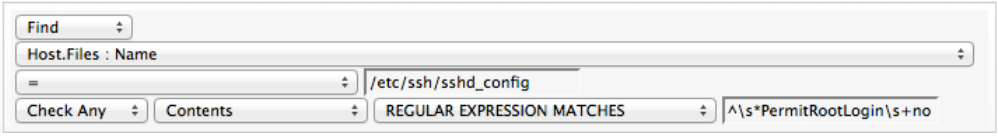
Infrastructure Automation Control provides the ability to create a compliance condition that checks file contents. Use this to be sure that internal operating system settings meet your security criteria. Regular expressions are used to create the search pattern. Test your regular expressions thoroughly before using them in a production environment.

Note that to search file contents you need to have collected the file using a host analysis profile.

1. Browse to menu: **Control > Explorer**.
2. Click the **Conditions** accordion, and select **Host Conditions**.
3. Click **Configuration**, **Add a New Host Condition**.
4. In **Basic Information**, type in a **Description** for the condition.



5. Editing the **Scope** area is not necessary for this procedure. Skip editing any **Scope** conditions.
6. If the **Expression** area is not automatically opened, click **Edit this Expression**, then edit the condition area to create a general condition based on a simple attribute. Based on what you choose, different options appear.
   - Click **Find**, then **Host.Files : Name**, and the parameters to select the file that you want to check.
   - Click **Check Any**, **Contents**, **Regular Expression Matches**, and type the expression. For example, if you want to make sure that permit root login is set to no, type `^\s*PermitRootLogin\s+no`.



7. Click **Commit expression element changes** to add the expression.
8. In **Notes** area, type in a detailed explanation of the condition.
9. Click **Add**.

Checking for Compliance

After you have created your compliance policies and assigned them to a policy profile, you can check compliance in two ways. You can either schedule the compliance check or perform the check directly from the summary screen.

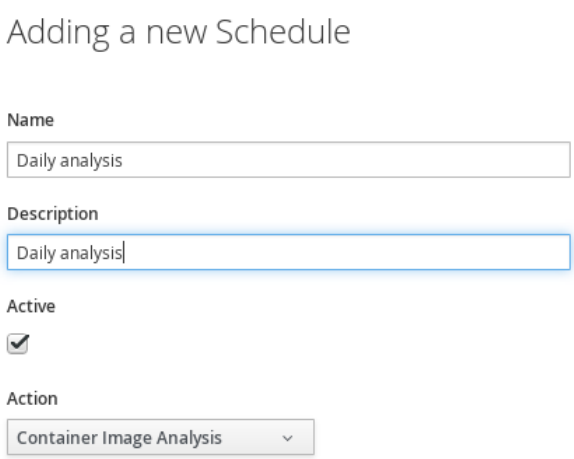
The compliance check runs all compliance policies that are assigned to the host or virtual machine. If the item fails any of the checks, it is marked as non-compliant in the item's summary screen.

Note:

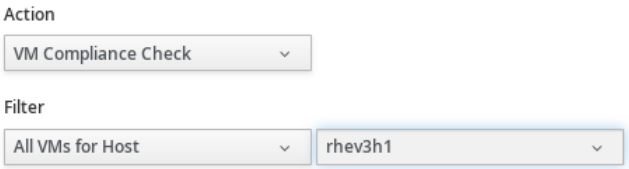
To schedule, you must have `EvmRole-administrator` access to the Infrastructure Automation server.

Scheduling a Compliance Check

1. Click **Configuration**.
2. Click the **Settings** accordion, and select **Schedules**.
3. Click **Configuration**, **Add a new Schedule**.
4. In the **Adding a new Schedule** area, type in a name and description for the schedule.



5. Select **Active** if you want to enable this scan.
6. From the **Action** dropdown, select the type of compliance check you want to schedule. Depending on the type of analysis you choose, you are presented with one of the following group boxes:
   - If you choose **VM Compliance Check**, you are presented with **VM Selection** where you can choose to check all VMs, all VMs for a specific provider, all VMs for a cluster, all VMs for a specific host, a single VM, or you can select VMs by using a global filter.

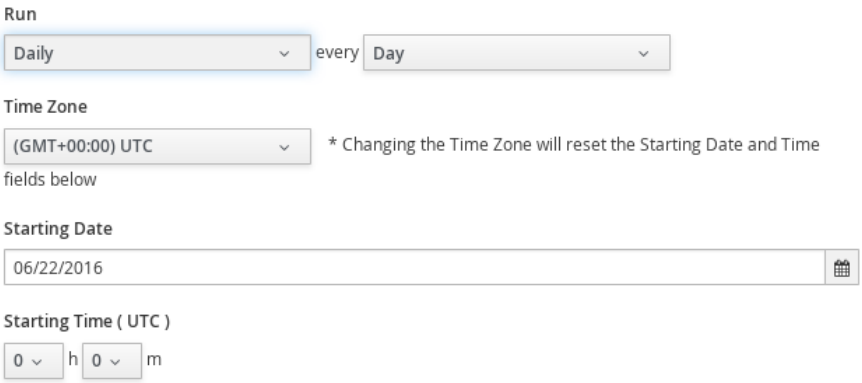


- If you choose **Host Compliance Check**, you are presented with **Host Selection** where you can choose to analyze all hosts, all hosts for a specific provider, all hosts for a cluster, a single host, or you can select hosts by using a global filter.
- If you choose **Container Image Compliance Check**, you are presented with **Image Selection** where you can choose to analyze all images, all images for a specific provider, or a single image.

Note:

You can schedule a host analysis only for connected virtual machines, not repository virtual machines that were discovered through that host. Since repository virtual machines do not retain a relationship with the host that discovered them, there is no current way to scan them through the scheduling feature. The host is shown because it might have connected virtual machines in the future when the schedule is set to run.

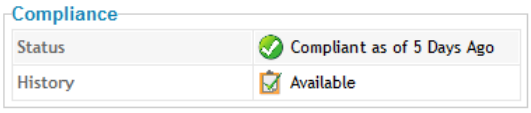
1. From the **Run** menu, select how often you want the analysis to run. Your options after that depend on which run option you choose.



   - Select **Once** to have the analysis run just one time.
   - Select **Hourly** to run the analysis hourly. You're prompted to select how many hours you want between each analysis.
   - Select **Daily** to run the analysis daily. You're prompted to select how many days you want between each analysis.
   - Select **Weekly** to run the analysis weekly. You're prompted to select how many weeks you want between each analysis.
   - Select **Monthly** to run the analysis monthly. You're prompted to select how many months you want between each analysis.
2. Select the time zone for the schedule.
3. Type or select a date to begin the schedule in **Starting Date**.
4. Select a starting time based on a 24-hour clock in the selected time zone.
5. Click **Add**.

Checking a Virtual Machine for Compliance from the Summary Screen

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**, select the virtual machine that you want to check for compliance.
2. Click **Policy**, and then **Check Compliance of Last Known Configuration**.
3. A confirmation message appears. Click **OK**.
4. To view the compliance history, click the virtual machine. Under **Compliance**, if **History** is **Available**, click it to see its compliance history.



Checking a Host for Compliance from the Summary Screen

1. Browse to menu: **Compute > Infrastructure > Hosts**, click the host that you want to check for compliance.
2. Click **Policy**, and then **Check Compliance of Last Known Configuration** or **Analyze then Check Compliance**.
3. To view the compliance history, click **Available** next to **History**.



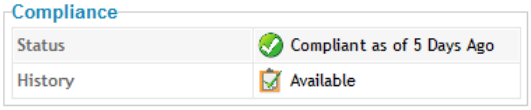
Checking a Replicator for Compliance from the Summary Screen

1. Browse to menu: **Compute > Containers > Replicators**, select the replicator that you want to check for compliance.
2. Click **Policy**, and then **Check Compliance of Last Known Configuration**.
3. A confirmation message appears. Click **OK**.
4. . To view the compliance history, click the replicator. Under **Compliance**, if **History** is **Available**, click to see its compliance history.



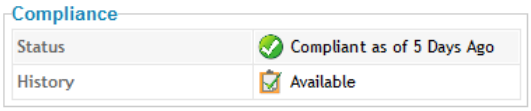
Checking a Pod for Compliance from the Summary Screen

1. Browse to menu: **Compute > Containers > Pods**, select the pod that you want to check for compliance.
2. Click **Policy**, and then **Check Compliance of Last Known Configuration**.
3. A confirmation message appears. Click **OK**.
4. To view the compliance history, click the pod. Under **Compliance**, if **History** is **Available**, click to see its compliance history.



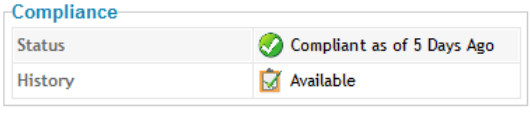
Checking a Container Node for Compliance from the Summary Screen

1. Browse to menu: Compute > Containers > Container Nodes**, click the node that you want to check for compliance.
2. Click **Policy**, and then **Check Compliance of Last Known Configuration**.
3. A confirmation message appears. Click **OK**.
4. To view the compliance history, click the node. Under **Compliance**, if **History** is **Available**, click to see its compliance history.



Checking a Container Image for Compliance from the Summary Screen

1. Browse to menu: **Compute > Infrastructure > Container Images**, select the container image that you want to check for compliance.
2. Click **Policy**, and then **Check Compliance of Last Known Configuration**.
3. A confirmation message appears. Click **OK**.
4. To view the compliance history, click the container image. Under **Compliance**, if **History** is **Available**, click to see its compliance history.



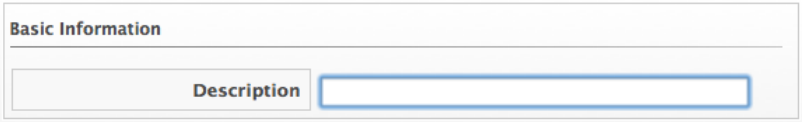
Conditions

Conditions are tests that are performed on attributes of virtual machines. A condition can contain two elements, a scope, and an expression. The expression is mandatory, but the scope is optional. A scope is a general attribute that is quickly checked before evaluating a more complex expression. For example, you might use a scope to check the operating system, and use an expression to check for a specific set of applications or security patches that only apply to the operating system referenced in the scope. If no conditions, scope, or expression, are defined for a policy, the policy is considered unconditional and returns a true value.

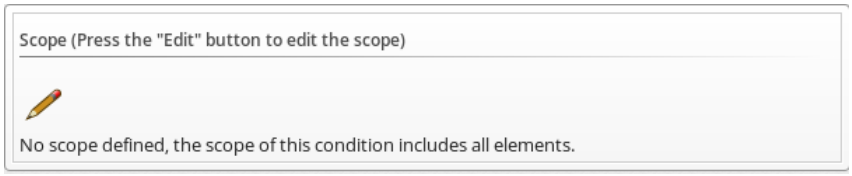
Creating a Condition

You can create a condition either from within a policy screen or by going directly to the expression editor in the Infrastructure Automation console. You need to define a description and an expression element. The expression element defines what criteria you want to use to test the condition.

1. Browse to menu: **Control > Explorer**.
2. Click the **Conditions** accordion, and select either **Host / Node Conditions** or **VM and Instance Conditions** or **Replicator Conditions** or **Pod** or **Node Conditions** or **Image Conditions**.
3. Click **Configuration**, then **Add a New Host / VM / Replicator / Pod / Node / Image Condition**.
4. Enter a **Description** for the condition.



5. Click **Edit this Scope** in the **Scope** area to create a general condition based on a simple attribute. Based on what you choose, different options appear. Creating a scope is optional.



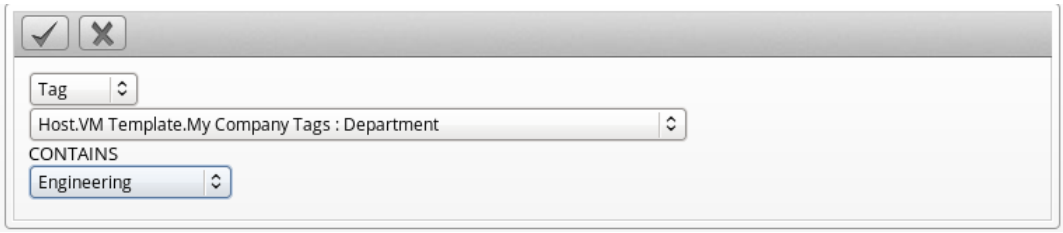
- Click **Field** to create criteria based on field values.



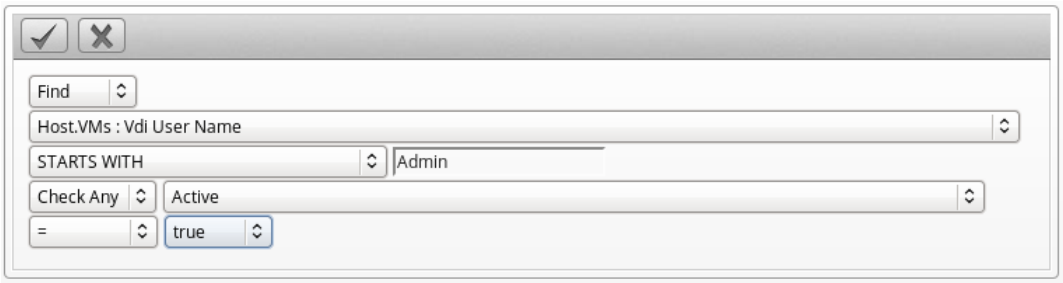
- Click **Count of** to create criteria based on the count of something, such as the number of network adapters on the host.



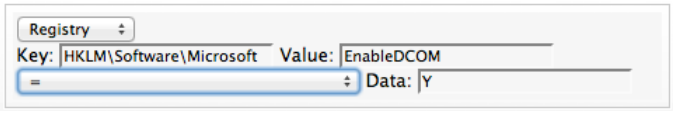
- Click **Tag** to create criteria based on tags that are assigned to your resources. For example, you can check the power state of a virtual machine or see whether it is tagged as production.



- Click **Find** to seek a particular value, and then check a property. For example, finding the Admin account and checking that it is enabled. Use the following check commands:
  - **Check Any**: The result is true if one or more of the find results satisfy the check condition.
  - **Check All**: All of the find results must match for a true result.
  - **Check Count**: If the result satisfies the expression in check count, the result is true.

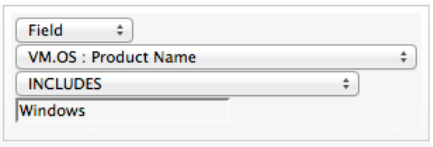


- Click **Registry** to create criteria based on registry values. For example, you can check whether DCOM is enabled on a Windows System. This applies only to Windows operating systems. Registry will only be available if you are creating a VM Condition.

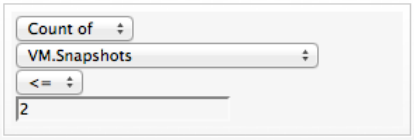


6. Click ✓ **Commit expression element changes** to add the scope.
7. Click **Edit this Expression** in the **Expression** area to create a general condition based on a simple attribute. Based on what you choose, different options appear.

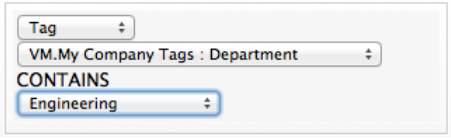
- Click **Field** to create criteria based on field values.



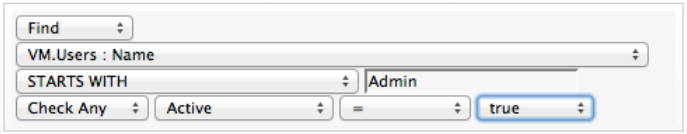
- Click **Count of** to create criteria based on the count of something, such as the number of snapshots for a virtual machine, or the number of virtual machines on a host.



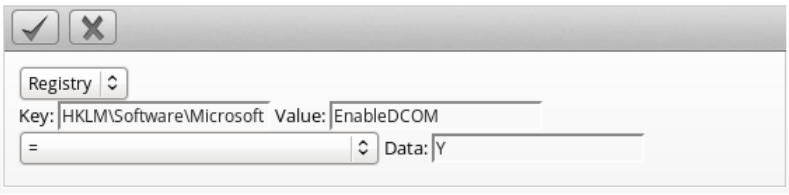
- Click **Tag** to create criteria based on tags assigned to your resources. For example, you can check the power state of a virtual machine or see whether it is tagged as production.



- Click **Find** to seek a particular value, and then check a property. For example, finding the Admin account and checking that it is enabled. Use the following check commands.
  - **Check Any**: The result is true if one or more of the find results satisfy the check condition.
  - **Check All**: All of the find results must match for a true result.
  - **Check Count**: If the result satisfies the expression in check count, the result is true.



- Click **Registry** to create criteria based on registry values. For example, you can check whether DCOM is enabled on a Windows System. This applies only to Windows operating systems.

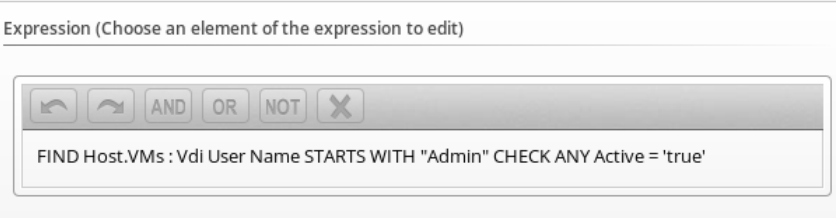


8. Click **Commit expression element changes** to add the expression.
9. In **Notes**, type in a detailed explanation of the condition.
10. Click **Add**.

Editing a Condition

Edit a condition to add more expressions to it or modify its properties. You can edit conditions that you created.

1. Browse to menu: **Control > Explorer**.
2. Click the **Conditions** accordion, and click the condition that you want to edit.
3. Click **Configuration**, **Edit this Condition**.
4. Click in either the **Scope** or **Expression** area, and click the part of the condition to edit.



5. Make the edits for the current expression.
   - Click **Commit expression element changes** to add the changes.
   - Click **Undo the previous change** to cancel the last action executed.
   - Click **Redo the previous change** to repeat the previous action executed.
   - Click AND **AND with a new expression element** to create a logical AND with a new expression element.
   - Click OR **OR with a new expression element** to create a logical OR with a new expression element.
   - Click NOT **Wrap this expression element with a NOT** to create a logical NOT on an expression element.
   - Click **Remove this expression element** to take out the current expression element.
6. When you complete all of the changes to the condition, click **Save**.

Copying a Condition

You can copy a condition to create a similar condition, then change the values that are associated with it. You can copy the sample conditions that are provided to customize them to your environment.

1. Browse to menu: **Control > Explorer**.
2. Click the **Conditions** accordion, and select the condition that you want to copy.
3. Click **Configuration**, **Copy this Condition to a new Condition**.
4. Make all the changes that you need for the new condition. The description must be unique to all conditions.
5. Click **Add**.

Deleting a Condition

Remove conditions that are no longer applicable. You can delete only the conditions that are not part of a policy. To be able to delete the condition, you must remove the policy first.

1. Browse to menu: **Control > Explorer**.
2. Click the **Conditions** accordion, and click the condition that you want to remove.
3. Click **Configuration**, **Delete this VM and Instance Condition**.
4. Click **OK** to confirm.

Actions

Actions are performed after the condition is evaluated. Control comes with a set of default actions that you can choose from. You can also create some of your own.

Action	Description
Cancel vCenter Task	Stop current vCenter Task. Due to limitations of vCenter, this applies only to cloning tasks.
Check Host or VM Compliance	Run compliance checks.
Collect Running Processes on VM Guest OS	Collect the list of running processes from the guest operating system.
Connect All CD-ROM Drives for Virtual Machine	Connect all the CD-ROM drives for the virtual machine.
Connect All Floppy Drives for Virtual Machine	Connect all the floppy drives for the virtual machine.
Connect All Floppy and CD-ROM Drives for Virtual Machine	Connect all of the floppy and CD-ROM drives for virtual machine.
Convert to Template	Convert this virtual machine to a template.
Delete all Snapshots	Remove all snapshots for a virtual machine.
Delete Most Recent Snapshot	Removes a virtual machine's most recent snapshot.
Delete VM from Disk	Remove the virtual machine from disk.
Disconnect All CD-ROM Drives for Virtual Machine	Disconnect all the CD-ROM drives for the virtual machine.
Disconnect All Floppy Drives for Virtual Machine	Disconnect all the floppy drives for the virtual machine.
Disconnect All Floppy and CD-ROM Drives for Virtual Machine	Disconnect all of the floppy and CD-ROM drives for virtual machine.
Execute an external script	Run an external script.
Generate Audit Event	Write an entry to the audit log and to the VMDB.
Generate log message	Write an entry to the Infrastructure Automation log.
Initiate SmartState Analysis for Host	Start a SmartState Analysis for a host.
Initiate SmartState Analysis for VM	Start a SmartState Analysis for a virtual machine.
Invoke a Custom Automation	For use with Infrastructure Automation automate. It enables you to run tasks and notifications automatically.
Mark as Non-Compliant	Used with compliance policies. Mark resource as non-compliant. (Compliance status is viewable in summary screens.)
Prevent current event from proceeding	Stop the current event from continuing.
Put Virtual Machine Guest OS in Standby	Put the virtual machines operating system in standby mode.
Raise Automation Event	Used with Infrastructure Automation automate.
Refresh data from vCenter	Perform a refresh of the vCenter.
Remove Virtual Machine from Inventory	Take the virtual machine out of inventory.
Retire Virtual Machine	Retire the virtual machine. (It remains in inventory, but cannot be started.)
Show EVM Event on Timeline	To show the EVM event on the timeline.
Shutdown Virtual Machines Guest OS	Shut down the virtual machine's operating system.
Start Virtual Machine	Power on the virtual machine.
Stop Virtual Machine	Power off the virtual machine.
Suspend Virtual Machine	Suspend the virtual machine.

Default Actions and Descriptions

Custom Actions

You can create a custom action by using the Infrastructure Automation console. Enter a description and action type. Procedures for each type of action are shown in the following sections. When you create a policy, you can associate actions with specific events.

Custom Action	Description
Assign Profile to Analysis Task	When initiating a Smart State Analysis event, you can assign a specific analysis profile.
Create a Snapshot	Creates a snapshot with a name that you provide.
Delete Snapshots by Age	Removes snapshots based on how old they are.
Evaluate Alerts	Checks for alerts. This is required for the alert to be delivered.
Inherit Parent Tags	Assigns tags from the parent cluster, host, datastore, or resource pool.
Invoke a Custom Automation	For use with Infrastructure Automation automate.
Reconfigure CPUs	Reconfigure the number of CPUs for a virtual machine to the number you specify.
Reconfigure Memory	Reconfigure the amount of memory for a virtual machine to the amount you specify.
Remove Tags	Removes tags from the resource.
Run Ansible Playbook	Run an Ansible playbook against an inventory selection.
Send an E-mail	Send an email to an address that you provide. This type of action can be used in an alert.
Send an SNMP trap	Send an SNMP (Simple Network Management Protocol) trap to the host you specify. This type of action can be used for an alert.
Set a Custom Attribute in vCenter	Set the value of a custom attribute in vCenter.
Tag	Assign a company tag that you specify to a virtual machine.

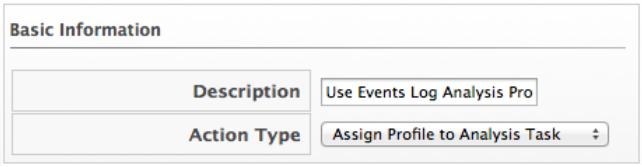
Custom Actions and Descriptions

Creating an Assign Profile to Analysis Task Action

Use this action for assigning specific analysis profiles to virtual machines. You must create an analysis profile before assigning it to an action. You can only assign this action to an analysis start event.

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.

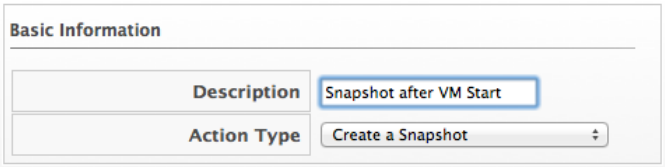
3. Type in a **Description** for the **Action Type**.



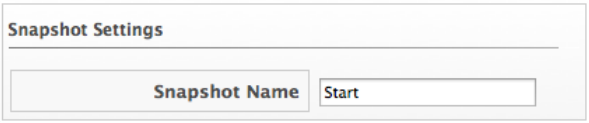
4. Select **Assign Profile to Analysis Task** from **Action Type**.
5. Select a profile from the **Analysis profiles**.
6. Click **Add**.

Creating a Snapshot Action

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



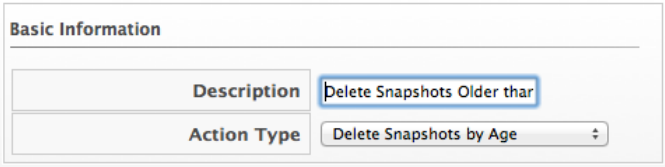
4. Select **Create a Snapshot** from **Action Type**.
5. Type in a **Snapshot Name**.



6. Click **Add** when you are finished.

Deleting Snapshots by Age

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



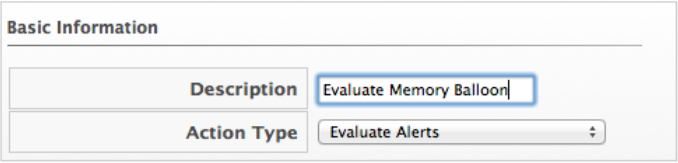
4. Select **Delete Snapshots by Age** from **Action Type**.
5. Select the age of snapshots to delete.



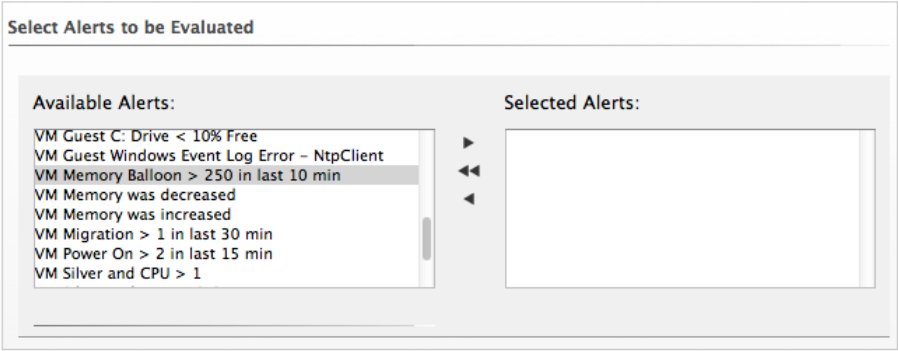
6. Click **Add**.

Evaluating an Alert

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



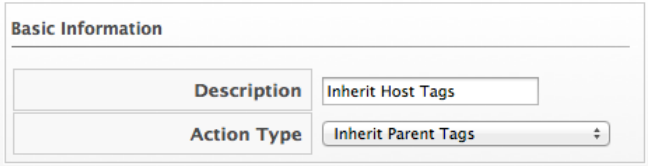
4. Select **Evaluate Alerts** from **Action Type**.
5. Select the alerts to be evaluated and click (Move selected Alerts into this Action). Use the `Ctrl` key to select multiple alerts.



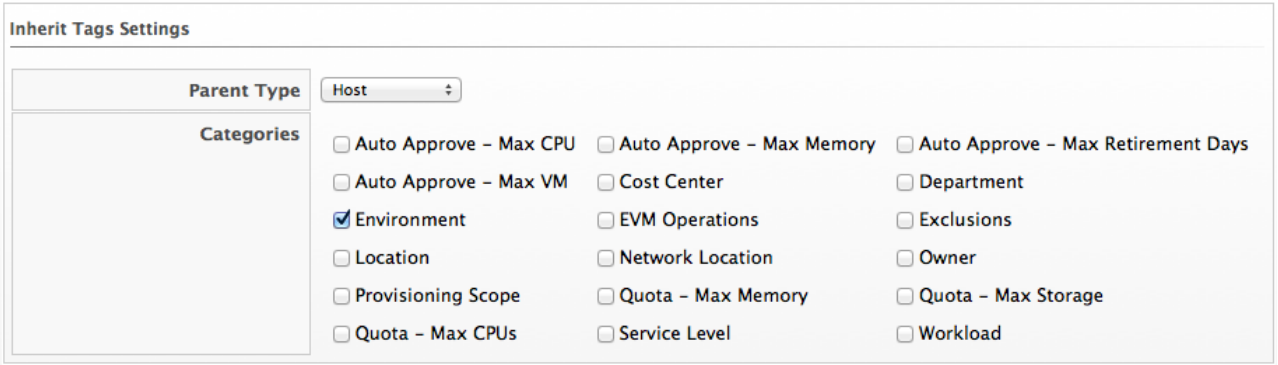
6. Click **Add**.

Creating an Inherit Tag Action

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, and click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



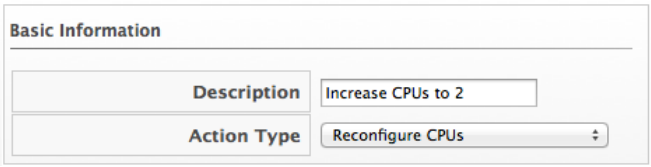
4. Select **Inherit Parent Tag** from **Action Type**.
5. Select the type of parent item to inherit from in **Parent Type**.
6. Check all categories that you want inherited.



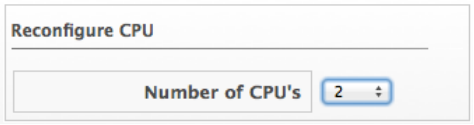
7. Click **Add**.

Creating a CPU Reconfigure Action

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



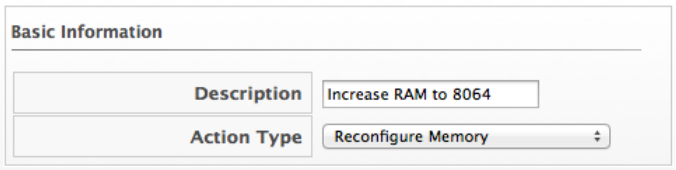
4. Select **Reconfigure CPUs** from **Action Type**.
5. Select a number from **Number of CPUs**.



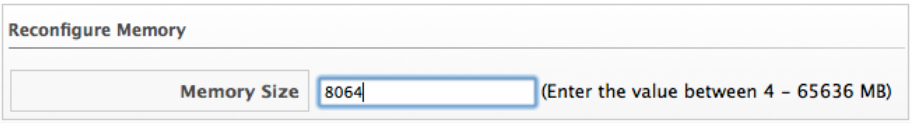
6. Click **Add**.

Creating a Memory Reconfigure Action

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



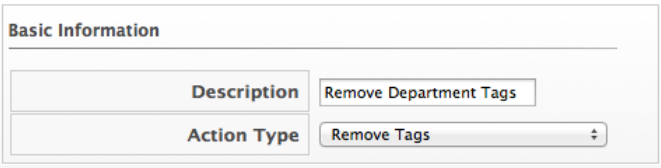
4. Select **Reconfigure Memory** from **Action Type**.
5. Type in a new value for **Memory Size**.



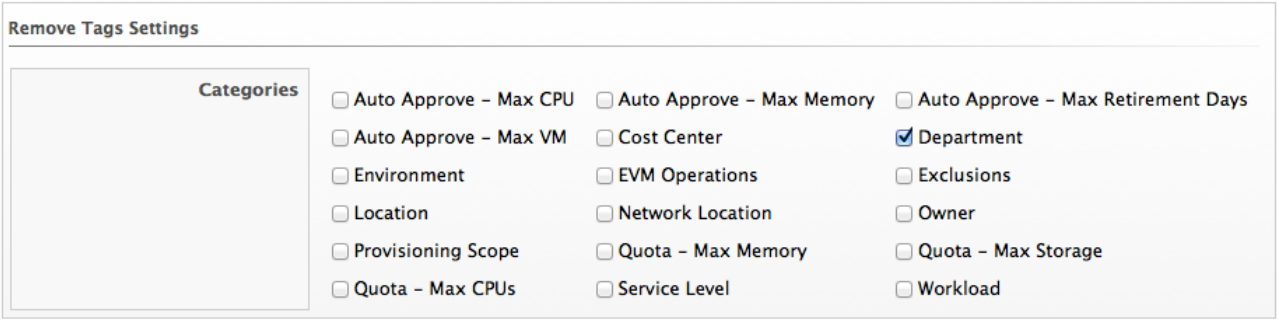
6. Click **Add**.

Creating a Remove Tag Action

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



4. Select **Remove Tags** from **Action Type**.
5. Check the category of tags you want to remove.



6. Click **Add**.

Creating an Ansible Playbook Run Action

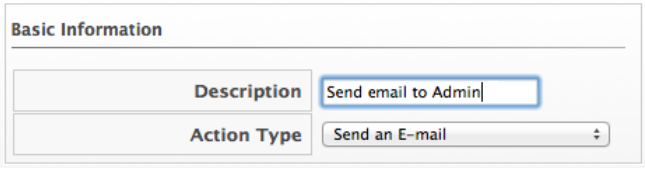
Use this action to run an Ansible Playbook against your inventory. You must first sync a playbook repository and add an Ansible Playbook service catalog item.

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, and click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.
4. Select **Run Ansible Playbook** from **Action Type**.
5. Select the playbook catalog item to run from **Playbook Catalog Item**.
6. Check the inventory against which you run the Ansible playbook.
   - If **Specific Hosts** is selected, provide the IP or DNS names.
7. Click **Add**.

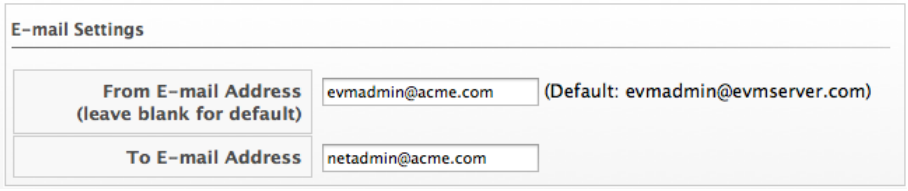
Creating an E-mail Action

To send emails from the Infrastructure Automation server, you must have the notifier server role enabled and have defined settings for SMTP email.

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



4. Select **Send an E-mail** from **Action Type**.
5. Type in a **From E-mail Address** and **To E-mail Address**.



6. Click **Add**.

Creating an SNMP Action

To send SNMP traps from the Infrastructure Automation server, you must have the **`Notifier`** server role and the SNMP daemons enabled.

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Enter a **Description** for the action.
4. Select **Send an SNMP** Trap from **Action Type**.
5. Type in the IP for the host to send the trap to, select the version of SNMP that you are using, and type in the Trap Object ID. Type in multiple hosts if you require the trap sent to multiple SNMP hosts.
   - If using SNMP V1, you are prompted for a Trap Number. Type 1, 2, or 3, based on the appropriate Suffix Number from table below.
   - If using SNMP V2, you are prompted for a Trap Object ID. Type info, warning, or critical, based on the table below.

Object ID	Suffix Number Added to PEN	PEN with the Suffix Added
info	1	1.3.6.1.4.1.33482.1
warn, warning	2	1.3.6.1.4.1.33482.2
crit, critical, error	3	1.3.6.1.4.1.33482.3

Trap Object ID and Suffix Number

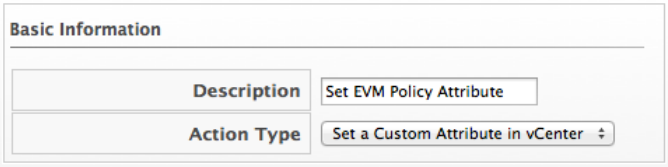
6. Type in the variables that you require in your message.
7. Click **Add**.

Note: When adding an SNMP action, be sure to set it as asynchronous.

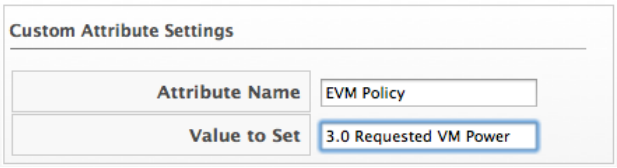
Creating a Set Custom Attribute Action

The custom attribute must already exist in vCenter. See vCenter documentation for instructions. In this example, an attribute called Infrastructure Automation policy already exists.

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



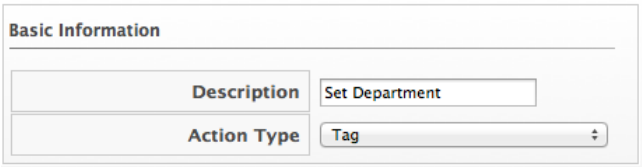
4. Select **Set a Custom Attribute in vCenter** from **Action Type**.
5. Type in the **Attribute Name** and **Value to Set**.



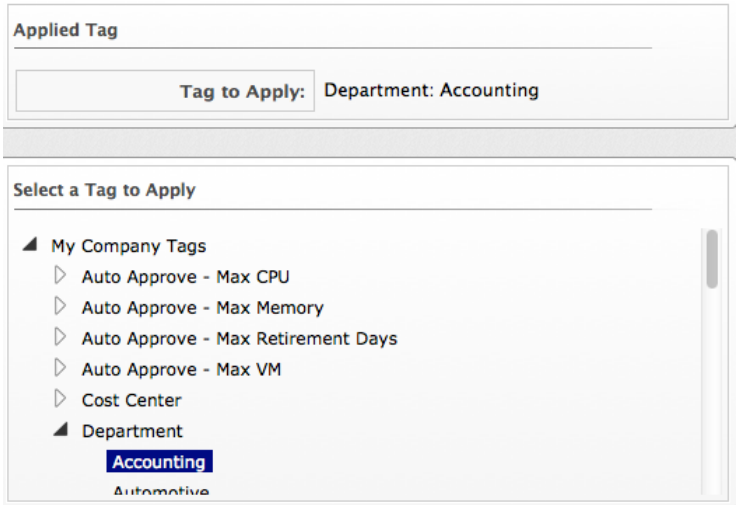
6. Click **Add**.

Creating a Tag Action

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, then click **Configuration**, **Add a new Action**.
3. Type in a **Description** for the action.



4. Select **Tag** from **Action Type**.
5. Click on the appropriate tag to apply from the list provided.



6. Click **Add**.

Editing an Action

Edit an action to modify its properties. You can't edit any of the default actions that are supplied with Infrastructure Automation. Only actions that you create can be changed.

Note: When you view an action, you can see what policies it is assigned to.

1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, and then click the action that you need to edit.
3. Click **Configuration**, **Edit this Action** on the detail view of the action.
4. Make any required changes.
5. Click **Save**.

The action is modified and can be added to a policy. If the action is already party of a policy, the policy is automatically updated.

Deleting an Action

Delete unused actions to keep your environment uncluttered. You can't delete default actions or actions that are currently assigned to a policy. The delete button is unavailable if the action is in use.

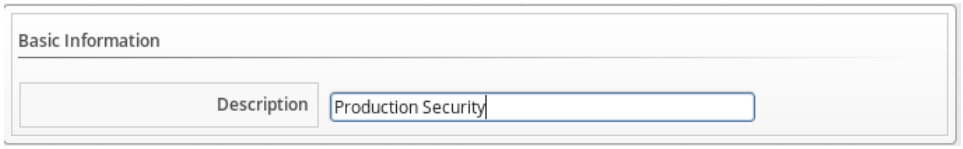
1. Browse to menu: **Control > Explorer**.
2. Click the **Actions** accordion, click the action that you need to remove.
3. Click **Configuration**, **Delete this Action** on the detail view of the tree.
4. Click **OK** to confirm.

Policy Profiles

Policy profiles are groups of policies that you can assign wholesale to virtual machines, providers, clusters, hosts, resource pools, replicators, pods, container nodes, and container images. Policy profiles provide a framework for easily managing and assigning different levels of security, across various types of cloud resources.

Creating Policy Profiles

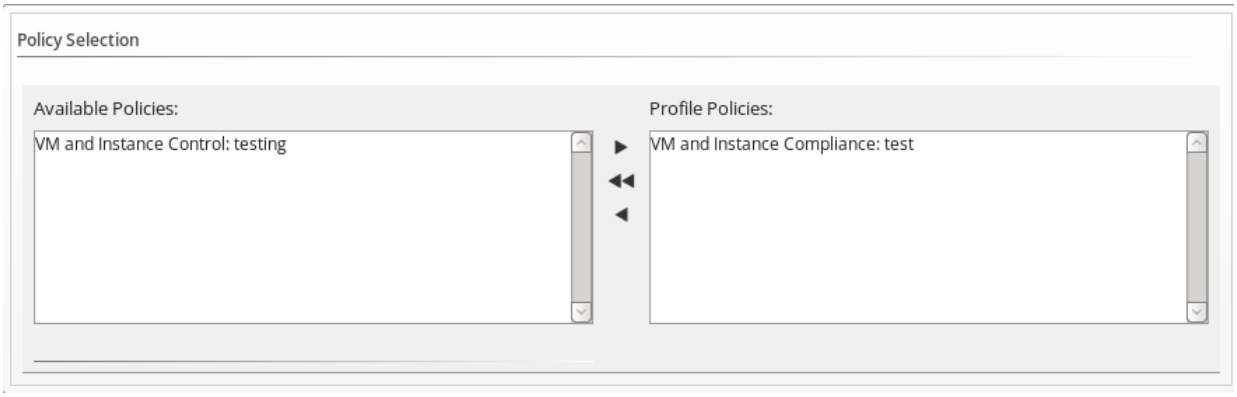
1. Browse to menu: **Control > Explorer**.
2. Click on the **Policy Profiles** accordion, then click **Configuration**, then (**Add a New Policy Profile**).
3. In the **Basic Information** area, type in a unique description for the policy profile.



4. From **Available Policies** in the **Policy Selection** area select all the policies you need to apply to this policy profile. Use the `Ctrl` key to select multiple policies.



5. Click to add the **Policies**.



6. Add to the **Notes** area if required.
7. Click **Add**.

The policy profile is added. You can now assign the policy profile to providers, hosts, and repositories. In addition, you can verify that the virtual machine complies with the policy profile by using the **Resultant Set of Policy** feature.

Deleting a Policy Profile

Remove policy profiles that you no longer need. This doesn't remove the policies that are associated with the policy profile.

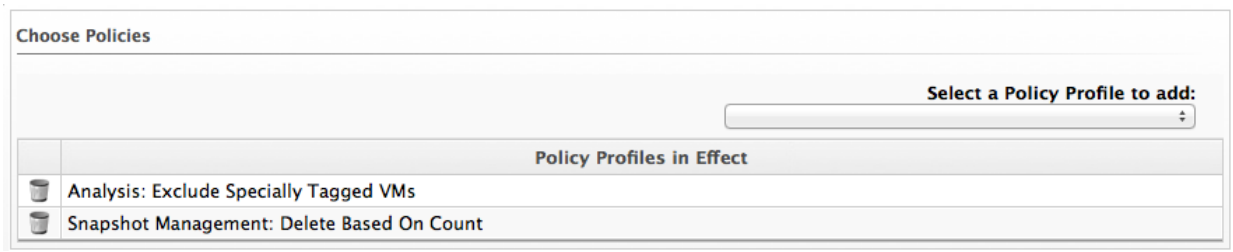
1. Browse to menu: **Control > Explorer**.
2. Click the **Policy Profile** accordion, and then click the policy profile that you want to remove.
3. Click **Configuration**, **Remove this Policy Profile**.
4. Click **OK** to confirm.

Simulating Policy

Before assigning a policy profile to a virtual machine, use the Infrastructure Automation controls policy simulation feature to determine if a virtual machine passes a policy profile.

Simulating Policy Profiles on Virtual Machines

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**, select the virtual machines that you need to evaluate.
2. Click **Policy**, and then click **Policy Simulation**.
3. From the **Select a Policy Profile to add** dropdown, click the policy that you need to apply to the selected virtual machines.



4. The virtual machine thumbnail displays in the **Policy Simulation** area.
   - A check sign in the virtual thumbnail shows that the virtual machine passes policy.
   - A minus sign in the virtual thumbnail shows that the virtual machine fails policy.
5. Click a virtual machine in the **Policy Simulation** area to see its details.
6. Expand a policy profile by clicking it to see its member policies and the status of the conditions.
   - Check **Show out of scope items** to show all conditions, whether the virtual machine passes the scope part of the condition. Uncheck it to hide conditions where the scope part fails.
   - Next to **Show policies**, check **Successful** to show policies that are passed and check **Failed** to see the policies that have failed. The default is to show both.
   - Items in green text passed the condition.
   - Items in red text failed the condition.
   - Items in red italics failed the condition, but do not change the outcome of the scope.

If you evaluate multiple policy profiles, you can see both policy profiles and a tree expanding down to their conditions.

Assigning Policy Profiles

After creating your policy profiles, you are ready to evaluate and assign them.

- Assign a policy profile to a virtual machine to apply the policy profile to a specific virtual machine, independent of its related host, provider, or repository.
- Assign a policy profile to a provider to apply the policy profile to all virtual machines, hosts, replicators, pods, container nodes or container images registered to that provider.
- Assign a policy profile to a replicator to apply the policy profile to that specific replicator.
- Assign a policy profile to a pod to apply the policy profile to that specific pod.
- Assign a policy profile to a container node to apply the policy profile to that specific node.
- Assign a policy profile to a container image to apply the policy profile to that specific image.
- Assign a policy profile to a cluster to apply the policy profile to all virtual machines or hosts assigned to that cluster.
- Assign a VM policy profile to a host to apply the policy profile to that specific host or all virtual machines registered to that host.
- Assign a VM policy profile to a resource pool to apply the policy profile to all virtual machines or hosts assigned to that resource pool.

Assigning Policy Profiles to an Infrastructure Provider

1. Browse to menu: **Compute > Infrastructure > Providers**, verify the provider you need to assign the policy profiles to.
2. Click (**Policy**), and then click (**Manage Policies**).
3. From the **Select Policy Profiles** area, you can click on the triangle next to a desired policy profile to expand it and see its member policies.
4. Check the policy profiles you require to apply to the provider. It turns blue to show its assignment state has changed.
5. Click **Save**.

Removing Policy Profiles from an Infrastructure Provider

1. Browse to menu: **Compute > Infrastructure > Providers**, check the providers that you want to remove the policy profile from.
2. Click **Policy**, and then click **Manage Policies**.
3. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
4. Click **Save**.

Assigning Policy Profiles to a Cluster

1. Browse to menu: **Compute > Infrastructure > Clusters**, check the clusters that you need to assign policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, you can click the triangle next to a desired policy profile to expand it and see its member policies.
4. Check the policy profiles that you need to apply to the cluster. It turns blue to show that its assignment state changed.
5. Click **Save**.

Removing Policy Profiles from a Cluster

1. Browse to menu: **Compute > Infrastructure > Clusters**, check the clusters that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, you can click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profiles that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Host

1. Browse to menu: **Compute > Infrastructure > Hosts**, check the hosts that you need to assign policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Check the policy profiles that you need to apply to the host. It turns blue to show that its assignment state changed.

5. Click **Save**.

Removing Policy Profiles from a Host

1. Browse to menu: **Compute > Infrastructure > Hosts**, check the hosts that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. Uncheck the policy profiles that you need to remove. It turns blue to show that its assignment state changed.
4. Click **Save**.

Assigning Policy Profiles to a Virtual Machine

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**, check the virtual machines that you need to assign policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Check the policy profiles that you need to apply to the host. It turns blue to show that its assignment state changed.
5. Click **Save**.

Removing Policy Profiles from a Virtual Machine

1. Browse to menu: **Compute > Infrastructure > Virtual Machines**, check the virtual machines that you want to remove the policy profile from.
2. Click **Policy**, and then click **Manage Policies**.
3. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
4. Click **Save**.

Assigning Policy Profiles to a Resource Pool

1. Browse to menu: **Compute > Infrastructure > Resource Pools**, check the resource pools that you need to assign policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Click the policy profiles that you need to apply to the resource pools. It turns blue to show that its assignment state changed.
5. Click **Save**.

Removing Policy Profiles from a Resource Pool

1. Browse to menu: **Compute > Infrastructure > Resource Pools**, check the resource pools that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profiles that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Cloud Provider

1. Browse to menu: **Compute > Clouds > Providers** and check the provider that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand it and see its member policies.
4. Check the policy profiles that you need to apply to the provider. The ones that are different from the previous setting shows in blue.
5. Click **Save**.

Removing Policy Profiles from a Cloud Provider

1. Browse to menu: **Compute > Clouds > Providers**, check the providers that you need to remove the policy profile from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.

5. Click **Save**.

Assigning Policy Profiles to a Network Provider

1. Browse to menu: **Networks > Providers**, check the network provider that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand it and see its member policies.
4. Check the policy profiles that you need to apply to the provider. The ones that are different from the previous setting shows in blue.
5. Click **Save**.

Removing Policy Profiles from a Network Provider

1. Browse to menu: **Networks > Providers**, check the network providers that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Container Provider

1. Browse to menu: **Compute > Containers > Providers** and select the provider that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand and see its member policies.
4. Select the policy profiles that you need to apply to the provider. It turns blue to show the selection.
5. Click **Save**.

Removing Policy Profiles from a Container Provider

1. Browse to menu: **Compute > Containers > Providers**, select the container providers that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Replicator

1. Browse to menu: **Compute > Containers > Replicators** and select the replicator that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand and see its member policies.
4. Select the policy profiles that you need to apply to the replicator. It turns blue to show the selection.
5. Click **Save**.

Removing Policy Profiles from a Replicator

1. Browse to menu: **Compute > Containers > Replicators**, select the replicators that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Pod

1. Browse to menu: **Compute > Containers > Pods** and select the pod that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.

3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand and see its member policies.
4. Select the policy profiles that you need to apply to the pod. It turns blue to show the selection.
5. Click **Save**.

Removing Policy Profiles from a Pod

1. Browse to menu: **Compute > Containers > Pods**, select the pods that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Container Node

1. Browse to menu: **Compute > Containers > Container Nodes** and select the container node that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand and see its member policies.
4. Select the policy profiles that you need to apply to the node. It turns blue to show the selection.
5. Click **Save**.

Removing Policy Profiles from a Container Node

1. Browse to menu: Compute > Containers > Container Nodes**, select the container nodes that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to a Container Image

1. Browse to menu: **Compute > Containers > Container Images** and select the image that you need to assign the policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand and see its member policies.
4. Select the policy profiles that you need to apply to the image. It turns blue to show the selection.
5. Click **Save**.

Removing Policy Profiles from a Container Image

1. Browse to menu: **Compute > Containers > Container Images**, select the container images that you need to remove the policy profiles from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Assigning Policy Profiles to an Instance

1. Browse to menu: **Compute > Clouds > Instances**, check the instances that you want to assign policy profiles to.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a wanted policy profile to expand it and see its member policies.
4. Check the policy profiles that you want to apply to the instances. It turns blue to show that its assignment state changed.
5. Click **Save**.

Removing Policy Profiles from an Instance

1. Browse to menu: **Compute > Clouds > Instances**, check the instances that you need to remove the policy profile from.
2. Click **Policy**, and then click **Manage Policies**.
3. From the **Select Policy Profiles** area, click the triangle next to a desired policy profile to expand it and see its member policies.
4. Uncheck the policy profile that you need to remove. It turns blue to show that its assignment state changed.
5. Click **Save**.

Disabling a Policy in a Policy Profile

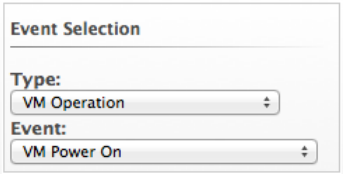
You can disable one policy in a profile without removing it from the policy, perhaps for trouble shooting purposes or because the policy is not required temporarily.

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, then navigate to the policy that you need to disable or navigate to the policy from the policy profile.
3. Click **Configuration**, **Edit Basic Info, Scope, Notes**.
4. Uncheck **Active**.
5. Click **Save**.

Viewing Policy Simulation - Resultant Set of Policy (RSOP)

After the Policy Profiles are assigned, you can see the final result of the resolution of all policies based on which Events occur. Based on the result, you can adjust your Policies. To view RSOP, go to the control area in the Infrastructure Automation console.

1. Browse to menu: **Control > Simulation**.
2. From the **Event Selection** area, select a type of event, and then the specific event you need the result for.



3. From the **VM Selection** area, select the virtual machine from a provider, cluster, host, or a single virtual machine.



4. Click **Submit**.

Exporting and Importing Analysis Profiles

Infrastructure Automation SmartState analysis requires an analysis profile to select the files to be scanned by a compliance policy. Infrastructure Automation allows the export and import of SmartState analysis profiles by using the command line and rake commands. As a result, approved configurations can be easily imported into customer environments, without having to manually recreate the profile through the user interface.

1. Change to the `vmdb` directory:

   ```
   cd /var/www/miq/vmdb
   ```

2. Create an export directory:

   ```
   $ mkdir exports
   ```

3. To export an analysis profile, run:

   ```
   bundle exec rake evm:export:scan_profiles -- --directory exports
   ```

4. To import the default analysis profile, run:

   ```
   bundle exec rake evm:import:scan_profiles -- --source exports/host_default.yaml
   ```

   Note:

   If the default profile exists in Infrastructure Automation, the new profile does not overwrite the old profile. Instead, it duplicates the file items in the default profile.

Appendix

Events

Events are triggers that cause a condition to be tested. Control provides several Events, that can be divided into functional types. Events cannot be modified.

Category	Description
Container Operation	Events related to container analysis.
Datastore Operation	Events related to datastore analysis.
Authentication Validation	Events related to credential validation for hosts and providers.
Company Tag	Events related to assigning and removing company tags from an infrastructure object.
Compliance	Events related to checking compliance policies.
Host Operation	Events related to the connection state of a host and status of a SmartState Analysis on a host.
Orchestration Lifecycle	Events related to orchestration lifecycle, such as retirement.
Physical Server Operation	Events related to the connection state of a physical server.
VM Configuration	Events associated with a change in configuration of a virtual machine. These include, but are not limited to, clone, create, template create, and settings change.
VM Lifecycle	Events such as virtual machine discovery, provisioning, and virtual machine retirement.
VM Operation	Events associated with power states or locations of virtual machines and virtual desktop machines. These include, but are not limited to, power off, power on, reset, resume, shutdown, and suspend.
Service Lifecycle	Events associated with service lifecycle. These include, but are not limited to, provisioning completed, start request, started, stop request, stopped, retirement warning, and retired.

Event Types

Each type has a set of events that you can select to trigger the checking of a condition.

Event	Description
Container Image Analysis Request	Check the condition when an analysis of a container image is requested.
Container Image Analysis Complete	Check the condition when an analysis of a container image completes.
Container Image Discovered	Check the condition when a new container image is discovered.
Container Image Compliance Check	Check the condition when a compliance check is performed on an image.
Container Image Compliance Passed	Check the condition when an image passes a compliance check.
Container Image Compliance Failed	Check the condition when an image fails a compliance check.
Container Node Failed Mount	Check the condition when a node fails to mount a volume for a pod.
Container Node Invalid Disk Capacity	Check the condition when a node's disk capacity is invalid.
Container Node Not Ready	Check the condition when a node is not ready.
Container Node Not Schedulable	Check the condition when a node is not schedulable.
Container Node Ready	Check the condition when a node is ready.
Container Node Schedulable	Check the condition when a node is schedulable.
Container Node Rebooted	Check the condition when a node reboots.
Container Node Compliance Check	Check the condition when a compliance check is performed on a node.
Container Node Compliance Passed	Check the condition when a node passes a compliance check.
Container Node Compliance Failed	Check the condition when a node fails a compliance check.
Pod Compliance Check	Check the condition when a compliance check is performed on a pod.
Pod Compliance Passed	Check the condition when a pod passes a compliance check.
Pod Compliance Failed	Check the condition when a pod fails a compliance check.
Pod Container Created	Check the condition when a container is created in a pod.

Event	Description
Pod Container Failed	Check the condition when a container in a pod fails.
Pod Container Killing	Check the condition when a container in a pod is killed.
Pod Container Started	Check the condition when a container in a pod is started.
Pod Container Stopped	Check the condition when a container in a pod is stopped.
Pod Container Unhealthy	Check the condition when a container in a pod is unhealthy.
Pod Deadline Exceeded	Check the condition when a pod with specified deadline exceeds it and is terminated.
Pod Failed Scheduling	Check the condition when scheduling a pod fails.
Pod Failed Sync	Check the condition when getting a pod to its desired state fails (a frequent reason is failure to download the image).
Pod Failed Validation	Check the condition when a pod validation fails.
Pod hostPort Conflict	Check the condition when a pod hostPort conflict occurs.
Pod Insufficient Free CPU	Check the condition when there is an insufficient free CPU in a pod.
Pod Insufficient Free Memory	Check the condition when there is an insufficient free memory in a pod.
Pod nodeSelector Mismatching	Check the condition when a pod nodeSelector mismatches.
Pod Out of Disk	Check the condition when a pod is out of disk space.
Pod Scheduled	Check the condition when a pod is scheduled onto the node.
Replicator Failed Creating Pod	Check the condition when a replicator fails creating a pod.
Replicator Successfully Created Pod	Check the condition when a replicator successfully creates a pod.
Replicator Compliance Check	Check the condition when a compliance check is performed on a replicator.
Replicator Compliance Passed	Check the condition when a replicator passes a compliance check.
Replicator Compliance Failed	Check the condition when a replicator fails a compliance check.
Database Failover Executed	Check the condition when a database failover is executed.
Datastore Analysis Complete	Check the condition when a SmartState Analysis of a datastore completes.
Datastore Analysis Request	Check the condition when a SmartState Analysis for a datastore is requested from the user interface.
Host Added to Cluster	Check the condition when a host is added to a cluster.
Host Analysis Complete	Check the condition when a SmartState Analysis of host completes.
Host Analysis Request	Check the condition when a SmartState Analysis is requested from the Infrastructure Automation console.
Host Auth Changed	Check the condition when host authentication credentials are changed in the Infrastructure Automation console.
Host Auth Error	Check the condition if there is any other error connecting to the host such as ssh/vim handshaking problems, timeouts, or any other uncategorized error.
Host Auth Incomplete Credentials	Check the condition if host authentication credentials are not complete in the user interface.
Host Auth Invalid	Check the condition if Infrastructure Automation is able to communicate with the host and the credentials fail.
Host Auth Unreachable	Check the condition if Infrastructure Automation is unable to communicate with the host.
Host Auth Valid	Check the condition when the host authentication credentials entered in the Infrastructure Automation console are valid.
Host C & U Processing Complete	Check the condition when the processing of capacity and utilization data has finished.
Host Compliance Check	Check the condition when a compliance check is performed on a host.
Host Compliance Failed	Check the condition when a host fails a compliance check.
Host Compliance Passed	Check the condition when a host passes a compliance check.
Host Connect	Check the condition when a host connects to a provider.
Host Disconnect	Check the condition when a host disconnects from a provider.
Host Maintenance Enter Request	Check the condition when a host requests to enter maintenance mode.
Host Maintenance Exit Request	Check the condition when a host requests to exit maintenance mode.
Host Provision Complete	Check the condition when the host provision is complete.
Host Reboot Request	Check the condition when someone tries to reboot a host from the Infrastructure Automation console.

Event	Description
Host Removed from Cluster	Check the condition when a host is removed from a cluster.
Host Reset Request	Check the condition when a host is restarted from the Infrastructure Automation console.
Host Shutdown Request	Check the condition when a host is shut down from the Infrastructure Automation console.
Host Standby Request	Check the condition when someone tries to put the operating system of a host in standby from the Infrastructure Automation console.
Host Start Request	Check the condition when a host is started from the Infrastructure Automation console.
Host Stop Request	Check the condition when a host is requested to stop from the Infrastructure Automation console.
Host Vmotion Disable Request	Check the condition when a request to disable vMotion on a host is created from the Infrastructure Automation console.
Host Vmotion Enable Request	Check the condition when a request to enable vMotion on a host is created from the Infrastructure Automation console.
Orchestration Stack Retire Request	Check the condition when an orchestration stack retirement request is created from Infrastructure Automation.
Physical Server Reset	Check the condition when a physical server is restarted from the Infrastructure Automation console.
Physical Server Shutdown	Check the condition when a physical server is shut down from the Infrastructure Automation console.
Physical Server Start	Check the condition when a physical server is started from the Infrastructure Automation console.
Provider Auth Changed	*For use only with Infrastructure Automation automate, for future use in policies.* Check the condition when provider authentication credentials are changed in the user interface.
Provider Auth Error	*For use only with Infrastructure Automation automate, for future use in policies.* Check the condition if there is any other error connecting to the provider such as ssh/vim handshaking problems, timeouts, or any other uncategorized error.
Provider Auth Incomplete Credentials	*For use only with Infrastructure Automation automate, for future use in policies.* Check the condition if provider authentication credentials are not complete in the Infrastructure Automation console.
Provider Auth Invalid	*For use only with Infrastructure Automation automate, for future use in policies.* Check the condition if Infrastructure Automation is able to communicate with the provider and the credentials fail.
Provider Auth Unreachable	*For use only with Infrastructure Automation automate, for future use in policies.* Check the condition if Infrastructure Automation is unable to communicate with the provider.
Provider Auth Valid	*For use only with Infrastructure Automation automate, for future use in policies.* Check the condition when the provider authentication credentials entered in the user interface are valid.
Provider Compliance Check	Check the condition when a compliance check is performed on a provider.
Provider Compliance Failed	Check the condition when a provider fails a compliance check.
Provider Compliance Passed	Check the condition when a provider passes a compliance check.
Service Provision Complete	Check the condition when the service provision is complete.
Service Retire Request	Check the condition when a service retirement request is created from Infrastructure Automation.
Service Retired	Check the condition when the service has been retired.
Service Retirement Warning	Check the condition when the service is about to retire.
Service Start Request	Check the condition when the service has been requested to start.
Service Started	Check the condition when the service has started.
Service Stop Request	Check the condition when the service has been requested to stop.
Service Stopped	Check the condition when the service has stopped.
Tag Complete	Check the condition after a company tag is assigned.
Tag Parent Cluster Complete	Check the condition after a company tag is assigned to a virtual machine’s parent cluster.
Tag Parent Datastore Complete	Check the condition after a company tag is assigned to a virtual machine’s parent datastore.
Tag Parent Host Complete	Check the condition after a company tag is assigned to a virtual machine’s parent host.
Tag Parent Resource Pool Complete	Check the condition after a company tag is assigned to a virtual machine’s parent resource pool.
Tag Request	Check the condition when assignment of a company tag is attempted.
Un-Tag Complete	Check the condition when a company tag is removed.
Un-Tag Parent Cluster Complete	Check the condition after a company tag is removed from a virtual machine’s parent cluster.
Un-Tag Parent Datastore Complete	Check the condition after a company tag is removed from a virtual machine’s parent datastore.
Un-Tag Parent Host Complete	Check the condition after a company tag is removed from a virtual machine’s parent host.
Un-Tag Parent Resource Pool Complete	Check the condition after a company tag is removed from a virtual machine’s parent resource pool.
Un-Tag Request	Check the condition when an attempt is made to remove a company tag.

Event	Description
VDI Connecting to Session	Check the condition when a VDI session is started.
VDI Disconnected from Session	Check the condition when a VDI session is disconnected.
VDI Login Session	Check the condition when a user logs on to a VDI session.
VDI Logoff Session	Check the condition when a user logs off from a VDI session.
VM Analysis Complete	Check the condition when a SmartState Analysis of virtual machine completes.
VM Analysis Failure	Check the condition when a SmartState Analysis of virtual machine fails.
VM Analysis Request	Check the condition when a SmartState Analysis is requested from the Infrastructure Automation console.
VM Analysis Start	Check the condition when a SmartState Analysis of virtual machine is started.
VM C & U Processing Complete	Check the condition when the processing of capacity and utilization data has finished.
VM Clone Complete	Check the condition when a virtual machine is cloned.
VM Clone Start	Check the condition when a virtual machine clone is started.
VM Compliance Check	Check the condition when a compliance check is performed on a virtual machine.
VM Compliance Failed	Check the condition when a virtual machine fails a compliance check.
VM Compliance Passed	Check the condition when a virtual machine passes a compliance check.
VM Create Complete	Check the condition when a virtual machine is created.
VM Delete (from Disk)	Check the condition when a disk on a virtual machine is deleted.
VM Delete (from Disk) Request	Check the condition when someone tries to delete a virtual machine from disk from the user interface.
VM Guest Reboot	Check the condition when a virtual machine is rebooted.
VM Guest Reboot Request	Check the condition when someone tries to reboot a virtual machine from the Infrastructure Automation console.
VM Guest Shutdown	Check the condition when the operating system of a virtual machine shuts down.
VM Guest Shutdown Request	Check the condition when someone tries to shut down the operating system of a virtual machine from the user interface.
VM Live Migration (VMOTION)	Check the condition when a vMotion migration is performed.
VM Pause	Check the condition when a virtual machine is paused.
VM Pause Request	Check the condition when someone tries to pause a virtual machine from the Infrastructure Automation console.
VM Power Off	Check the condition when a virtual machine is turned off.
VM Power Off Request	Check the condition when someone tries to power off a virtual machine from the Infrastructure Automation console.
VM Power On	Check the condition when a virtual machine is turned on.
VM Power On Request	Check the condition when someone tries to turn on a virtual machine from the Infrastructure Automation console.
VM Provision Complete	Check the condition when a virtual machine is provisioned.
VM Remote Console Connected	Check the condition when a virtual machine is connected to a remote console.
VM Removal from Inventory	Check the condition when a virtual machine is unregistered.
VM Removal from Inventory Request	Check the condition when a request is sent from the Infrastructure Automation console to unregister a virtual machine.
VM Renamed Event	Check the condition when a virtual machine is renamed on its provider.
VM Reset	Check the condition when a virtual machine is restarted.
VM Reset Request	Check the condition when a virtual machine is restarted from the Infrastructure Automation console.
VM Resume	Check the condition when a virtual machine is resumed.
VM Retire Request	Check the condition when a virtual machine retirement request is created from Infrastructure Automation.
VM Retired	Check the condition when a virtual machine is retired.
VM Retirement Warning	Check the condition when a warning threshold is reached for retirement.
VM Settings Change	Check the condition when the settings of virtual machine are changed.
VM Shelve	Check the condition when a virtual machine is shelved.
VM Shelve Offload	Check the condition when a virtual machine is removed and deleted with the shelve offload operation.
VM Shelve Offload Request	Check the condition when a shelve offload request is created from Infrastructure Automation for a virtual machine.
VM Shelve Request	Check the condition when a virtual machine shelve request is created from Infrastructure Automation.
VM Snapshot Create Complete	Check the condition when a snapshot is completed.
VM Snapshot Create Request	Check the condition when someone tries to create a snapshot of a virtual machine from the user interface.
VM Snapshot Create Started	Check the condition when a snapshot creation is started.
VM Standby of Guest	Check the condition when the operating system of a virtual machine goes to standby.

Event	Description
VM Standby of Guest Request	Check the condition when someone tries to put the operating system of a virtual machine in standby from the Infrastructure Automation console.
VM Suspend	Check the condition when a virtual machine is suspended.
VM Suspend Request	Check the condition when someone tries to suspend a virtual machine from the Infrastructure Automation console.
VM Template Create Complete	Check the condition when a virtual machine template is created.

Events and Descriptions

- Monitoring, Alerts, and Reporting
  - Overview Dashboard
    - Adding a Widget
    - Resetting to the Default Set of Widgets
    - Removing a Widget
    - Zooming in to a Chart Widget
    - Opening a Chart or Report Widget in Full Screen View
    - Minimizing or Maximizing a Widget
    - Downloading a Report Widget as a File
  - Creating Dashboard Widgets
    - Creating a Report Widget
    - Creating a Chart Widget
    - Creating a Menu Widget
    - Editing a Widget
    - Copying a Widget
    - Deleting a Widget
    - Importing a Widget
    - Exporting a Widget
    - Generating Widget Content Immediately
  - Reports
    - Running Reports
      - Generating a Single Report
    - Scheduling a Report
      - Modifying a Report Schedule
      - Running a Scheduled Report Immediately
    - Viewing Reports
      - Changing Report Views
        - Filtering Report Data
        - Sorting Reports by Column
      - Downloading a Report
      - Showing a Report in Full Screen
    - Adding a Report
    - Copying a Report
    - Editing a Report
    - Deleting a Report
    - Importing a Report
    - Exporting a Report
    - Managing Report Menus
      - Managing Report Menu Accordions
      - Managing Report Menu Folders
      - Organizing Reports in Report Menus
  - Chargeback
    - Chargeback Rates
      - Memory Used Cost
      - CPU Total Cost
      - CPU Used Cost
      - Storage Allocated Cost
      - Storage Total Cost
      - Storage Used Cost
    - Creating Chargeback Rates
      - Editing a Chargeback Rate
      - Copying a Chargeback Rate
      - Adding a Chargeback Rate
    - Assigning Chargeback Rates
      - Configuring Chargeback for Compute
      - Configuring Chargeback for Storage
    - Creating a Chargeback Report
  - Optimization
    - Running an Optimization Report
    - Viewing Optimization Reports
  - Alerts

        - Assigning the Notifier Role
        - Creating an Alert
        - Creating a Hardware Reconfigured Alert
        - Creating a Normal Operating Range Alert
        - Creating a Real Time Performance Alert
        - Creating an Hourly Performance Alert
        - Creating a hostd Log Threshold Alert
        - Creating a VMware Alarm Alert
        - Creating an Expression Alert
        - Create an Operational Alert
        - Operational Alert Types
        - Editing an Alert
        - Copying an Alert
        - Deleting an Alert
        - Evaluating an Alert
    - Alert Profiles
        - Creating Alert Profiles
        - Editing an Alert Profile
        - Deleting an Alert Profile
        - Assigning an Alert Profile
    - Importing and Exporting
        - Importing and Exporting Policies, Policy Profiles, and Alerts
        - Exporting a Policy, Policy Profile, or an Alert
    - OpenShift Prometheus Alerts
        - Configuring Prometheus Alerts in Infrastructure Automation
            - Deploying and Configuring Prometheus on OpenShift
            - Assigning Prometheus Alert Profiles to the Enterprise
            - Adding an OpenShift Containers Provider with a Prometheus Alert Endpoint
- Regular Expressions
- Reportable Fields in Infrastructure Automation

Monitoring, Alerts, and Reporting

Overview Dashboard

Overview shows your virtual environment's events, reports, and configurable alerts. This information supports better information technology decision making and predictions for future virtual machine management.

When you log in to the console, it brings you directly to the **Overview Dashboard** page by default. The console uses widgets to organize this page, providing you with a default set of commonly used widgets. The configuration items on these widgets are clickable, allowing you to drill down directly to a referenced item.

Customize this page to include charts and reports you specifically want to see as soon as you log in to the console. You can add, remove, move, minimize, zoom into, and maximize widgets. Only users with the proper access can create widgets.

Adding a Widget

You can add widgets to the dashboard to accommodate the information you want to see upon login.

1. Navigate to **Overview > Dashboard**.
2. Click **(Add a Widget)**.
3. Select the widget you want to add from the list.

Note:

Only widgets that are not currently showing on the dashboard will appear in this list.

Resetting to the Default Set of Widgets

You can reset to default set of widgets according to your needs with the following steps:

1. Navigate to **Overview > Dashboard**.
2. Click **(Reset Dashboard Widgets to the defaults)**.

Removing a Widget

You can remove a widget when you no longer need the widget to be displayed in the dashboard.

1. Navigate to **Overview > Dashboard**.
2. From the widget that you want to remove, click **(Remove from Dashboard)** in the upper right corner of the widget.

3. Click **OK**.

The widget is removed from the dashboard, but it is not deleted. It can be added again if needed.

Zooming in to a Chart Widget

1. Navigate to **Overview > Dashboard**.
2. From the chart widget that you want to enlarge, click **(Zoom in)** in the upper right corner of the widget.

The enlarged chart is opened in its own pop-up window. To close this window, click **(Close)** in the upper right corner.

Opening a Chart or Report Widget in Full Screen View

1. Navigate to **Overview > Dashboard**.
2. From the chart or report widget that you want to enlarge, click **(Full Screen view)** in the upper right corner of the widget.
3. Click **OK**.

Minimizing or Maximizing a Widget

1. Navigate to **Overview > Dashboard**.
2. From the chart or report widget that you want to maximize, click **(Minimize)** or **(Full Screen)** in the upper right corner of the widget.

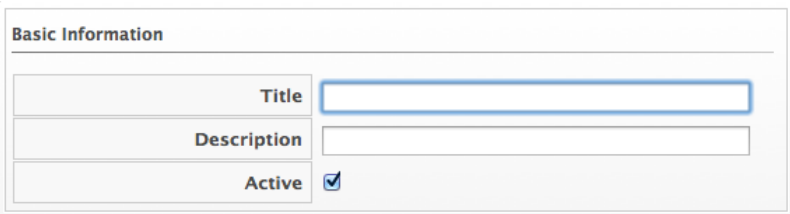
Downloading a Report Widget as a File

1. Navigate to **Overview > Dashboard**.
2. From the chart or report widget that you want to download as a PDF, click **[(Download PDF)** in the upper right corner of the widget.
3. Click **OK**.

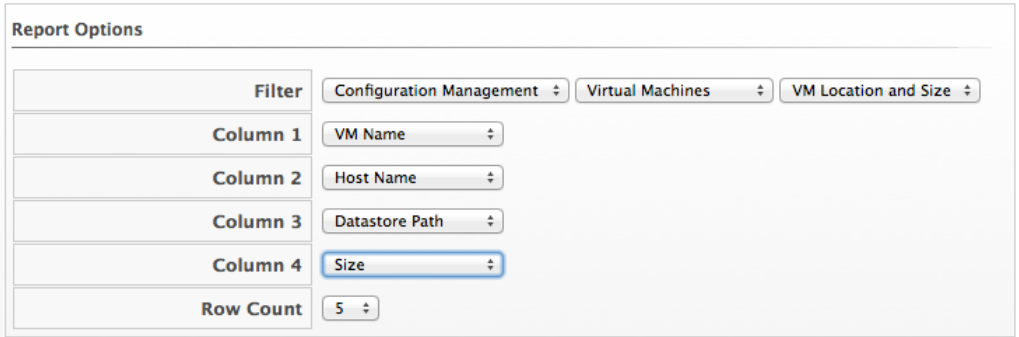
Creating Dashboard Widgets

Creating a Report Widget

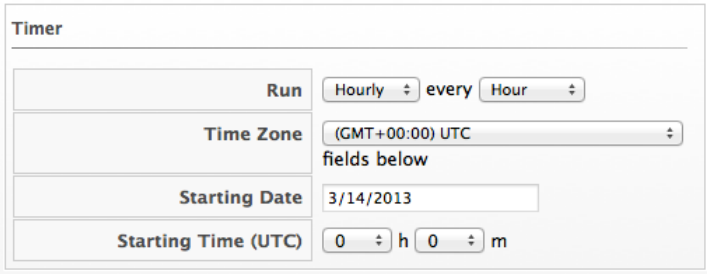
1. Navigate to **Overview > Reports**.
2. Click on the **Dashboard Widgets** accordion, then choose the **Reports** folder.
3. Click **(Configuration)**, then click **(Add a new Widget)**.
4. In the **Basic Information** area, type in a **Title** and **Description**. By default the widget will be active as soon as you create it. To make it inactive, uncheck the **Active** box.



5. From the **Report Options** area, select the filters until you are at the report you want for this widget. Select up to four columns from that report. Finally, for **Row Count**, select the number of rows that you want displayed.



6. In the **Timer** area, click the **Run** drop down to specify how often you want the widget data to get updated. The options displayed will depend on which **Run** option you choose. Select **Hourly**, **Daily**, **Weekly**, or **Monthly**.



7. Select a **Time Zone**. Type or select a date to begin the schedule in **Starting Date**. Select a **Starting Time** based on a 24 hour clock in the selected time zone.

   Note:

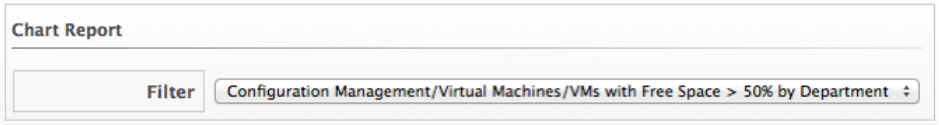
   If you change the time zone, you will need to reset the starting date and time.

8. In the **Visibility** area, select **<To All Users>**, so that all users can use this widget no matter what user role they are assigned. Select **<By Role>** to assign this widget to specific user roles. Select **<By Group>** to assign this widget to specific groups.

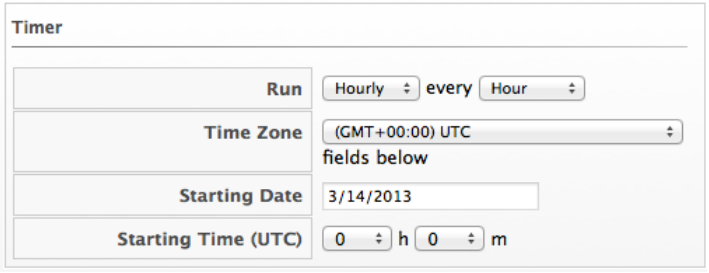
9. Click **Add**.

Creating a Chart Widget

1. Navigate to **Overview > Reports**.
2. Click the **Dashboard Widgets** accordion and click the **Charts** folder.
3. Click **(Configuration)**, then click **(Add a new Widget)**.
4. In the **Basic Information** area, type in a **Title** and **Description**. By default the widget will be active as soon as you create it. To make it inactive, uncheck the **Active** box.
5. From the **Chart Report** area, select a chart to display in the widget.



6. In the **Timer** area, click the **Run** drop down to specify how often you want the widget data to get updated. The options displayed will depend on which **Run** option you choose. Select **Hourly**, **Daily**, **Weekly**, or **Monthly**.



7. Select a **Time Zone**. Type or select a date to begin the schedule in **Starting Date**. Select a **Starting Time** based on a 24 hour clock in the selected time zone.

   Note:

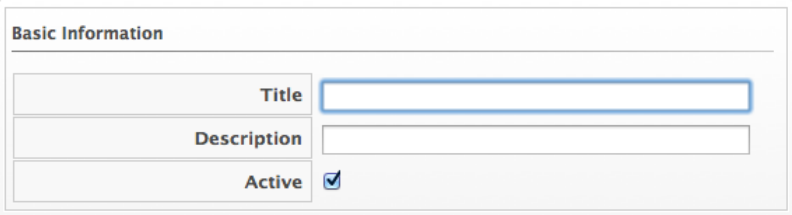
   If you change the time zone, you will need to reset the starting date and time.

8. In the **Visibility** area, select **<To All Users>**, so that all users can use this widget no matter what user role they are assigned. Select **<By Role>** to assign this widget to specific user roles. Select **<By Group>** to assign this widget to specific groups.

9. Click **Add**.

Creating a Menu Widget

1. Navigate to **Overview > Reports**.
2. Click the **Dashboard Widgets** accordion and click the **Menus** folder.
3. Click **(Configuration)**, then click **(Add a new Widget)**.
4. In the **Basic Information** area, type in a **Title** and **Description**. By default the widget will be active as soon as you create it. To make it inactive, uncheck the **Active** box.



5. In the **Menu Shortcuts** area, use the **Add a Shortcut** dropdown to select all the places in the console that you want to add to this widget.
6. In the **Visibility** area, select **<To All Users>**, so that all users can use this widget no matter what user role they are assigned. Select **<By Role>** to assign this widget to specific user roles. Select **<By Group>** to assign this widget to specific groups.
7. Click **Add**.

Editing a Widget

1. Navigate to **Overview > Reports**.
2. Click the **Dashboard Widgets** accordion and select the widget you want to edit.
3. Click **(Configuration)**, and then **(Edit this Widget)**.
4. Make the required changes.
5. Click **Save**.

Copying a Widget

1. Navigate to **Overview > Reports**.
2. Click the **Dashboard Widget** accordion, and select the widget you want to copy.
3. Click **(Configuration)**, and then **(Copy this Widget)**.
4. Type a unique name for the widget and edit its properties.
5. Click **Save**.

Deleting a Widget

Note:

Default widgets cannot be deleted, but they can be copied.

1. Navigate to **Overview > Reports**.
2. Click the **Dashboard Widget** accordion and select the widget you want to delete.
3. Click **(Configuration)**, and then **(Delete this Widget from the Database)**.
4. Click **OK**.

Importing a Widget

You can share widgets between appliances using the export and import features.

1. Navigate to **Overview > Reports**.
2. In the **Import/Export** accordion, click **Widgets**.
3. In the **Import** area, click **Browse** to select an import file.
4. Click **Upload**.

Exporting a Widget

You can share widgets between appliances using the export and import features.

1. Navigate to **Overview > Reports**.
2. In the **Import/Export** accordion, click **Widgets**.
3. In the **Export** area, select the widgets that you want to export.
4. Click **Export**.

Generating Widget Content Immediately

1. Navigate to **Overview > Reports**.
2. Click the **Dashboard Widgets** accordion and select the widget you want to generate.

3. Click **(Configuration)**, and then **(Generate Widget Content now)**.
4. Click **OK**.

The content is generated immediately instead of waiting for the next scheduled update. Generation of widget content is shown under **Tasks**, accessed from the settings menu.

Reports

Click the **Reports** accordion under **Overview > Reports** to see a list of reports available. These reports have been constructed to help you view the most commonly requested and significant data. From here, you can also create reports if you have appropriate access. Infrastructure Automation provides a large group of default reports organized into categories. Each category has its own set of subfolders.

- **Configuration Management**: hardware, application, network, service, user account, operating system, and snapshot information for all of your items.
- **Migration Readiness**: information related to items required to migrate a virtual machine.
- **Operations**: free space on registered and unregistered virtual machines, power states for virtual machines, and SmartState analysis status.

  This category also provides reports relating to the operation of Infrastructure Automation, such as user IDs and snapshots taken by Infrastructure Automation.
- **VM Sprawl**: usage information and disk waste.
- **Relationships**: virtual machine, folder, and cluster relationships.
- **Events**: operations and configuration management events.
- **Performance by Asset Type**: performance of your virtual infrastructure.

  You must be capturing capacity and utilization data to get this information.
- **Running Processes**: information on processes running on a virtual machine.

  You must have domain credentials entered for the zone to collect the information for these reports, and the virtual machine must have been analyzed at least once.
- **Trending**: projections of datastore capacity, along with host CPU and memory use.
- **Tenants**: quotas report aggregated by each tenant that shows quota name, total quota, in use, allocated, and available. The report currently lists all tenants and there is no nesting information available by parent and child tenants.
- **Provisioning**: provisioning activity based on the approver, datastore, requester, and virtual machine.

For a complete list of reportable fields in Infrastructure Automation, see Reportable Fields.

Running Reports

There are two different ways to generate a report: by scheduling the report, or running the report manually by clicking the report generation button on the **Reports** page. Infrastructure Automation uses interactive report generation so that reports are placed on a queue. A visual indicator of the reports status is shown. All reports are automatically saved so that they can be downloaded and analyzed later.

Generating a Single Report

1. Navigate to **Overview > Reports**.
2. Click the **Reports > All Reports** accordion. Navigate to the report you want to view by selecting the appropriate folder.
3. Click (**Queue**).
4. The report generation is placed on the queue and its status shows in the reports page.

	Queued At	Run At	Source	User ID	Status
	10/21/11 14:14:11 UTC		Requested by user	admin	Queued
	10/21/11 14:00:08 UTC	10/21/11 14:00:22 UTC	Requested by user	admin	Finished

5. Click **(Reload current display)** to update the status.
6. When a report has finished generating, click on its row to view it.

Scheduling a Report

You can view historical data by creating reports on a scheduled basis. In addition, scheduled reports can be emailed directly to users.

Note:

You may need to disable, change the report filter, or change the frequency of a schedule. To do this, you will need to edit the schedule.

There are two ways to schedule a report:

Add a new Schedule through the **Reports** accordion:

1. Navigate to **Overview > Reports**.
2. Select a report from the **Reports** accordion and click **Configuration**.
3. Click **Add a New Schedule** or click the Schedules accordion and click **Configuration** > **Add a New Schedule**.

or

Add a new Schedule through the **Schedules** accordion:

1. Navigate to **Overview > Reports**.
2. Click the **Schedules** accordion.
3. Click **Configuration**.
4. Click **Add a New Schedule**.

Adding new Schedule details.

1. In the **Basic Information** area, type in a **Name** and **Description** for the schedule.
   - By default, **Active** is checked to enable the scan.
2. The **Report Selection** area is pre-populated if you added the schedule directly from the report. If you are adding from the schedule according, use the **Filter** drop downs to select the report that you want to schedule.
3. In the **Timer** area, click the **Run** drop down to specify how often you want the analysis to run. Your options after that will depend on which run option you choose.
   - Click **Once** to have the analysis run just one time.
   - Click **Daily** to run the analysis on a daily basis. You will be prompted to select how many days you want between each analysis.
   - Click **Hourly** to run the analysis hourly. You will be prompted to select how many hours you want between each analysis.
   - Type or select a date to begin the schedule in **Starting Date**.
   - Select a **Starting Time** based on a 24 hour clock in the Infrastructure Automation appliance's Time Zone.
4. Use the **E-Mail after Running** feature to send an email after the report has been generated to selected users. The email will include a link to the report.
   - Check **Send an E-mail**.
   - In **From (leave blank for default)**, type in the sending email.
   - Use **Add a User**, to select a specific user. The user must have a valid email address entered under accounts.
   - Use **Add (enter manually)** to type in the address not registered to a User.
   - Then, click **(Add)**.
5. If you are sending an email after the report runs, then you can select further options under Email Options.
   - Check **Send if Report is Empty** if you want an email even if no records exist in the report.
   - Next to Attachments, check if you would like the report attached as a **Text**, **CSV**, or **PDF** file.
6. Click **Add**.

Note:

See Outgoing SMTP Email Settings in the *General Configuration* guide, to learn how to verify the address and validate outgoing email settings.

Modifying a Report Schedule

1. Navigate to **Overview > Reports**
2. Click the **Schedules** accordion and select the schedule you want to edit.
3. Click (**Configuration**), then click (**Edit this Schedule**).
4. Make the required changes.
5. Click **Save**.

Running a Scheduled Report Immediately

1. Navigate to **Overview > Reports**.
2. Click the **Schedules** accordion and select the schedule you want to run.

3. Click (**Configuration**), then click (**Queue up this Schedule to run now**).

Viewing Reports

Once you have run a report, you can view it at any time after the first scheduled time has occurred.

1. Navigate to **Overview > Reports**.
2. Click the **Saved Reports** accordion or the **Reports** accordion.
3. Click on the instance of the report you want to view.

Changing Report Views

After running a report, you can customize your view in several ways.

Some reports can be viewed as charts as well as lists. This will depend on the type of data and on how the report has been created. Where applicable, you will see these additional buttons:

1. Navigate to **Overview > Reports**.
2. Click a completed report to view. Click one of the following buttons for the view you want:
   - Click for **Graph View**.
   - Click for **Hybrid View**.
   - Click for **Tabular View**.

Filtering Report Data

To filter the report information you see, use the **Data View** button.

1. Navigate to **Overview > Reports**.
2. Click a completed report to view.
3. Click the button for **Data View**, which opens the search field.
4. Select the criteria from the dropdown menu to filter by, and enter your search term.
5. Press **Enter** to search.

Sorting Reports by Column

In tabular and hybrid view, you can sort a list of reports by column.

1. Navigate to **Overview > Reports > Saved Reports**.
2. Click a report name to show a list of completed reports.
3. Click any blue column heading to sort the reports by the values in that column.

Downloading a Report

Download reports to analyze the data using other tools or to print the report.

When you click on one of the supplied reports, you are presented with a group of buttons to download the report in one of three formats or to view the report in a full screen.

1. Navigate to **Overview > Reports**.
2. Click the report you want to view.
3. Click on the row for the instance of the report you want to download. If the report needs to be generated, see Running Reports.
4. Click on the report download buttons for the type of export you want.
   - Click **(Download this report in text format)** to download as text.
   - Click **(Download this report in csv format)** to download as a comma-separated file.
   - Click **(Download this report in PDF format)** to download as PDF.
   - The report is automatically named with the type of report and date.

Note:

Edit and delete buttons are only visible to administrators and super administrators. Edit and delete functions are only available to customer-created reports. The Infrastructure Automation pre-configured reports cannot be edited or deleted, but they can be copied.

Showing a Report in Full Screen

View the report in full screen to examine report details. From full screen, you can also print the chart that accompanies a report.

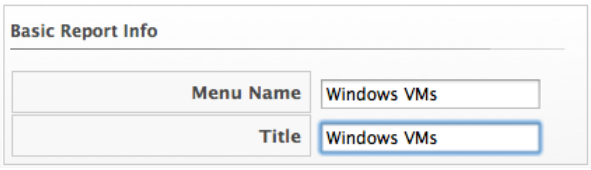
1. Navigate to **Overview > Reports**.
2. Click the **Reports > All Reports** accordion and select the report that you want to view.
3. Click the **Saved Reports** tab and select a report.
4. On the **Saved Report** detail screen, click (**Configuration**), and click (**Show full screen Report**).

The report will appear in full screen view. Click back on your browser to return to the reports explorer.

Adding a Report

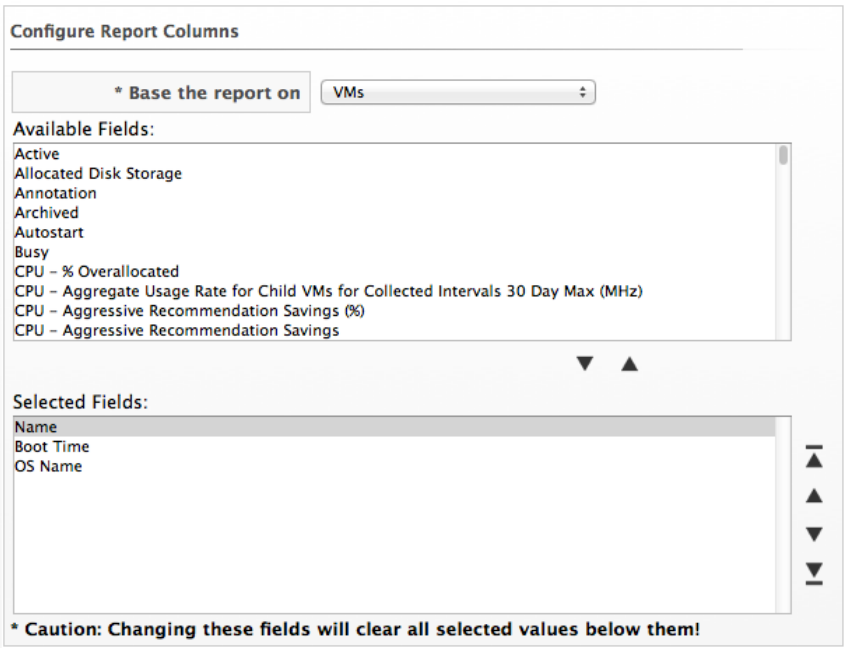
Add reports if the default reports do not include what you need or you want to narrow the scope of a report. For example, you may want a report that shows only Windows virtual machines.

1. Navigate to **Overview > Reports**.
2. Click the **Reports > All Reports** accordion.
3. Click **(Configuration)**, then click **(Add a New Report)**.
4. In the **Columns** tab, edit the **Basic Report Info** area.



   - Type a unique name in **Menu Name** for how you want the report described in the menu list.
   - Type the **Title** you want displayed on the report in title.

5. Add fields in the **Configure Report Columns** area.



   - Use the **Base the report on** table dropdown to choose a table to get fields from.

     Note:

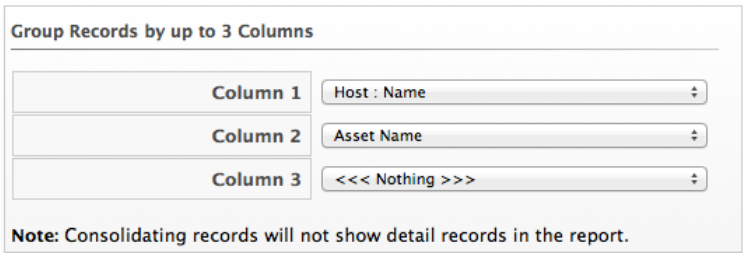
     If you change the report base or the interval, all selections below will be reset.

   - Select fields that you want in the report from **Available Fields**, and then click **(Move selected fields down)**. In addition to the fields, you can also select any tags that you have created and assigned.
   - Change the order of the fields in the report by clicking **(Move selected fields up)** or **(Move selected fields down)**.

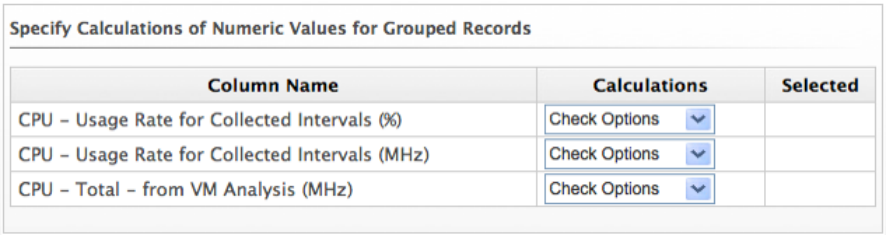
6. Click on the **Consolidation** tab to consolidate and aggregate data points into maximum, minimum, average, and total. Specifically, this would be useful for analyzing performance data over a specific period of time.

Note:

Please note that if you do this, you will not see individual records, but rather the calculation as a column header.



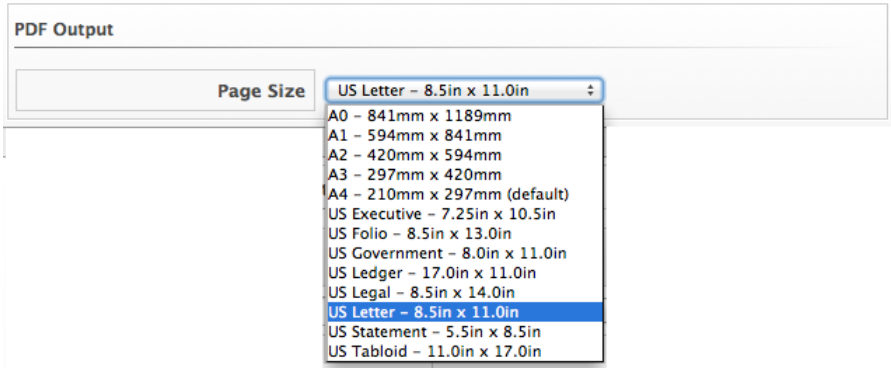
- Select the columns to group by.



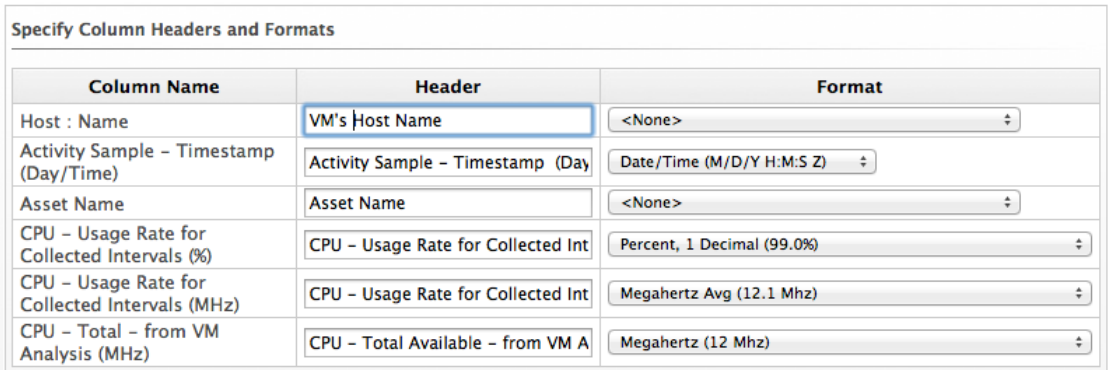
- For each numeric field selected in the report, you can click the dropdown under **Calculations**. Check the calculations you want to use.

7. Click on the **Formatting** tab to set the size of paper for a PDF and column header format.

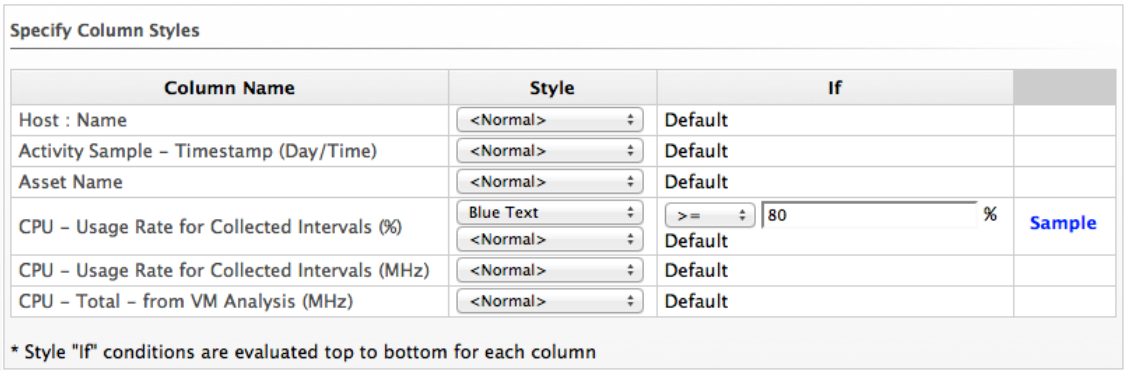
- From the **PDF Output** area, select the page size from the dropdown.



- From **Specify Column Headers and Formats**, type the text you want displayed for each field. For each numeric field, you can also set the numeric format.



8. Click on the **Styling** tab to change the color of the text or the background for a row based on a condition.



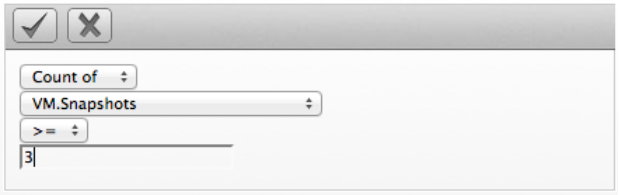
- Use **Style** to select the format for the value. You can choose to change the text color or the background.
- Use **If** to create a conditional statement for the style.

9. Click on the **Filter** tab to set filters for the data displayed in the report. There are two types of filters: the first is the **Record Filter** which is the primary filter of the main tables records, the second is also a **Display Filter**, which is a secondary filter of rows based on the fields of the child table. Click in the appropriate area to use the expression editor to choose the appropriate options for your criteria. Based on what you choose, different options will show.

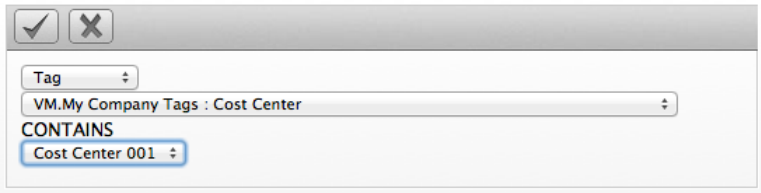
- Click **Field** to create criteria based on field values.



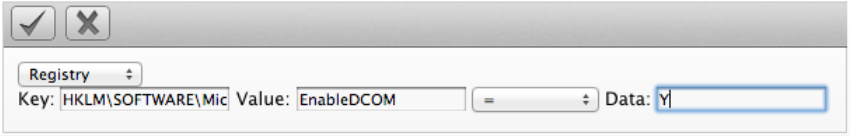
- Click **Count of** to create criteria based on the count of something, such as the number of snapshots for a virtual machine, or the number of virtual machines on a host.



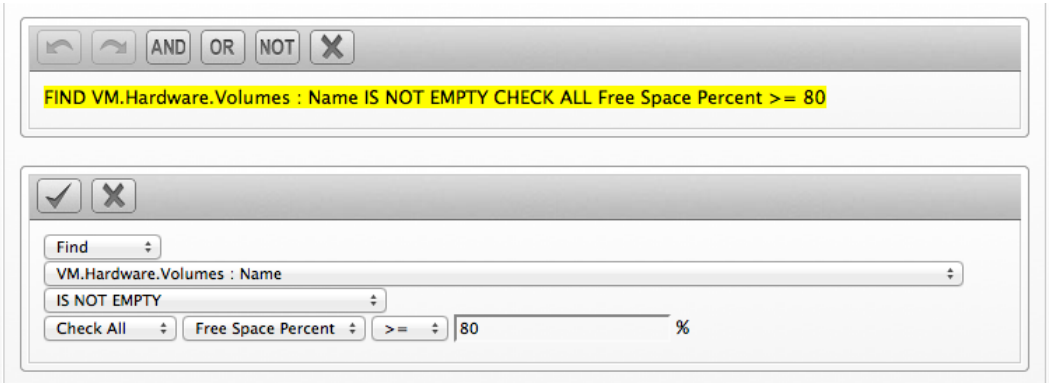
- Click **Tag** to create criteria based on tags assigned to your resources. For example, you may want to check the power state of a virtual machine or see if it is tagged as production.



- Click **Registry** to create criteria based on registry values. For example, you may want to check if DCOM is enabled on a Windows system. (Note that this applies only to Windows operating systems.)



- Click **Find** to seek a particular value, and then check a property. For example, finding the `Admin` account and checking that it is enabled.

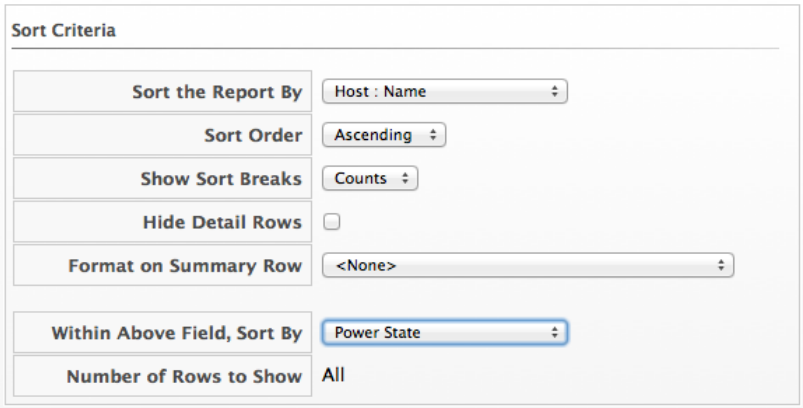


- Click ✓ **(Commit Expression Element Changes)** to add the expression.

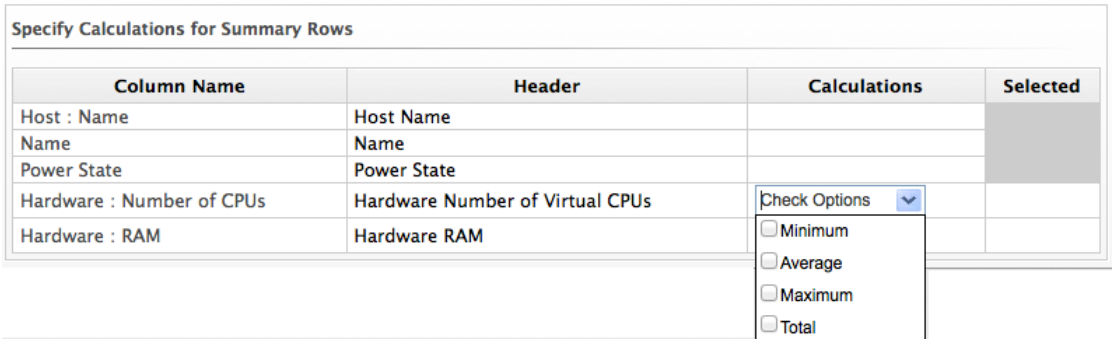
  Note:

  The filters that you apply will show at the bottom of the report so that you know which filters have been applied.

10. Click on the **Summary** tab to select sort order, sort type, groupings, and group calculations for the report. **Summary** groups rows of data based on the sort breaks. You can only sort by fields that display in the report.

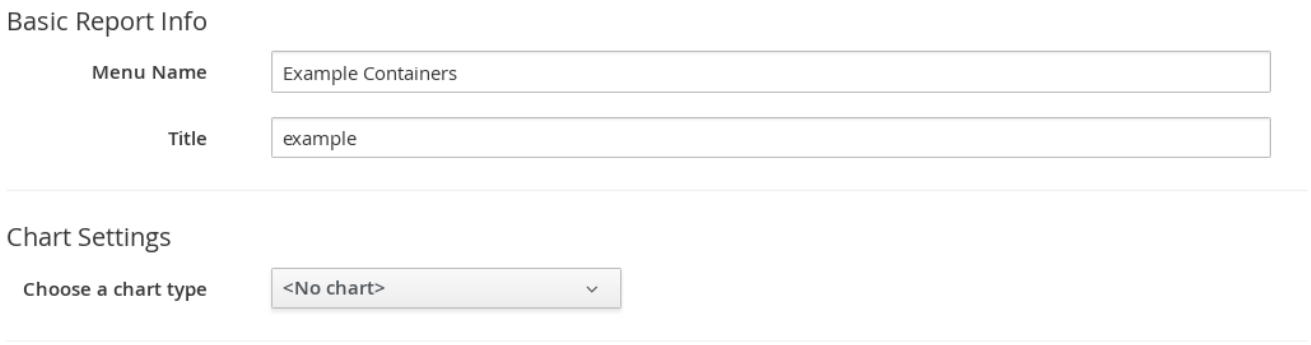


- Set the primary sort in **Sort the Report by**.
- Set the next sorts in **Within Above Field, Sort By**.
- Select the type of sort, ascending or descending, in **Sort order**.
- In **Show Sort breaks**, select **Yes** to show the sort breaks, **Counts** to show sort breaks with the count, or **No** for no sort breaks.



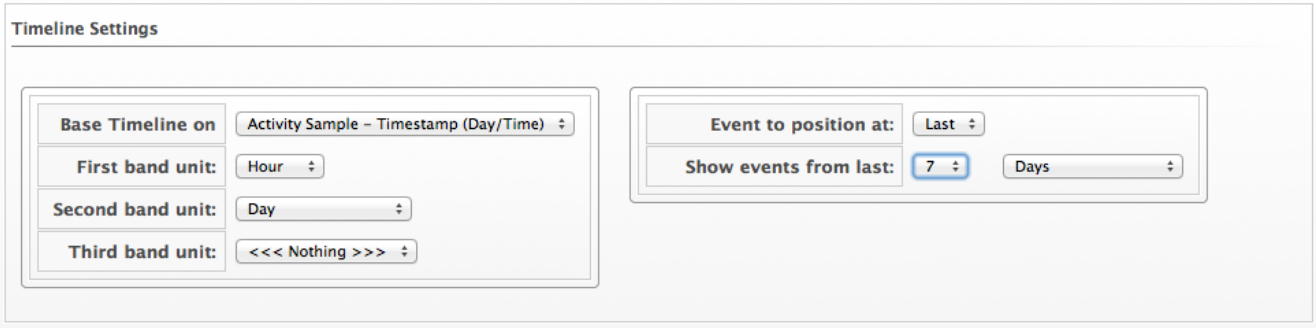
- For any numeric field, you can select to show minimum, average, maximum, and total in the sort break.

11. Click on the **Charts** tab to create a chart for the report. This is not required.



- Use **Choose a chart type** to select a type of chart. Note that some charts may not produce the result you are looking for based on the types of fields in the report and its sort order.
- If you only want to see the top values, select the number of top values from **Top values to show**.
- If you want to see the total number of values that are not categorized or evaluated against others, check **Sum 'Other' values**.

12. Click on the **Timeline** tab to select a timeline for the report. You must have a field of time or date format to use this feature.



- Use **Base Timeline on** to select a column in date or time format for the report.
- Select a unit of time for the first band in **First band unit**.
- Select a unit of time for the second band in **Second band unit**.
- Select a unit of time for the third band in **Third band unit**.
- Select an **Event to position at**.
- Select the range for the event to position from **Show events from last**.

13. Click the **Previews** tab to see a sample of your report.
14. When you have the report that you want, click **Add** to create the new report.

Note:

After the new report is created, to make the report accessible from the report menu, you must add it to a report menu.

Copying a Report

Use this feature to copy a report that is similar to one that you want to create. By copying a report, you only need to make minor edits instead of creating an entirely new report.

1. Navigate to **Overview > Reports**.
2. Click the **Reports > All Reports** accordion and select the report that you want to copy.
3. Click (**Configuration**), then click (**Copy this report**).
4. On the Columns tab, edit the **Basic Report Info** area to include a new **Menu Name**. Each **Menu Name** must be unique.
5. Make any other changes you need. See "Adding a Report" for details on the changes you can make.
6. Click **Add**.

Editing a Report

If you find that a report is not giving you the data that you need, you can edit it after it has been created. Note that only reports that you have created can be modified. Only administrators and super administrators of Infrastructure Automation can add, copy, edit, and delete reports.

1. Navigate to **Overview > Reports**.
2. Click the **Reports > All Reports** accordion and select the report you want to edit.
3. Click (**Configuration**), (**Edit this Report**).
4. Make any changes you need.
5. Click **Save**.

Deleting a Report

Delete reports when you find that they are no longer useful. Only administrators and super administrators of Infrastructure Automation can add, copy, edit, and delete reports.

Note:

Please note that only customer-created reports can be deleted.

1. Navigate to **Overview > Reports**.
2. Click the **Reports > All Reports** accordion and select the report you want to delete.
3. Click (**Configuration**), (**Delete this Report from the Database**).

   Note:

   The **Delete this report from the Database** option will only appear on reports you have created. Default reports cannot be deleted.
4. Click **OK**.

Importing a Report

Reports are stored in the VMDB; however, you can share customized reports among VMDBs. To do this use the export and import feature of reports.

1. Navigate to **Overview > Reports**.
2. Select **Custom Reports** from the **Import/Export** accordion.
3. In the **Import** area, click **Browse** to select an import file.
4. To overwrite an existing report with the same menu name, select **Overwrite existing reports**.
5. Click **Upload** to import the report to the **Company-Custom** folder.

Exporting a Report

Reports are stored in the VMDB. Custom reports may be shared among VMDBs. Share your custom reports using the export and import features.

1. Navigate to **Overview > Reports**.
2. Select **Custom Reports** from the **Import/Export** accordion.
3. In the **Export** area, select the reports to export.
4. Click **Export**.

Managing Report Menus

By default, all account roles have the same reports available. If you have the super administrator or administrator role, you can customize the accordions, the folders within them, and the locations of your reports.

Note:

When you create your own report, the report is not automatically available. You must add it to a report menu.

The report menu is a hierarchical structure that consists of the following components:

- The top level, under which the accordions are shown. (Top level is only displayed when you are in the report menu editor.)
- Accordions that are general categories for the reports. The defaults supplied are **Configuration Management**, **Migration Readiness**, **Operations**, **VM Sprawl**, **Relationships**, **Events**, **Performance by Asset Type**, **Trending**, **Tenants**, **Provisioning**, and **Running Processes**.
- Folders that are used to further organize reports within an accordion. For example, under the **Configuration Management**, there are folders for virtual machines, hosts, and other virtual infrastructure components.
- Reports that are stored directly in the folders.

Managing Report Menu Accordions

1. Navigate to **Overview > Reports**.

2. Click the **Edit Report Menus** folder.
3. Click the role whose menus you want to customize.
4. Click on **Top Level** to organize, add, and delete folders.
    - Click  **(Move selected folder to top)** to move the folder to the top of the list.
    - Click  **(Move selected folder up)** to move the folder up.
    - Click  **(Move selected folder down)** to move the folder down.
    - Click  **(Move selected folder to bottom)** to move the folder to the bottom of the list.
    - Click  **(Delete selected folder and its contents)** to delete a folder.
    - Click  **(Add folder to selected folder)** to add an folder.

      Note:

      If you are creating a new folder, **Top Level** must be selected under **Reports**. Be sure to select the folder you want to create a subfolder for on the left pane. To name the folder, double-click on **New Folder**, then click on **Top Level** in the **Reports** area.
5. When you are finished adding folders, click **Commit**. To revert, click **Discard**.
6. Click **Save**.

The new folder is added, and you can add folders in which to store reports to it. You can also organize the reports into folder that are meaningful to you.

Managing Report Menu Folders

1. Log in to the console as a user who is assigned either the super administrator or administrator account role.
2. Navigate to **Overview > Reports**.
3. Click the **Edit Report Menus** accordion.
4. Click the role whose menus you want to customize.
5. Click on the accordion name you want to organize or add folders to.
    - Click  **(Move selected folder to top)** to move the folder to the top of the list.
    - Click  **(Move selected folder up)** to move the folder up.
    - Click  **(Move selected folder down)** to move the folder down.
    - Click  **(Move selected folder to bottom)** to move the folder to the bottom of the list.
    - Click  **(Delete selected folder and its contents)** to delete an folder.
    - Click  **(Add subfolder to selected folder)** to add a folder. When creating a subfolder, be sure to select the folder that you want the subfolder to show under. To name the folder, double-click on **`New Folder`**.
6. When you are finished making changes click **Commit**. To revert, click **Discard**.
7. Click **Save**.

Note:

Only reports that are not already in another folder can be assigned.

Organizing Reports in Report Menus

1. Log in to the console as a user who is assigned either the Super Administrator or Administrator Account Role.
2. Navigate to **Overview > Reports**.
3. Click the **Edit Report Menus** accordion.
4. Click the role whose menus you want to customize.

5. Expand the **Report** accordion and menus using the triangles to the left of the item name until you are able to select the subfolder where you want to put reports.
6. Choose one of the following actions:
   - To add a report, select a report from the **Available Reports** area on the right and click  **(Move selected reports left)**.
   - To remove a report from a folder, select the report from the **Selected Reports** area and click  **(Move selected reports right)**.
   - To move a report to the top of the folder, select the report and click  **(Move selected reports to top)**.
   - To move a report up one place in the folder, select the report and click  **(Move selected reports up)**.
   - To move a report down one place in the folder, select the report and click  **(Move selected reports down)**.
   - To move a report to the bottom of the folder, select the report and click  **(Move selected reports to bottom)**.
7. When you are finished making changes click **Commit**.
8. Click **Save**.

The changes are made. The next time a user with this Account Role logs in, the new report and menu structure will be displayed.

Chargeback

The chargeback feature allows you to calculate monetary virtual machine charges based on owner or company tag. To use this feature you must be collecting capacity and utilization data.

Chargeback Rates

Infrastructure Automation provides a default set of rates for calculating chargeback costs, but you can create your own set of computing and storage costs by navigating to **Overview > Chargeback** and clicking the **Rates** accordion.

You can configure chargeback rates for various resources by selecting either **Compute** or **Storage** in the **Rates** accordion. **Compute** sets chargeback rates for CPU, disk I/O, fixed compute cost, memory, and network I/O items, while **Storage** sets chargeback rates for fixed storage cost and disk storage.

Chargeback costs are computed using a set formula based on hourly cost per unit and hourly usage.

Chargeback can be calculated in the following currencies:

- Australian dollar (AUD)
- Brazilian real (BRL)
- Swiss franc (CHF)
- Chinese yuan renminbi (CNY)
- Euro (EUR)
- Pound sterling (GBP)
- Indian rupee (INR)
- Japanese yen (JPY)
- South Korean won (KRW)
- Mexican peso (MXN)
- Norwegian krone (NOK)
- Russian ruble (RUB)
- Swedish krona (SEK)
- Turkish lira (TRY)
- United States dollar (USD)
- South African rand (ZAR)

Chargeback rates can be assigned at a single rate or by tiers, where rates are assigned in ranges depending on level of usage.

Additionally, chargeback can be calculated at one fixed rate, or by a combination of fixed and variable rates per tier. Fixed rates are charged once per unit of time, and the variable rate is calculated by the level of usage multiplied by the number of resources used in a unit of time.

Memory Used Cost

Calculating the Memory Used Cost in dollars ($) for a day can be expressed in the following ways:

- Memory allocation per hour (in MB) * Hourly Allocation cost per megabyte * Number of Memory Allocation metrics available for the day
- Sum of Memory allocation for the day (in MB) * Hourly Allocation cost per megabyte
- Sum of Memory allocation for the day (in MB) * Daily Allocation cost per megabyte / 24

Memory costs can be measured in B, KB, MB, GB, or TB.

In a scenario where 9.29 GB of memory is used in a day with the chargeback rate set at one dollar ($1) per megabyte per day, the Memory Used Cost would be $396.42.

- 9.29 GB = 9514.08 MB
- 9514.08 MB * $1 (per MB per day) = $9514.08
- $9514.08 / 24 = $396.42 Memory Used Cost

CPU Total Cost

The CPU Total Cost is defined as the number of virtual CPUs over the selected interval (hour, day, week, month). CPU costs can be measured in units of Hz, KHz, MHz, GHz, or THz, as specified when creating a chargeback rate.

In a scenario where 16 CPUs are used in a day with the chargeback rate set at one dollar per CPU per day, the CPU Total Cost would be $16.

- 16 CPUs * $1 (per CPU per day) = $16 CPU Total Cost

CPU Used Cost

The CPU Used Cost is defined as the average CPU used in MHz over the selected rate interval (hour, day, week, month). CPU Used Cost is not supported for containers providers.

In a scenario where 2.5 GHz is used in a day with the chargeback rate set at $0.01 per MHz per day, the CPU Used Cost would be $25.

- 2.5 GHz = 2500 MHz
- 2500 MHz * $0.01 (per MHz per day) = $25 CPU Used Cost

Storage Allocated Cost

The Storage Allocated Cost is defined as the Allocated Disk Storage in bytes over the selected rate interval (hour, day, week, month). Storage costs can be measured in B, KB, MB, GB, or TB.

In a scenario where 500 GB are used in a day with the chargeback rate set at $0.10 per GB per day, the Storage Allocated Cost would be $50.

- 536,870,912,000 bytes = 500 GB
- 500 GB * $0.10 (per GB per day) = $50 Storage Allocated Cost

Storage Total Cost

The Storage Total Cost is defined as the Used Disk Storage in bytes over the selected rate interval (hour, day, week, month).

In a scenario where 250 GB are used in a day with the chargeback rate set at $0.10 per GB per day, the Storage Total Cost would be $25.

- 268,435,456,000 bytes = 250 GB
- 250 GB * $0.10 (per GB per day) = $25 Storage Total Cost

Storage Used Cost

The Storage Used Cost is defined as the Used Disk Storage in bytes over the selected rate interval (hour, day, week, month).

In a scenario where 250 GB are used in a day with the chargeback rate set at $0.10 per GB per day, the Storage Used Cost would be $25.

- 268,435,456,000 bytes = 250 GB
- 250 GB * $0.10 (per GB per day) = $25 Storage Used Cost

Note:

The following chargeback rates are not supported for containers providers:

- Allocated CPU count
- Used CPU
- Used disk I/O
- Allocated memory

Creating Chargeback Rates

Infrastructure Automation allows you to create your own set of computing and storage costs to use for billing.

Chargeback rates can be configured at a single rate or in tiers, where one rate is assigned to one usage range, and another rate is assigned to a different usage range. You can also assign fixed and variable rates per tier if desired.

There are three methods to create a chargeback rate: editing or copying an existing rate, or creating a new rate.

Editing a Chargeback Rate

1. Navigate to **Overview > Chargeback**.
2. Click the **Rates** accordion and select an existing rate from under **Compute** or **Storage**.
3. Click **(Configuration)** and **(Edit this Chargeback Rate)**.
4. Edit the rate as desired.
5. Click **Save** at the bottom of the screen to save the chargeback rate.

Copying a Chargeback Rate

1. Navigate to **Overview > Chargeback**.
2. Click the **Rates** accordion and select an existing rate from under **Compute** or **Storage**.
3. Click **(Configuration)** and **(Copy this Chargeback Rate)**.
4. Edit the rate as desired.
5. Click **Add** at the bottom of the screen to save the chargeback rate.

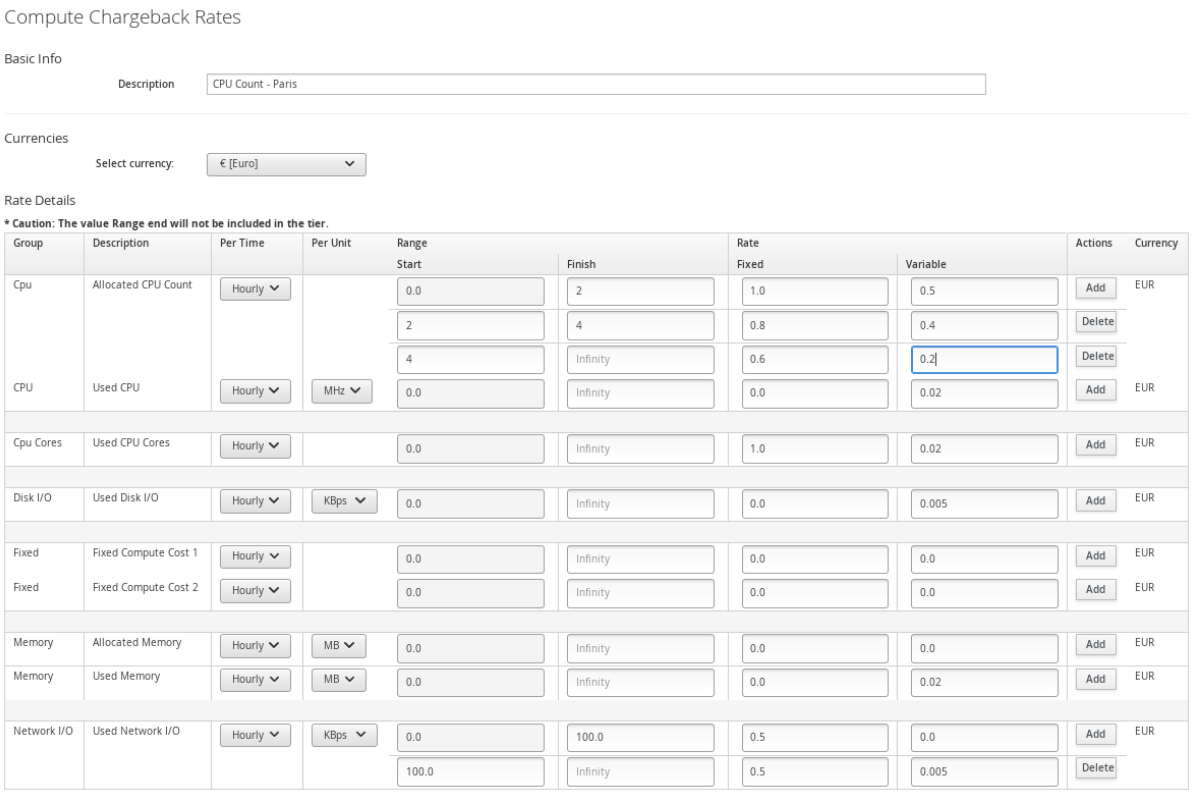
Adding a Chargeback Rate

The following example shows a scenario to create a new chargeback rate for allocated CPU usage in Euro per week measured by the following tiers, calculating both fixed and variable costs:

- Tier 1: 0-2 CPU = 1 Euro fixed + 0.5 Euro variable
- Tier 2: 2-4 CPU = 0.8 Euro fixed + 0.4 Euro variable
- Tier 3: 4 CPU and above = 0.6 Euro fixed + 0.2 Euro variable

To configure this rate:

1. Navigate to **Overview > Chargeback**.
2. Click the **Rates** accordion and select **Compute** to create a CPU chargeback rate.
    1. Click **(Configuration)** and **(Add a new Chargeback Rate)**.
3. Type in a **Description** for the chargeback rate.
4. Select **Euro** for currency.
5. In **Rate Details**, select the time, and fill in the range and rates for each tier. Alternatively, a single chargeback rate can be configured by using only one tier. For this example, in **Allocated CPU Count**, configure:
    1. For tier 1, set:
        1. In **Per Time**, select **Hourly**.
        2. In **Range**, specify the CPU range: **Start** remains as `0.0` and **Finish** is `2`.
        3. In **Rate**, specify `1` for the **Fixed** rate, and `0.5` for the **Variable** rate.
        4. Click **Add** under **Actions** to create a new, blank **Allocated CPU Count** line for tier 2.
    2. For tier 2, set:
        1. In the new line in **Range**, specify the CPU range: **Start** is `2` and **Finish** is `4`.
        2. In **Rate**, specify `0.8` for the **Fixed** rate, and `0.4` for the **Variable** rate.
        3. Click **Add** under **Actions** to create another new, blank **Allocated CPU Count** line for tier 3.
    3. For tier 3, set:
        1. In the new line in **Range**, specify the CPU range: **Start** is `4` and leave **Finish** as `Infinity`.
        2. In **Rate**, specify `0.6` for the **Fixed** rate, and `0.2` for **Variable**.



6. Click **Add** at the bottom of the screen to save the chargeback rate.

The rate can now be assigned.

Assigning Chargeback Rates

Infrastructure Automation allows you to assign chargeback rates by choosing from **Compute** and **Storage**. You can assign either a custom chargeback rate, or a default rate for Compute, Storage, or Container Images.

Note:

The **Default Container Image Rate** can be useful when applied to a containers provider with multiple images, where only some of the images have a chargeback rate assigned; the images without a custom rate will then use the default rate.

1. Navigate to **Overview > Chargeback**.
2. Click the **Assignments** accordion, and click either **Compute** or **Storage**.

Configuring Chargeback for Compute

Use **Compute** to assign a compute chargeback rate. You can assign chargeback rates to **The Enterprise**, **Selected Cloud/Infrastructure Providers**, **Selected Clusters/Deployment Roles**, **Selected Containers Providers**, **Tagged VMs and Instances**, **Tagged Images**, **Labeled Images**, or **Tenants**.

1. From the **Basic Info** area, use the **Assign To** list to select a type of assignee to assign the rate set to.
2. For **Tagged VMs and Instances** and **Tagged Images** select a **Tag Category**.
3. For each item to set, select the chargeback rate to use. The options displayed vary based on the type you select.
4. Click **Save**.

Configuring Chargeback for Storage

Use **Storage** to assign a storage chargeback rate. You can assign chargeback rates to **The Enterprise**, **Selected Datastores**, **Tagged Datastores**, or **Tenants**.

1. From the **Basic Info** area, use the **Assign To** list to select a type of assignee to assign the rate set to.
2. For **Tagged Datastores** select a **Tag Category**.
3. For each item to set, select the chargeback rate to use. The options displayed vary based on the type you select.
4. Click **Save**.

The rate is assigned. The next time you generate a chargeback report, these values will be used.

Note:

When viewing chargeback, there is a rate for a virtual machine for the number of the CPUs. The chargeback for this parameter is calculated based on when the virtual machine is running. If the virtual machine is not running, then it is not charged for CPU allocation.

Creating a Chargeback Report

Infrastructure Automation allows you to create chargeback reports to monitor costs you charged.

Note:

When creating a chargeback report, the **Total Cost** is the complete cost for the chosen workload over the selected time frame. Total costs are calculated by adding the calculated costs for the amount used and the amount allocated for the time frame. When adding cost related columns to display on your report, be sure to select all rate assigned fields to present a comprehensive accounting of total costs.

1. Browse to menu: **Overview > Reports**.
2. Click the **Reports** accordion.
3. Click **Configuration**, then click **Add a new Report**.
4. On the **Columns** tab, fill out the **Basic Report Info** area.
   - Type a unique name in **Menu Name** for how you want the report described in the menu list.
   - Type the **Title** to display on the report.
5. Add fields in the **Configure Report Columns** area.
   - From the **Base the report on** list, select **Chargebacks**.
   - Select the fields to include in the report from the **Available Fields** list, then click **Move selected fields down**. In addition to the fields, you can also select any tags that you have created and assigned.
   - Change the order of the fields in the report by clicking **Move selected fields up** or **Move selected fields down**.
6. Click the **Formatting** tab to set the size of paper for a PDF and column header format.
   - From the **PDF Output** area, select the page size from the **Page Size** list.
   - From **Specify Column Headers and Formats**, type the text to display for each field. For each numeric field, you can also set the numeric format.
7. Click the **Filter** tab to set filters for the data displayed in the report.
   - From **Chargeback Filters**, select how you want the costs to show, the tag category, the tag, and how you want the items grouped.
   - From **Chargeback Interval**, select the time interval. You must have a full interval worth of data in order to select an option other than **Partial** in the **Daily Ending With** list.
8. Click the **Preview** tab, and then **Load** to see what the report will look like.
9. When you are satisfied that you have the report that you want, click **Add** to create the new report.

The new report is created. To make the report accessible from the **Report** menu, you must add it to a report menu.

Optimization

Optimization provides administrators with a series of reports to provide meaningful information on areas for optimizing resources such as CPU, memory and disk and identify potential savings.

Running an Optimization Report

1. Navigate to **Overview > Optimization**.
2. Click **Queue Report** for each report type you choose to run.

Viewing Optimization Reports

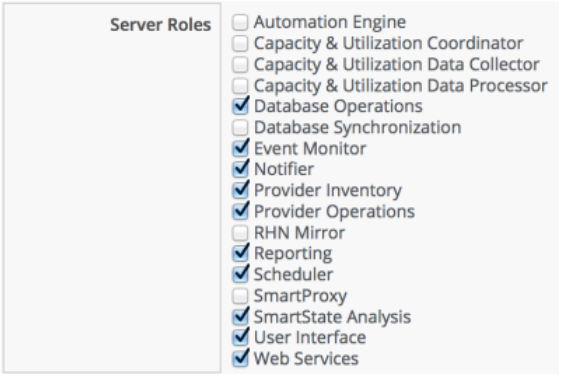
1. Navigate to **Overview > Optimization**.
2. Click on the report type you wish to view details for.
3. On the report type screen, click on a report instance to view details.
4. Sort reports by clicking on the field names at the top of the report.

Alerts

Assigning the Notifier Role

1. Click (**Configuration**).
2. Click the **Settings** accordion, and select the Infrastructure Automation server.

3. From the **Server Control** tab, select the **Notifier** role.



4. Click **Save**.

Creating an Alert

This section outlines the basic procedure to create an alert.

Note:

To send emails or SNMP traps from the Infrastructure Automation server, you must enable the Notifier server role and set up SMTP email or SNMP traps. For more information, see *General Configuration*.

1. Navigate to **Control > Explorer**.
2. Click the **Alerts** accordion, then click (**Configuration**), (**Add a New Alert**).
3. Enter the basic details of the alert:
    1. Enter a description in the **Description** field.
    2. Select the **Active** check box to enable the alert after creation.
    3. Select the severity level from the **Severity** list.
    4. Select the inventory item on which to base the alert from the **Based On** list.
    5. Select the type of event that triggers the alert from the **What to Evaluate** list.
    6. Select the frequency with which to be notified if the event log threshold is reached from the **Notification Frequency** list.
4. Configure the parameters of the alert.

    Note:

    The available parameters depend on the options you selected in the **Based On** and **What to Evaluate** lists. See later sections for the details of these parameters.
5. Optionally, select **Send an E-mail** to configure options so that an email is sent when the alert is triggered:
    1. Enter the email address from which to send the email in the **From** field.
    2. Select a user from the **Add a User** list to add a user registered in Infrastructure Automation. The user must have a valid email address entered under accounts.
    3. Enter the email address of a user in the **Add (enter manually)** field and click + to add a user not registered in Infrastructure Automation.
6. Optionally, select **Send an SNMP Trap** to configure options so that an SNMP trap is sent when the alert is triggered:
    1. Enter the IP addresses of the hosts to send the trap in the **Host** fields.
    2. Select the version of SNMP to use from the **Version** list:
        1. If you select **v1**, enter a trap number in the **Trap Number** field. Enter **1**, **2**, or **3**, based on the appropriate suffix number in table_title.
        2. If you select **v2**, enter a trap object ID in the **Trap Object ID** field. Enter **info**, **warning**, or **critical** based on the values in table_title.

Object ID	Suffix Number Added to PEN	PEN with the Suffix Added
info	1	1.3.6.1.4.1.33482.1
warn, warning	2	1.3.6.1.4.1.33482.2
crit, critical, error	3	1.3.6.1.4.1.33482.3

SNMP Trap Identifiers

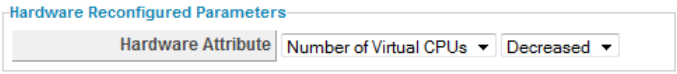
7. Optionally, select **Send a Management Event** to trigger an automation event:
    1. Enter the name of the event that exists in the **Process/Event Class** in the **Event Name** field.

8. Click **Add**.

Creating a Hardware Reconfigured Alert

Use a hardware reconfigure alert to detect changes to the amount of memory or the number of CPUs on a virtual machine.

1. Navigate to **Control > Explorer**.
2. Click the **Alerts** accordion, then click (**Configuration**), (**Add a new Alert**).
3. In the **Info** area:
   - Type in a description for the alert.
   - From **Based On**, select **VM and Instance**.
   - From **What to Evaluate**, select **Hardware Reconfigured**.
   - In **Notification Frequency**, select how often you want to be notified if hardware reconfiguration is detected.
4. From **Hardware Attribute**, select Number of CPUs. From the next dropdown, select **Decreased**.



5. After setting the parameters, select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.
6. Click **Add**.

Creating a Normal Operating Range Alert

Normal operating range alerts enables you to be notified when the normal operating range is exceeded, or falls below for a period of time from 1 minute to 2 hours. Capacity and utilization must be enabled for normal operating ranges to be calculated. See *General Configuration* for more information.

1. Navigate to **Control > Explorer**.
2. Click the **Alerts** accordion, then click (**Configuration**), (**Add a new Alert**).
3. In the **Info** area:
   - Type in a **Description** for the alert.
   - From **Based On**, select **VM and Instance**.
   - For **What to Evaluate**, select **Normal Operating Range**.
   - In **Notification Frequency**, select how often you want to be notified if the performance threshold is reached.
4. Set the threshold in the **Normal Operating Range Parameters** area.



   - From **Performance Field**, select the field to check and whether you want to be notified if the field is exceeded or fell below.
   - In **Field Meets Criteria for**, select the amount of time that the threshold requires to be met to trigger the alert.
5. After setting the parameters, you then select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process. See Creating an Alert.
6. Click **Add**.

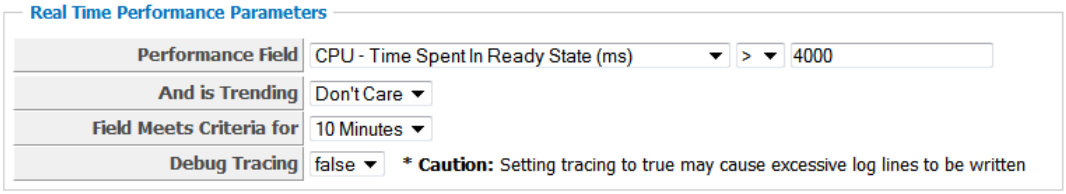
Creating a Real Time Performance Alert

Real Time Performance alerts enables you to be notified immediately when a performance threshold has been met for a virtual machine, host, or cluster. Capacity and Utilization must be enabled for performance thresholds to be detected.

1. Navigate to **Control > Explorer**.
2. Click the **Alert** accordion, then click (**Configuration**), (**Add a new Alert**).
3. In the **Info** area:
   - Type in a **Description** for the alert.
   - From **Based On**, select **VM and Instance**.
   - For **What to Evaluate**, select **Real Time Performance**.

- In **Notification Frequency**, select how often you want to be notified if the performance threshold is reached.

4. Set the threshold in the **Real Time Performance Parameters** area.



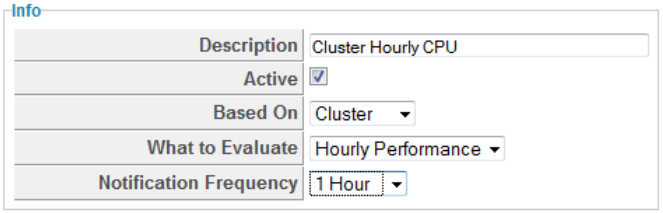
- From **Performance Field**, select the field to check and any other parameters required for that field.
- In **And is Trending**, select **Don't Care** if it does not matter how the performance metric is trending. Otherwise, choose from the possible trending options.
- In **Field Meets Criteria for**, select the amount of time that the threshold requires to be met to trigger the alert.
- Set **Debug Tracing** to true only when directed to do so by Red Hat Support. This provides an extremely detailed level of logging and can result in many more log lines being written.

5. After setting the parameters, you then select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.
6. Click **Add**.

Creating an Hourly Performance Alert

Hourly performance alerts enable you to be notified immediately when an hourly performance threshold has been met for a cluster. Capacity and Utilization must be enabled for performance thresholds to be detected.

1. Navigate to **Control > Explorer**.
2. Click the **Alerts** accordion.
3. Click (**Configuration**), (**Add a new Alert**).
4. In the **Info** area:



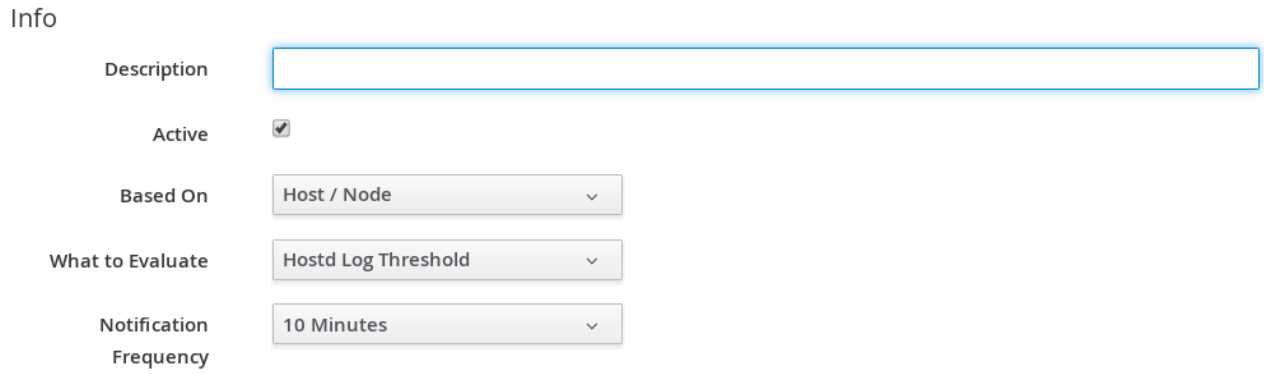
- Type in a **Description** for the alert.
- From **Based On**, select **Cluster**.
- For **What to Evaluate**, select **Hourly Performance**.
- In **Notification Frequency**, select how often you want to be notified if threshold is met.

5. In the **Hourly Performance Parameters** area select performance field and the criteria. You can also select options from the **And is Trending** dropdown box and whether the **Debug Tracing** is true or false.
6. After setting the parameters, you then select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.
7. Click **Add**.

Creating a hostd Log Threshold Alert

Use the **hostd Log Threshold** to send a notification when certain items are found in the event logs for a host. A default analysis profile with event log items is required for this feature. The following example shows steps to check the host's log for a failure to validate a virtual machine's IP address.

1. Navigate to **Control > Explorer**.
2. Click the **Alert** accordion.
3. Click (**Configuration**), (**Add a new Alert**).
4. In the **Info** area:



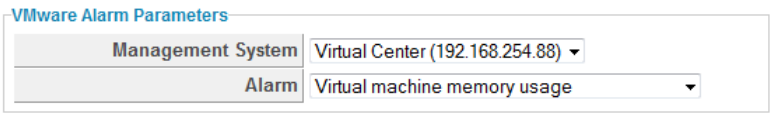
- Type in a **Description** for the alert.
- From **Based On**, select **Host/Node**.
- For **What to Evaluate**, select **Hostd Log Threshold**.
- In **Notification Frequency**, select how often you want to be notified if the log item is detected.

5. In the **Hostd Log Threshold Parameters** area, select the parameters for the event log message. You can set a threshold for a filter, level, or message source.
   - Use **Message Filter** to look for specific text in a message. Use **Message Level** to filter based on message level. Infrastructure Automation reports on the specified level and above. Use **Message Source** to filter log messages based on its source.
   - Set **How Far Back to Check** in days you want to look for this message.
   - If you only want an alert triggered when the log message has occurred a certain number of times, type the number in **Event Count Threshold**.
6. After setting the parameters, select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.
7. Click **Add**.

Creating a VMware Alarm Alert

Infrastructure Automation can use VMware alarms as a trigger for an alert. This type of alert can be created for a cluster, host, or virtual machine.

1. Navigate to **Control > Explorer**.
2. Click the **Alerts** accordion, then click (**Configuration**), (**Add a new Alert**).
3. In the **Info** area:
   - Type in a description for the alert.
   - From **Based On**, select **Cluster**, **Host**, or **VM**.
   - For **What to Evaluate**, select **VMware Alarm**.
   - In **Notification Frequency**, select how often you want to be notified if the log item is detected.
4. In the **VMware Alarm Parameters** area select the provider and alarm.

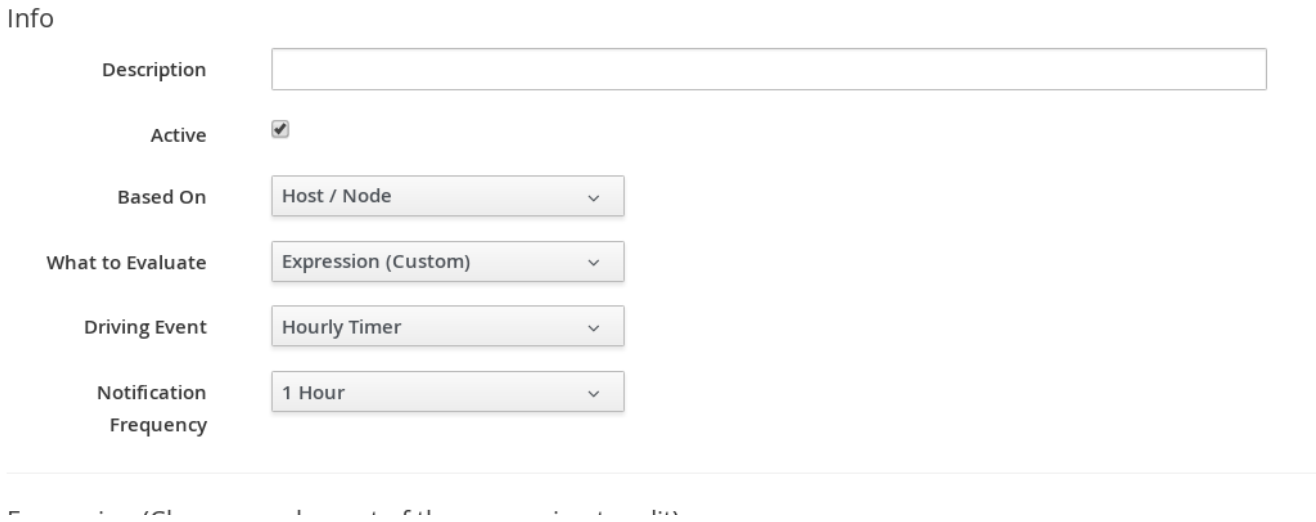


5. After setting the parameters, you then select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.
6. Click **Add**.

Creating an Expression Alert

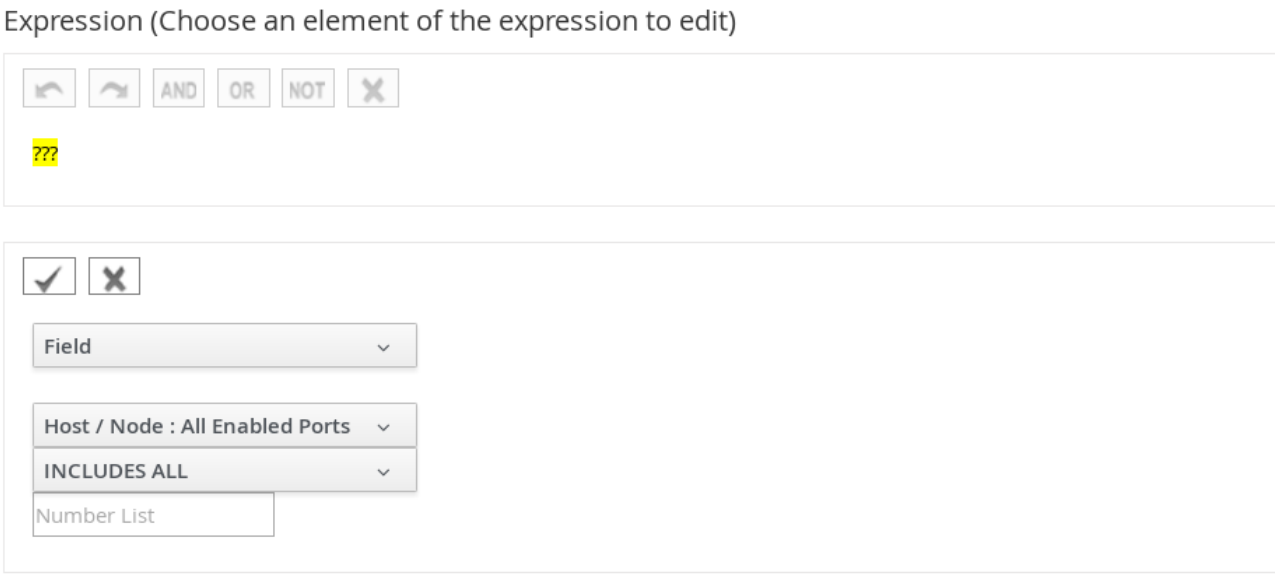
Expression alerts enables you to create a notification based on any possible criteria for clusters, datastores, hosts, and virtual machines. The following procedure creates an alert for when a host's datastore has less than 5% free space.

1. Navigate to **Control > Explorer**.
2. Click on the **Alerts** accordion, then click (**Configuration**), (**Add a new Alert**).
3. In the **Info** area:



- Type in a description for the alert.
- From **Based On**, select **Host/Node**.
- For **What to Evaluate**, select **Expression (Custom)**.
- In **Notification Frequency**, select how often you want to be notified if the expression is evaluated to true.

4. Use the expression editor to create your expression. This is the same expression editor used to create **Conditions**.



5. Click ✓ (**Commit expression element changes**) to accept the expression.
6. After setting the parameters, you then select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.
7. Click **Add**.

Create an Operational Alert

1. Navigate to **Control > Explorer**.
2. Click on the **Alerts** accordion, then click ⚙ (**Configuration**), ⊕ (**Add a new Alert**).
3. In the **Info** area:
   - Type in a description for the alert.
   - Check **Active** when you feel that the alert is ready to be enabled.
   - From **Based On**, select **Server**.
   - Select the appropriate driving event.

- In **Notification Frequency**, select how often you want to be notified if the event log threshold is reached.

4. After setting the parameters, select what you want the alert to do. You can send an email, create an SNMP Trap, or send a management event to start an automation process.

5. Click **Add**.

Operational Alert Types

Operational Alerts

Driving Event	Explanation (Thresholds, Description)	Proposed Action and Next Steps
EVM Server Start	Alert is raised when an server starts.	Send e-email. This is a notification.
EVM Server Stop	Alert is raised when a server stops.	Send e-mail. Review logs to see why the server stopped.
EVM Server Not Responding	Alert is raised when one server detects that another server has not responded in 2 minutes.	This is a sign of a problem that should be investigated. Check logs.
EVM Server Exceeded Memory Limit	Alert is raised when an server has exceeded its system memory limit and begins killing workers. Default is 80%. Threshold configured in Advanced Settings. server: :worker_monitor: :kill_algorithm: :name: :used_swap_percent_gt_value :value: 80	This may be caused by the following issues: The server is running with too few resources. The server is enabled with too many roles or number of workers. The server picked up all the roles because another server has failed. A runaway process has taken up most of the memory.
EVM Server is Master	When one server takes over as a master server.	Typically, this should only occur when first starting a set of servers, perhaps following expected outages. If a server picks up as master in other situations, the old master had an issue that needs to be researched (such as server not responding in time).
EVM Server High System Disk Usage	The server's system disk is 80% full. This check is run as part of a system schedule. Threshold configured in Advanced Settings. server: events: :disk_usage_gt_percent: 80	Temp files used by the operating system may be filling the disk. Yum updates, normal /tmp files, or temp files in /var/lib/data/miqtemp/ may cause the problem.
EVM Server High App Disk Usage	The server's app disk is 80% full. This check is run as part of a system schedule. Threshold configured in Advanced Settings. server: events: :disk_usage_gt_percent: 80	Server temp files may remain.
EVM Server High Log Disk Usage	The server's log disk is 80% full. This check is run as part of a system schedule. Threshold configured in Advanced Settings. server: events: :disk_usage_gt_percent: 80	Logs are getting too big or are not being log rotated properly every day. Review the most recent logs.
EVM Server High DB Disk Usage	The server's db disk is 80% full. This check is run as part of a system schedule. Applies if using PostgreSQL as the VDMB. Threshold configured in Advanced Settings. server: events: :disk_usage_gt_percent: 80	Database or database logging is getting too large. May require **full** vacuuming of PostgreSQL database.
EVM Worker Started	Alert is raised when a worker is about to start.	This is a notification. Failover may trigger this alert.
EVM Worker Stopped	Alert is raised when a worker is requested to stop.	Review logs for reason in the event the worker was not purposefully stopped.
EVM Worker Killed	Alert is raised when a non- responsive worker does not restart on its own and is killed.	Review logs for reason the worker was killed. May be the result of EVM Worker Not Responding.

Driving Event	Explanation (Thresholds, Description)	Proposed Action and Next Steps
EVM Worker Not Responding	Alert is raised when a worker has not responded for 2 minutes (:heartbeat_timeout) or has not started within 10 minutes (:starting_timeout).	An influx of events from the Virtual Center or Red Hat Virtualization causes an inability of EVM/CFME to handle the capacity at which they are being queued. Utilize the Event Handler Configuration to filter events that are causing problematic queue table growth.
EVM Worker Exceeded Memory Limit	Alert is raised when a worker exceeds the memory threshold. The default is 150 MB, but some workers have their own value in the :memory_threshold section for that specific worker.	Review logs for reason the worker is exceeding the memory limit. This may be the result of an overload to the worker process that requires further investigation.
EVM Worker Exceeded Uptime Limit	Alert is raised when a worker has been running longer than the :restart_interval. (Most workers are set to never restart using the 0.hours setting.) The EMS Refresh SmartProxy workers are set to restart every 2 hours.	Review logs for the reason the worker is exceeding the limit. This may be the result of an overload to the worker process that needs further investigation.
EVM Worker Exit File	Alert is raised when the scheduler worker exits due to a pending large ntp time change.	This is a notification.

Editing an Alert

After creating an alert, you can edit the threshold, expression, or the notification type.

1. Navigate to **Control > Explorer**.
2. Click on the **Alerts** accordion, then click on the alert that you need to edit.
3. Click **(Configuration)**, **(Edit this Alert)**.
4. Make the required changes.
5. Click **Save**.

Copying an Alert

You can copy an existing alert to create a new alert that is similar to the existing one, then change the values associated with it.

1. Navigate to **Control > Explorer**.
2. Click on the **Alert** accordion, then click on the alert that you want to copy.
3. Click (**Configuration**), (**Copy this Alert**). Click **OK** to confirm.
4. Make the required changes.
5. Click **Add**.

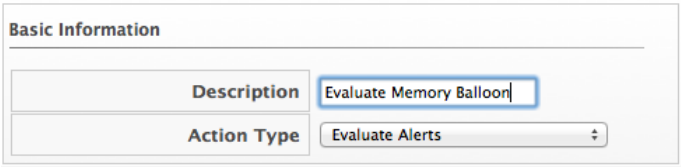
Deleting an Alert

When an alert is no longer needed, you can remove it from your VMDB.

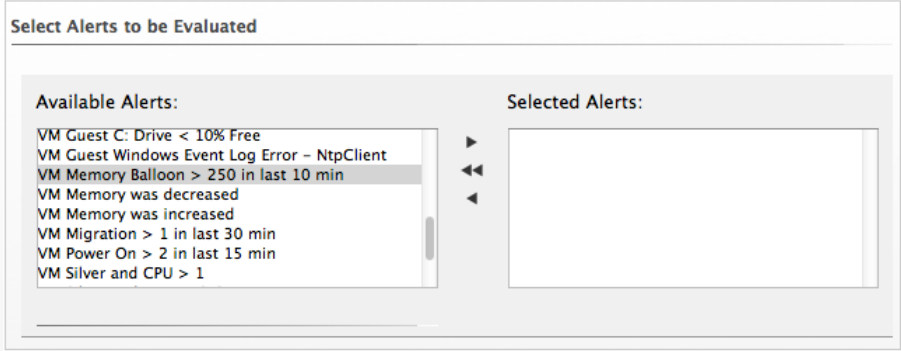
1. Navigate to **Control > Explorer**.
2. Click on the **Alerts** accordion, then click on the alert that you want to delete.
3. Click (**Configuration**), (**Delete this Alert**).
4. Click **OK** to confirm.

Evaluating an Alert

1. Navigate to **Control > Explorer**.
2. Click the **Actions** accordion, then click (**Configuration**), (**Add a new Action**).
3. Type in a **Description** for the action.



4. Select **Evaluate Alerts** from **Action Type**.
5. Select the alerts to be evaluated and click (**Move selected Alerts into this Action**). Use the `Ctrl` key to select multiple alerts.



6. Click **Add**.

Alert Profiles

Creating Alert Profiles

Alert profiles enable you to create groups of standard alerts. An alert profile can have as many alerts assigned as you need, and can be assigned to clusters, datastores, hosts, and virtual machines.

Note:

Each profile type under the **Alert Profile** accordion menu will have its own **Alert Selections** and subsequent **Assignments** choices.

1. Navigate to **Control > Explorer**.
2. Click on the **Alert Profiles** accordion, then click on the type of profile that you want to create.
3. Click (**Configuration**), (**Add a new Profile**).
4. In the **Basic Information** box, type in a unique **Description** for the alert profile.
5. Select the desired alerts from the **Alert Selection** area. Use the `Ctrl` key to select multiple alerts.
6. Click to add the Alerts.
7. Type in any additional description in the **Notes** area.
8. Click **Add**.

Editing an Alert Profile

You can edit an alert profile as your enterprise's needs change.

1. Navigate to **Control > Explorer**.
2. Click on the **Alert Profiles** accordion, then click the alert profile you want to edit.
3. Click (**Configuration**), (**Edit this Alert Profile**).
4. Make the required changes.
5. Click **Save**.

Deleting an Alert Profile

Remove alert profiles that you no longer need. This does not remove the alerts associated with the alert profile.

1. Navigate to **Control > Explorer**.
2. Click on the **Alert Profiles** accordion, then click the alert profile you want to remove.
3. Click (**Configuration**), (**Delete this Alert Profile**).
4. Click **OK** to confirm.

Assigning an Alert Profile

After an alert profile is created and verified, you can assign it directly to a resource.

1. Navigate to **Control > Explorer**.
2. Click on the **Alert Profiles** accordion, then click on the alert profile that you want to assign.
3. Click (**Configuration**), (**Edit Assignments for this Alert Profile**).

4. The options presented change based on if the alert is for a cluster, datastore, Infrastructure Automation server, host, or virtual machine and instance. You can assign to the enterprise, to specific hosts, cluster, resource pools, and providers, or based on assign tags. For a Infrastructure Automation server alert profile, you can only assign to Infrastructure Automation servers in the current Region.
5. Click **Save**.

Importing and Exporting

Importing and Exporting Policies, Policy Profiles, and Alerts

If you have multiple VMDBs, you can export policies, policy profiles, or alerts from one to another. You can export and import for use with other Infrastructure Automation infrastructures.

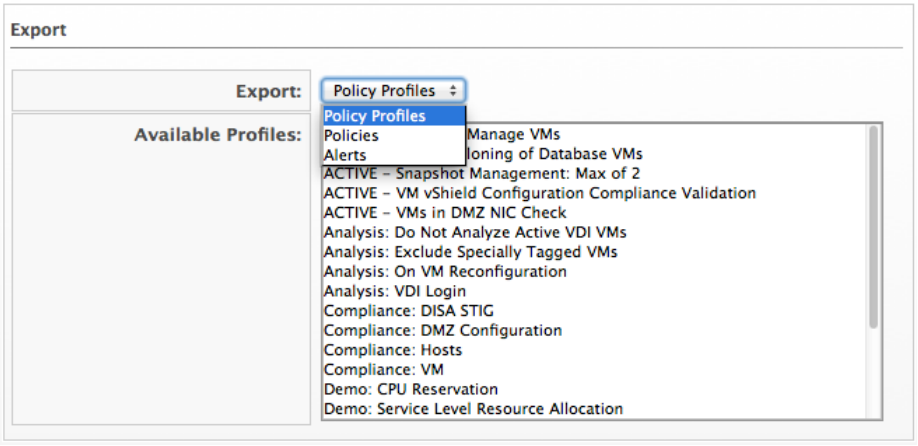
Note:

The appliances you are exporting or importing data between must use the same Infrastructure Automation version.

1. Copy the file to import to a location that is accessible to your Infrastructure Automation console.
2. Navigate to **Control > Import/Export**.
3. Click **Choose file** to navigate to the location of the file.
4. Select the file, and then click **Open** from the file selection box.
5. Click **Upload**.
6. Verify that these are the policies or policy profiles that you want to import.
7. Click **Commit**.

Exporting a Policy, Policy Profile, or an Alert

1. Navigate to **Control > Import/Export**.
2. From the **Export dropdown**, select policy profiles, policies, or alerts, depending on what you want to export.



3. From the **Available Profiles** or **Available Policies** or **Available Alerts** list, select the items to export. Use the `Ctrl` key to select multiple items to export into one file.
4. Click **Export**.
5. Follow the prompts in your browser to save the file.

OpenShift Prometheus Alerts

Note:

This feature is currently available as a technology preview only. For more information on the support scope for features marked as technology preview, see Technology Preview Features Support Scope.

Prometheus is used as an external alerting component. Infrastructure Automation integrates Prometheus alerts for use with OpenShift Container Platform. Infrastructure Automation collects events from Prometheus, generates alerts based on these events, and then attaches alerts to inventory objects.

It is possible to view ongoing alerts in Infrastructure Automation by navigating to **Monitor > Alerts** and manage their life cycle, including the ability to:

- View active alerts per provider in the **Overview** screen.
- View alert data and related objects.
- Comment, assign, and acknowledge alerts in the **All Alerts** screen.

Network utilization metrics for OpenShift Container Platform are only collected for `eth0` network interfaces. Therefore, network utilization trend will not be displayed for network interfaces other than `eth0` on the overview screen of the provider.

Configuring Prometheus Alerts in Infrastructure Automation

Configuring and enabling Prometheus alerts in Infrastructure Automation comprises the following steps:

1. Deploying and configuring Prometheus on OpenShift.
2. Assigning Prometheus alert profiles to the enterprise in Infrastructure Automation.
3. Adding an OpenShift Containers Provider with a Prometheus alert endpoint in Infrastructure Automation.

Deploying and Configuring Prometheus on OpenShift

1. Deploy Prometheus on OpenShift by following the steps covered in Prometheus on OpenShift Container Platform.
2. Add Prometheus to an OpenShift cluster and configure alert definitions. See the example code block below for configuring alerts (currently must be done on the Prometheus side):

```
$ oc edit configmap -n openshift-metrics prometheus
# Supported labels:
# severity: ERROR|WARNING|INFO. defaults to ERROR.
# Supported annotations:
# description: The error message displayed on the screen
# miqTarget: ContainerNode|ExtManagementSystem, defaults to ContainerNode.
# miqIgnore: "true|false", should ManageIQ pick up this alert, defaults to true.
  prometheus.rules: |
    groups:
    - name: example-rules
      interval: 30s # defaults to global interval
      rules:
      - alert: "NodeDown" << Alert starts here
        expr: up{job="kubernetes-nodes"} == 0
        annotations:
          miqTarget: "ContainerNode"
          severity: "ERROR"
          url: "https://www.example.com/node_down_fixing_instructions"
          description: "Node  is down"
      - alert: "TooManyRequests"
        expr: rate(authenticated_user_requests[2m]) > 12
        annotations:
          miqTarget: "ExtManagementSystem"
          severity: "ERROR"
          url: "https://www.example.com/too_many_requests_fixing_instructions"
          description: "Too many authenticated requests"
```

3. Reload the Prometheus configuration. You can do this by deleting the Prometheus pod, then redeploying with the new configuration.

Assigning Prometheus Alert Profiles to the Enterprise

Complete the following procedure to assign Prometheus alert profiles to the enterprise using the Infrastructure Automation user interface.

Note:

Both Node and Provider alert profiles are created automatically during the installation, so it is not required to create these profiles.

1. Navigate to **Control > Explorer**, then click **Alert Profiles** in the accordion menu.
2. Click to expand **Node Alert Profiles**, then click **Prometheus Node Profile**.
3. Click  (**Configuration**), then click  (**Edit assignments for this Alert Profile**). Assign the profile to **The Enterprise** from the **Assign To** list.
4. Click **Save**.
5. Click to expand **Provider Alert Profiles**, then click **Prometheus Provider Profile**.
6. Click **Configuration**, then click **Edit assignments for this Alert Profile**. Assign the profile to **The Enterprise** from the **Assign To** list.
7. Click **Save**.

   Note:

   Alerts triggered before assigning the profiles to the enterprise will not appear in Infrastructure Automation at all.

Adding an OpenShift Containers Provider with a Prometheus Alert Endpoint

Complete the following procedure to add an OpenShift Containers Provider with a Prometheus Alert Endpoint using the Infrastructure Automation user interface.

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click  (**Add a New Containers Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **OpenShift Container Platform**.

5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **default**.
6. In the **Default** tab, configure the following for the OpenShift provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.

          Note:

          You can obtain your OpenShift Container Platform provider's CA certificate for all endpoints (default, metrics, alerts) from **/etc/origin/master/ca.crt**. Paste the output (a block of text starting with **-----BEGIN CERTIFICATE-----**) into the **Trusted CA Certificates** field.
        - **SSL without validation**: Authenticate the provider insecurely (not recommended).
    2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.

       Important:

       The **Hostname** must use a unique fully qualified domain name.
    3. Enter the **API Port** of the provider. The default port is **6443**.
    4. Enter a token for your provider in the **Token** box.

       Note:

       To obtain a token for your provider, run the **oc get secret** command on your provider; see Obtaining an OpenShift Container Platform Management Token.

       For example:

       # oc get secret –namespace management-infra management-admin-token-8ixxs –template='' | base64 –decode
    5. Click **Validate** to confirm that Infrastructure Automation can connect to the OpenShift Container Platform provider.
7. In the **Metrics** endpoint tab optionally configure the metrics service details:
    1. Select a service type (e.g.: Prometheus) or leave disabled
    2. Select a **Security Protocol** method to specify how to authenticate the service:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
        - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)
    3. Enter the **Hostname** (or IPv4 or IPv6 address) of the metrics service.

       In order to find the hostname you can use the **oc get route** command.

       You can retrieve the Prometheus route by running:

       ```
       oc get route prometheus-k8s -n openshift-monitoring
       ```

       Example:

       ```
       oc get route prometheus-k8s -n openshift-monitoring
       NAME              HOST/PORT                                          PATH              SERVICES   PORT
       TERMINATION          WILDCARD
       prometheus-k8s    prometheus-k8s-openshift-monitoring.ocp.example.com  prometheus-k8s    web
       reencrypt/Redirect    None
       ```

       In this example you would use **prometheus-k8s-openshift-monitoring.ocp.example.com** as the metrics hostname.
    4. Enter the **API Port** of your metrics endpoint uses a non-standard port for access. The default port is **443**.
    5. Click **Validate** to confirm that Infrastructure Automation can connect to the metrics endpoint.
8. In the **Virtualization** endpoint tab optionally configure the virtualization service details:
    1. Select a service type (OpenShift Virtualization) or leave disabled
    2. Select a **Security Protocol** method to specify how to authenticate the service:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.

- **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
- **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)

3. Enter the **Hostname** (or IPv4 or IPv6 address) of the virtualization service.
4. Enter the **API Port** of your virtualization service. The default port is **`443`**.
5. Enter a token for your provider in the **Token** box.

9. Click the **Advanced** tab to add image inspector settings for scanning container images on your provider using OpenSCAP.

Note:

- These settings control downloading the image inspector container image from the registry and obtaining the Common Vulnerabilities and Exposures (CVE) information (for effective scanning) via a proxy.
- CVE URL that Infrastructure Automation requires to be open for OpenSCAP scanning: https://www.redhat.com/security/data/metrics/ds/. This information is based on the source code of OpenSCAP.

1. Enter the proxy information for the provider in either **HTTP**, **HTTPS**, or **NO Proxy** depending on your environment.
2. Enter the **Image-Inspector Repository** information. For example, **`openshift3/image-inspector`**.
3. Enter the **Image-Inspector Registry** information. For example, **`registry.access.redhat.com`**.
4. Enter the **Image-Inspector Tag** value. A tag is a mark used to differentiate images in a repository, typically by the application version stored in the image.
5. Enter **`https://www.redhat.com/security/data/metrics/ds/`** in **CVE location**.

10. Click **Add**.

Note:

You can also set global default image-inspector settings for all OpenShift providers in the advanced settings menu by editing the values under **`ems_kubernetes`**, instead of setting this for each provider.

For example:

```
:image_inspector_registry: registry.access.redhat.com
:image_inspector_repository: openshift3/image-inspector
```

Once you have completed the procedure, you will have OpenShift Prometheus alerts enabled in Infrastructure Automation, and can manage their life cyle from the **Monitor > Alerts** screen in the user interface.

Regular Expressions

In Infrastructure Automation, regular expressions can be used to search the contents of a file for a specific string for use in a condition. Below are listed the items most commonly used with Infrastructure Automation to search strings. These are a small subset of all the items available to use in regular expressions. If you are unfamiliar with regular expressions, there are many resources available on the Internet, including www.regular-expressions.info. Note that if you want to search a file, you must collect it as part of a host analysis profile.

Anchors		
^	start of string	
$	end of string	
Character Classes		
\s	white space including spaces, tabs, and line breaks	
\S	not white space	
\d	digit, same as [0-9]	
\D	not digit	
\w	word	
\W	not word	
Quantifiers		
*	0 or more of preceding characters	
+	1 or more of preceding characters	
?	0 or 1 of preceding character	
Escape Character		
\	put before a metacharacter to search for that actual character	
Metacharacters		
^ [. $ { * (\ +)	? < >	must be used with the Escape Character if you are searching specifically for it
Special characters		
\n	new line	

\t	tab	
Groups and Ranges		
.	any character except new line (\n)	
(a	b)	a or b
()	group	
[abc]	a or b or c	
[^abc]	not a or b or c	
[a-q]	letter between a and q	
[A-Q]	upper case letter between A and Q	
[0-7]	digit between 0 and 7	
Pattern modifiers		
i	case insensitive	
Other helpers		
.*	swallows text between 2 words	
\s+	guarantees minimum of 1 whitespace between 2 words	
\s*	guarantees 0 or more whitespace between 2 words	
^\s*	beginning of line with zero or more whitespace	
\s+.*	swallows all text and white space between 2 words	
\d+	guarantees minimum of 1 number between 2 words	
<\w>	identical to <[a-zA-Z0-(_]>	

Regular Expressions

Description	Regular Expression			
([a-zA-Z0-9]+)	Letters, numbers, hyphens			
Find the line beginning with **`sshd`**. Then, using a colon `:` as delimiter, check that the value four ":" over is equal to 99999.	^sshd:[^:]*:[^:]*:[^:]*:99999:			
Verify that **`PASS_MAX_DAYS`** exists starting in position `1` and a value after it is ⇐ 90.	^\s*PASS_MAX_DAYS\s+([0-9]	[1-8][0-9]	90)	
Verify that **`ROOTPW`** (in any case) exists on an uncommented line.	/[^#]*ROOTPW/i			
Verify that line in file starts with size and the value after is ⇐ 4096k.	^\s*size\s+(409[0-6]	40[0-8][0-9]	[123][09]{3}	\d{1,3})k
Find line with string **`restrict 127.0.0.1`** that starts in position `1 to ensure it is not commented out.	^\s*restrict\s+127\.0\.0\.1			
Find an uncommented line that contains **`/home`**. There will be additional text before the desired string.	[^#]*\/home			

Examples

Reportable Fields in Infrastructure Automation

Infrastructure Automation provides a set of out of the box reports covering a variety of areas. These fields are collected by Infrastructure Automation from the different parts of the infrastructure. Most of the fields are named directly as the vendor would name them. However, there are some fields that Infrastructure Automation calculates given this data. Note that this list may vary by Provider and may change as new Providers are added to Infrastructure Automation. Also, note that the value returned by each reportable field below is in megabytes (MB).

Reportable Field
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Avg Used for Collected Intervals 30 Day Avg

Reportable Field
VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg

Clusters

Reportable Field
Datastore : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Datastore : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Datastore : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
VM Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
VM Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
VM Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
VM Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VM and Instance : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VM and Instance : Memory - Avg Used for Collected Intervals 30 Day Avg
VM and Instance : Memory - Avg Used for Collected Intervals 30 Day High Avg
VM and Instance : Memory - Avg Used for Collected Intervals 30 Day Low Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Logical Disk.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
File Share.File System.Logical Disk.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
File Share.File System.Logical Disk.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
File Share.File System.Logical Disk.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
File Share.File System.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
File Share.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg

Datastore Files

Reportable Field
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.File System.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.File System.Logical Disk.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.File System.Logical Disk.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
File Share.File System.Logical Disk.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
File Share.File System.Logical Disk.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
File Share.File System.Logical Disk.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
File Share.File System.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg

Reportable Field
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
File Share.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
File Share.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
File Share.File System.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
File Share.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
File Share.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
File Share.Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
File Share.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
File Share.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
File Share.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
File Share.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Hosts : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg

Datastores

Reportable Field
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg

EVM Groups

Reportable Field
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Cluster : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Cluster : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Cluster : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Miq Templates : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max

Reportable Field
VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg

Hosts

Reportable Field
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
Cluster : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Cluster : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Cluster : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Datastore : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Datastore : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Datastore : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Datastores : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Host : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Host : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Host : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Provider : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Provisioned From Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Provisioned From Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Provisioned From Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Provisioned From Template : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
Provisioned VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Avg
Provisioned VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day High Avg
Provisioned VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Low Avg
Provisioned VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
Service.VMs : Memory - Aggregate Used for Child VMs for Collected Intervals 30 Day Max
Service.VMs : Memory - Avg Used for Collected Intervals 30 Day Avg
Service.VMs : Memory - Avg Used for Collected Intervals 30 Day High Avg
Service.VMs : Memory - Avg Used for Collected Intervals 30 Day Low Avg

Images

Chargeback

The Chargeback dashboard includes the utilization, reporting, and chargeback data from the Infrastructure Automation – Infrastructure Management component.

You can use the dashboard to view and manage chargeback cost. The chargeback feature provides the way to calculate monetary virtual machine charges based on owner or company tag. Access the Chargeback dashboard by browsing to the menu, click **Manage costs > Chargeback**

- Tag Categories and Tags
- Chargeback Rates
- Chargeback Assignments
- Chargeback Reports

Tags

With Infrastructure Management, you can create your own set of tags and tag categories. You can use tags to create a customized, searchable index for your resources. After you create these values, you can apply them to your resources.

Company Tag Categories and Tags

Infrastructure Management provides two types of tags. Company tags and System tags.

Company tags that you see under **My Company Tags** for a resource.

1. Create company tags from the console by browsing to the menu, click **Automate infrastructure > Infrastructure management**
2. Click **Settings**, then click **Application Settings**.
3. Select the region: **Region** and click the **Tags** tab.
4. Select the **My Company Tags** tab.

A selection of company tags is provided to you by default as samples. Tags can be deleted if you don't need them, but are not re-created by Infrastructure Management.

System tags are assigned automatically by Infrastructure Management.

For more information, see Tags.

Chargeback Rates

Infrastructure Management provides a default set of rates for calculating chargeback costs.

You can configure chargeback rates for various resources by selecting either Compute or Storage in the Rates tab. Compute sets chargeback rates for CPU, disk I/O, fixed compute cost, memory, and network I/O items. Storage sets chargeback rates for fixed storage costs and disk storage.

Chargeback costs are computed by using a set formula based on hourly cost per unit and hourly usage.

Chargeback can be calculated in many currencies.

Chargeback rates can be assigned at a single rate or by tiers, where rates are assigned in ranges that depend on the level of usage.

Creating Chargeback Rates

Infrastructure Management provides the capability to create your own set of computing and storage costs to use for billing.

Chargeback can be calculated at one fixed rate, or by a combination of fixed and variable rates per tier. Fixed rates are charged once per unit of time. The variable rate is calculated by the level of usage that is multiplied by the number of resources that are used in a unit of time.

Adding a Chargeback Rate

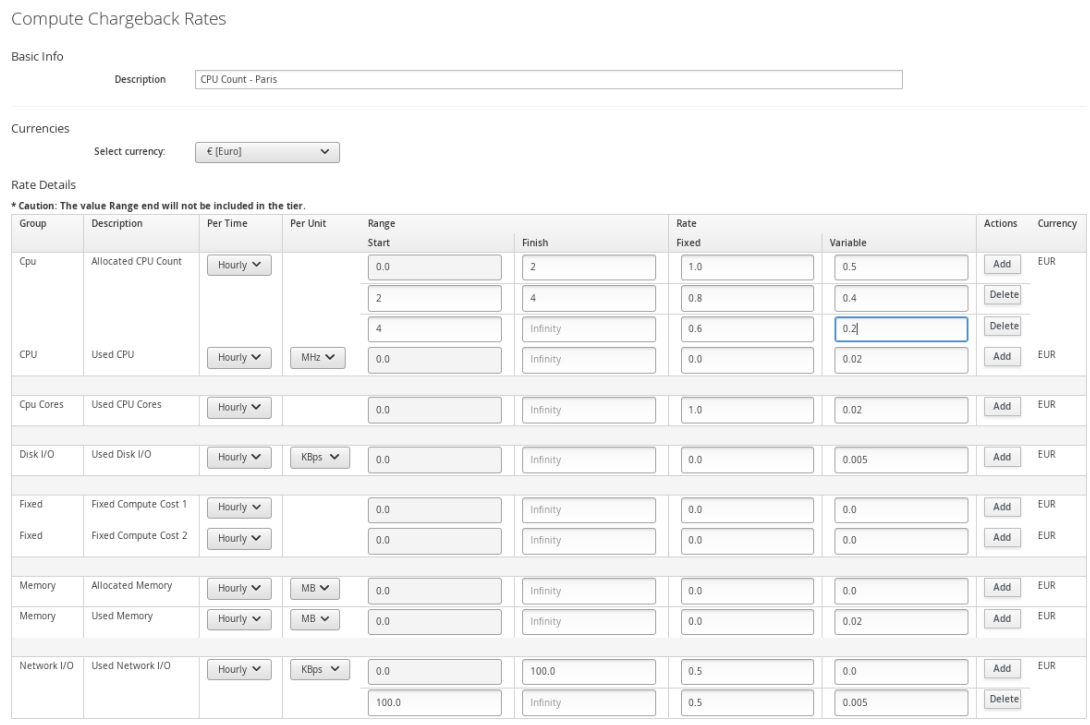
The following example shows a scenario to create a new chargeback rate for allocated CPU usage in Euro per week measured by the following tiers, calculating both fixed and variable costs:

- Tier 1: 0-2 CPU = 1 Euro fixed + 0.5 Euro variable
- Tier 2: 2-4 CPU = 0.8 Euro fixed + 0.4 Euro variable
- Tier 3: 4 CPU or more = 0.6 Euro fixed + 0.2 Euro variable

To configure this rate:

1. Browse to the menu, click **Manage costs > Chargeback**.
2. Click the **Rates** tab, then click **New rate**. The Infrastructure Management is launched in a new browser tab.

   Note: You can also navigate to Chargeback rates from the Infrastructure Management tab if it is already launched by selecting, **Overview** > **Chargeback** > **Rates**. Select **Configuration** > **Add a new Chargeback Rate**.
3. Enter a **Description** for the chargeback rate.
4. Confirm that the **Type** is set to **Compute**.
5. Select the currency for the new rate. For example, **Euro**.
6. In **Rate Details**, select the time, and complete the range and rates for each tier. Optionally, a single chargeback rate can be configured by using only one tier. For this example, in **Allocated CPU Count**, configure:
    1. For tier 1, set:
        1. In **Per Time**, select **Hourly**.
        2. In **Range**, specify the CPU range: **Start** remains as `0.0` and **Finish** is `2`.
        3. In **Rate**, specify `1` for the **Fixed** rate, and `0.5` for the **Variable** rate.
        4. Click **Add** under **Actions** to create a new, blank **Allocated CPU Count** line for tier 2.
    2. For tier 2, set:
        1. In the new line in **Range**, specify the CPU range: **Start** is `2` and **Finish** is `4`.
        2. In **Rate**, specify `0.8` for the **Fixed** rate, and `0.4` for the **Variable** rate.
        3. Click **Add** under **Actions** to create another new, blank **Allocated CPU Count** line for tier 3.
    3. For tier 3, set:
        1. In the new line in **Range**, specify the CPU range: **Start** is `4` and leave **Finish** as `Infinity`.
        2. In **Rate**, specify `0.6` for the **Fixed** rate, and `0.2` for **Variable**.



Tiered chargeback

7. Click **Add** to save the chargeback rate.

The rate can now be assigned.

For more information, see Chargeback Rates.

Chargeback assignments

Compute rates can be assigned to

- The enterprise
- Selected clusters
- Selected providers
- Tagged VMs and instances
- Tenants

Storage rates can be assigned to

- The enterprise
- Selected datastores
- Tagged datastores
- Tenants

Infrastructure Management provides you with the capability to assign chargeback rates by choosing from **Compute** and **Storage**. You can assign either a custom chargeback rate, or a default rate for Compute, Storage, or Container Images.

Note: The **Default Container Image Rate** can be useful when applied to a container provider with multiple images, where only some of the images have a chargeback rate assigned. The images without a custom rate use the default rate.

1. Browse to the menu, click **Manage costs > Chargeback**.
2. Click the **Assignments** tab, then click **Create assignments**. The Infrastructure Management component is called.
3. From the **Chargeback Assignments** menu, click **Compute** or **Storage**.

Configuring Chargeback for Compute

Use **Compute** to assign a compute chargeback rate. You can assign chargeback rates to **The Enterprise**, **Selected Providers**, **Selected Clusters**, **Tagged VMs and Instances**, **Tagged Configured Systems**, **Tagged Images**, **Labeled Images**, or **Tenants**.

1. From the **Selections** area, choose each item to set, select the chargeback rate to use. The options that are displayed vary based on the type you select.
2. For **Tagged VMs and Instances** and **Tagged Images** select a **Tag Category**.
3. Click **Save**.

Configuring Chargeback for Storage

Use **Storage** to assign a storage chargeback rate. You can assign chargeback rates to **The Enterprise**, **Selected Datastores**, **Tagged Datastores**, or **Tenants**.

1. From the **Selections** area, choose each item to set, select the chargeback rate to use. The options that are displayed vary based on the type you select.
2. For **Tagged Datastores**, select a **Tag Category**.
3. Click **Save**.

The rate is assigned. The next time that you generate a chargeback report, these values will be used.

Note: When you view chargeback, you see a rate for a virtual machine for the number of the CPUs. The chargeback for this parameter is calculated based on when the virtual machine is running. If the virtual machine isn't running, then it is not charged for CPU allocation.

For more information, see Assigning Chargeback Rates.

Chargeback Reports

Prerequisites for running chargeback reports

1. Enable Capacity & Utilization Collection and Metric Workers
    1. Browse to the menu, click **Automate infrastructure > Infrastructure management**.
    2. Click **Settings**, then click **Application Settings**.
    3. Click **Settings > Current Region**, then click the **C & U Collection** tab.
    4. Metrics data is needed to generate chargeback reports. It is required to enable collecting metrics data. You can select individual cluster(s), datastore(s) or select all clusters and datastores.
    5. Click **Save**.
2. Set Server Roles
    1. Browse to the menu, click **Automate infrastructure > Infrastructure management**.
    2. Click **Settings**, then click **Application Settings**.
    3. Click **Settings > Current Region**, then in the tree select **Zones > Current Zone > Current Server**.
    4. In the **Server Control** section turn on the required server roles to get metric data for chargeback reports:
        - Capacity & Utilization Coordinator
        - Capacity & Utilization Data Collector
        - Capacity & Utilization Data Processor
3. Set Assignments
    1. See Chargeback Assignments
    2. Assign rates to a specific scope of resources (The Enterprise, Selected Providers,...). All resources (VMs,...) have an assigned rate when you select **The Enterprise** (for Compute and Storage) in the **Assigned To** selection box.
4. Set Chargeback Filters on chargeback reports

    Filters on chargeback reports must match resources to see any results on the report result. Setting of the filters affects the scope of resources taken into account for chargeback report generation. You can display all resources (VMs,...) on the report result by setting **Show Costs by** to **tenant** and **Tenant** to **My Company** in the **Chargeback Filter** section. When creating reports, click the **Filter** tab to set filters for the data displayed in the report.

Chargeback Reports can be generated based on:

- Allocation
- Utilization

Creating a Chargeback Report

Infrastructure Management provides you with the capability to create chargeback reports to monitor costs.

Note: When you create a chargeback report, the **Total Cost** is the complete cost for the chosen workload over the selected time frame. Total costs are calculated by adding the calculated costs for the amount that is used and the amount that is allocated for the time frame. When you add cost-related columns to display on your report, be sure to select all rate assigned fields to present a comprehensive accounting of total costs.

1. Browse to the menu, click **Manage costs > Chargeback**.
2. Select the **Reports** tab, then click **Create reports** or **New report**.

    The Infrastructure Management is launched in a new browser tab.
3. From the Infrastructure Management, select the **Overview** > **Reports**.
4. Click **Reports** accordion, then select **Configuration** and click **Add a new Report**.
5. On the **Columns** tab, complete the **Basic Report Info** area.
    - Type a unique name in **Menu Name** for how you want the report to be named in the menu list.
    - Type the **Title** to display on the report.
6. Add fields in the **Configure Report Columns** area.
    - From the **Base the report on** list, select **Chargeback for Vms**, **Chargeback for Configured Systems**, **Containers**, or any other object.

- Select the fields to include in the report from the **Available Fields** list, then click **Move selected fields down**. Also, you can select any tags that you created and assigned.
- Change the order of the fields in the report by clicking **Move up selected fields** or **Move down selected fields**.

7. Click the **Formatting** tab to set the size of paper for a PDF and column header format.
   - From the **PDF Output** area, select the page size from the **Page Size** list.
   - From **Specify Column Headers and Formats**, type the text to display for each field. For each numeric field, you can also set the numeric format.
8. Click the **Filter** tab to set filters for the data displayed in the report.
   - From **Chargeback Filters**, select how you want the costs to show, the tag category, the tags, and how you want the items to be grouped. **Note:** You can select multiple tags.
   - From **Chargeback Interval**, select the time interval. You must have a full interval worth of data to select an option other than **Partial** in the **Daily Ending With** list.
9. Click the **Preview** tab, and then **Load** to see how the report appears.
10. When you're satisfied with the report, click **Add** to create the new report.

The new report is created. To make the report accessible from the **Report** menu, you must add it to a report menu.

For more information, see Creating a Chargeback Report.

Running Reports

Generating a Single Report

1. Browse to the menu, click **Manage costs > Chargeback**.
2. Select the **Reports** tab, and then click the report name in the **Reports** section.
3. Click **Run report** to generate the report.

Infrastructure Management uses interactive report generation so that reports are placed on a queue. A visual indicator of the reports status is shown. All reports are automatically saved so that they can be downloaded and analyzed later.

Viewing Reports

Once you have run a report, you can view it at any time after the report has been generated.

1. Browse to the menu, click **Manage costs > Chargeback**.
2. Select the **Reports** tab, and then click the report name in the **Reports** section.
3. From the **Saved Reports**, click the report **Last run** entry under the **Saved Reports** to view the report details.

Note: From the report details you can click the column headings to sort the details by value.

Exporting Reports

1. Browse to the menu, click **Manage costs > Chargeback**.
2. Select the **Reports** tab, and then click the report name in the **Reports** section.
3. From the **Saved Reports**, click the report **Last run** entry under the **Saved Reports** to view the report details.
4. Click **Export Report** and select **Download as Text** or **Download as CSV**.

Managing integrations for Infrastructure Management

Learn more about the available integrations for Infrastructure Automation - Infrastructure Management.

To learn more about managing integrations for Infrastructure Management, click a subject within the following image to view the related available documentation:

- ServiceNow integration
- Change risk integration

- Overview
- Configuring ServiceNow Connection
- Managing Record Attributes
  - Create and Update (_patch) Record
    - Attribute
  - Get and Delete Record
  - Get All Records
- Use Cases

- Provisioning a Virtual Machine from a Template
- Virtual Machine Retirement
- Virtual Machine Reconfiguration (VMware Only)

Overview

Infrastructure Automation integration with ServiceNow enables authentication with an existing ServiceNow database and **add/amend** items in the ServiceNow database during state machine processing, such as the virtual machine provisioning state machine. Note that information in this guide assumes you have credentials and access to a ServiceNow database instance.

The following new namespace and class delivers support for the management of **ServiceNow Configuration Management Database** (CMDB) records using **ServiceNow's RESTful** web service.

`/Integration/ServiceNow/CMDB`

You can manage records in the **CMDB_CI_SERVER** table, including `create`, `update`, and/or `delete`. The following methods are included:

Method	Action
create	Create record in specified ServiceNow table.
delete	Delete record in specified ServiceNow table.
get	Get record from specified ServiceNow table and list its attributes.
get_all	Get all records in ServiceNow and list attributes.
update	Get specified record, update required attributes and post updated record.
update_patch	Post required attributes to specified record.

Note:

Configuration item (CI) and record are used interchangeably and refer to items in a ServiceNow database table.

Configuring ServiceNow Connection

Configure the connection to the ServiceNow database by specifying the credentials in the **CMDB schema** or instances within.

The following methods are included:

snow_server	ServiceNow database IP address or resolvable hostname.
snow_user	ServiceNow user account with the necessary permissions.
snow_password	Associated user account password.

The table name cannot be changed unless there is a specific requirement to manage records elsewhere. Entries in this table appear in the **Configuration > Base Items > Servers** menu in the ServiceNow web user interface.

table_name	ServiceNow cmdb_ci_server database table.

Managing Record Attributes

You can specify any attribute via the **URI** or **CMDB** class instance. If neither exists, the value is determined from the Infrastructure Automation **VM** or **miq_provision** objects.

The attributes can be reduced or extended as required by amending the Ruby methods.

Note:

Some attributes are not free-text (variable) fields, that is, they must be specific values. For example, the `vendor` value must already exist in the ServiceNow Vendor table beforehand.

Create and Update (_patch) Record

You can use the following attributes when creating and/or updating a record.

Attribute

Attribute	Value
virtual	Boolean true or false, set to true.
name VMs	Virtual infrastructure name.
short_description	Infrastructure Automation virtual machine GUID.
host_name	The virtual machine's operating system hostname.
cpu_count	The virtual machine's CPU count.
ram	The virtual machine's memory.
vendor	The virtual machine's hardware vendor (provider).
sys_id [1]	ServiceNow record unique system ID.

Get and Delete Record

You can use the following attribute for getting or deleting a record.

Attribute	Value
sys_id	ServiceNow record unique system ID.

Get All Records

There are no attributes required to get all records. This method gets all records in the specified **ServiceNow table** and writes their attributes to `automation.log`.

1. sys_id attribute is not required during `create`. Its value is returned from the `create` request and the Infrastructure Automation object custom attribute **servicenow_sys_id** is created and updated.

Use Cases

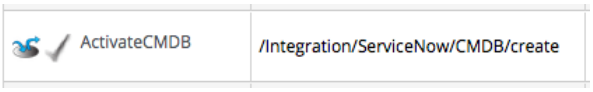
The following examples show how ServiceNow can be integrated with automation workflows.

Provisioning a Virtual Machine from a Template

The Cloud and Infrastructure **Provision VM from Template** State Machines contain **RegisterCMDB** and **ActivateCMDB** states.

To create a new ServiceNow record during virtual machine provisioning, amend the **ActiveCMDB** state to call the `create` method, for example:

`/Integration/ServiceNow/CMDB/create`

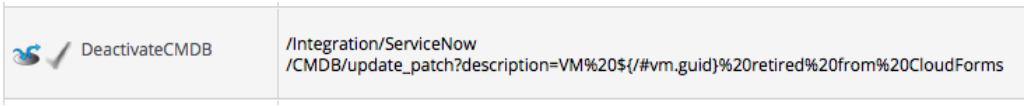


Virtual Machine Retirement

The Cloud and Infrastructure default **Retirement** State Machines contain the **DeactivateCMDB** state.

To update a virtual machine's ServiceNow record during virtual machine retirement, amend the **DeactivateCMDB** state to call the `update_patch` method, for example:

`Integration/ServiceNow/CMDB/update_patch?description=VM%20${/#vm.guid}%20retired%20from%20{productname_short}`



Virtual Machine Reconfiguration (VMware Only)

Create a new **System Event** instance to update the ServiceNow record after a virtual machine reconfiguration request has been approved and completed.

Create a new **/System/Event/ReconfigVM_Task_Complete** instance with a relationship value:

`/Integration/ServiceNow/CMDB/update_patch`



Change risk integration for Infrastructure Automation

You can enable an integration for Infrastructure Automation to gain the advantage of change risk assessments. With this integration, your IT operators can use data from ServiceNow to assess the risk of a code deployment based on prior incidents, estimate the chances of a successful update, identify risks, build logs, apply security policies and more.

The Change Risk module ingests un-structured data from ServiceNow, and analyzes the data to look for similar tickets and to understand how incidents and failures might have occurred. The module can then provide an assessment of the overall risk of suggested changes. With an integration with ServiceNow for change risk, Infrastructure Automation allows for the automation of the delivery of infrastructure services which integrates with ServiceNow. Since Infrastructure Automation is an end-to-end infrastructure provisioning and management capability, Infrastructure Automation can complete ongoing checks against configurations after infrastructure is provisioned to ensure systems are running as prescribed and adhere to system-wide compliance policies. Through the integration, updates can be made to the ServiceNow CMDB or to create change requests.

With the integration in place, the following flow demonstrates the possible interactions between Infrastructure Automation and the Change Risk components of IBM Cloud Pak for AIOps and ServiceNow:

Scenario: A change is proposed to add additional disk space to a virtual machine.

Interaction Flow:

1. The Operator invokes a day-2 operation in Infrastructure Automation to change the server configuration to create and attach the new disk.

2. The day-2 operation creates a Change Request ticket in ServiceNow, populating it with details on the requested change (virtual machine ID, new disk size, and more).
3. The Change Risk component monitors the ServiceNow tickets. Using a pre-trained model to analyzing similar tickets, IBM Cloud Pak for AIOps can classify the change as a high or low risk change and add a confidence level of the prediction onto the ServiceNow ticket.
4. The ServiceNow operator approves the change request based on the confidence level, and marks it ready for implementation.
5. The day-2 operation in Infrastructure Automation waits for the ServiceNow Change Request to be approved before proceeding.
6. After observing the change in the state of the ServiceNow Change Request, Infrastructure Automation carries out the provisioning change to add the disk to the virtual machine.
7. After completing the provisioning action, Infrastructure Automation updates the ServiceNow Change Request to mark it as complete. The request can be reviewed and closed by a ServiceNow operator.
8. Infrastructure Automation policies are applied against the configuration and alerts are raised if server configuration isn't in compliance. Both server VM parameters as well as application middleware packages can be checked for compliance against defined policies.

For more information abouth the steps to configure the AI algorithm to determine change risk, see Configuring change risk.

Creating IBM Infrastructure Management integrations

You can send event data, inventory data, and metrics data from Infrastructure Management to IBM Cloud Pak for AIOps through an Infrastructure Management integration.

- Before you begin
- Creating Infrastructure Management integrations
- Editing Infrastructure Management integrations
- Deleting Infrastructure Management integrations

Before you begin

Before you configure an Infrastructure Management integration, ensure that the following prerequisites are met:

- You have a working Infrastructure Management cluster. Infrastructure Management must be installed on an Red Hat OpenShift Container Platform cluster. The integration cannot be created for an Infrastructure Management appliance that is running on a virtual machine (VM).
- Event syndication, inventory syndication, and metrics syndication must be enabled within Infrastructure Management as required.

  For information on how to enable event, inventory, and metrics syndication, see *Enabling events, inventory and metrics syndication* within the topic General Configuration.

Note: Infrastructure Management and Cloud Pak for AIOps can be located in the same cluster or in different clusters.

Creating Infrastructure Management integrations

To create an Infrastructure Management integration, complete the following steps:

1. Log in to the IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the **Integrations** page, click **Add integration**.
4. From the list of available integrations, find and click the Infrastructure Management tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Select the **Events**, **Topology**, and **Metrics** from the **Category** list.
5. On the side panel, review the information and when ready to continue, click **Get started**.

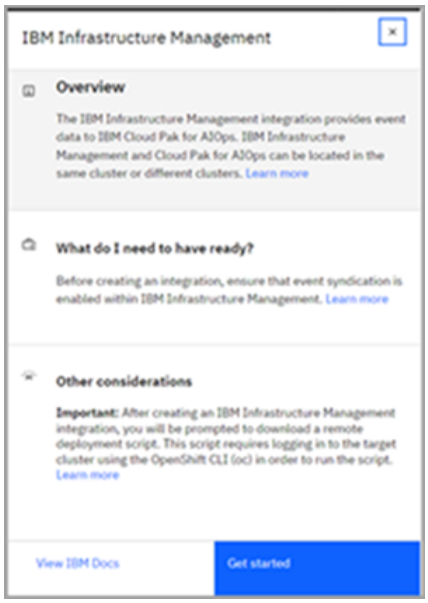


Figure. Add Infrastructure Management integration - get started

6. On the Add integration page, define the general integration details:

   Name: The display name for the integration.

   Description (optional): An optional description for the integration.

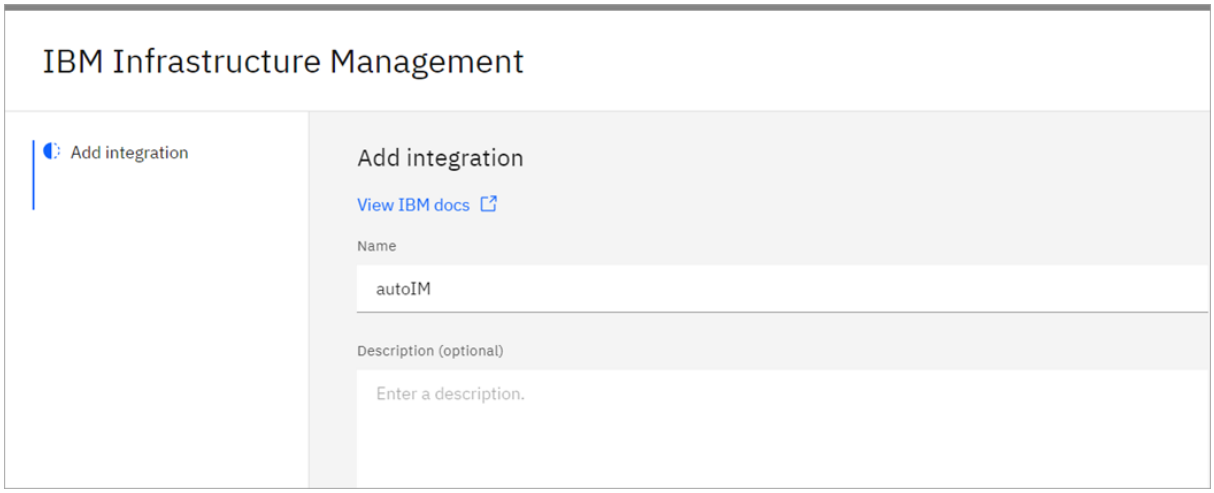


Figure. Add Infrastructure Management integration

7. Click **Done**.

   Important: After creating the integration, you are prompted to copy or download a boostrap command for running a remote deployment script. The script is generated when you create the integration in IBM Cloud Pak for AIOps. This script is required for establish the integration from Infrastructure Management to IBM Cloud Pak for AIOps.

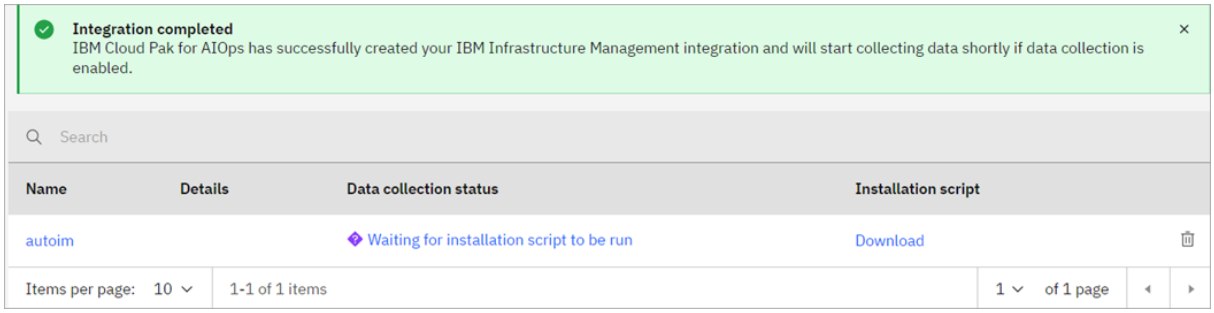


Figure. Add Infrastructure Management integration - completed

8. Copy or download the bootstrap command for the deployment script.

   You can click the **Copy to clipboard** button to copy the command, or click **Download as shell script** to download the command as a `bootstrap.sh` file.

   Note: If you do not obtain the command now, you can copy or download the command later by completing the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Infrastructure Management** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Infrastructure Management page, click the **Download** link in the **Remote deployment script** column for the integration.
5. Either copy or download the bootstrap command for the deployment script.

9. Run the remote deployment script on your Infrastructure Management cluster.
    1. Get the OpenShift CLI **`oc login`** command from the cluster where Infrastructure Management is installed.
    2. From a command line, log in to the Infrastructure Management cluster with the **`oc login`** command.
    3. Switch to the project (namespace) where Infrastructure Management is installed.
    4. Run the deployment script as a **`sh`** (script file), such as the downloadable **`bootstap.sh`** file.

Editing Infrastructure Management integrations

After you create a integration, you can edit the integration to update the name or description. Complete the following steps to edit the integration:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Infrastructure Management** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Infrastructure Management page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required.
6. Click **Save** when you are done editing.

Deleting Infrastructure Management integrations

If you no longer need your IBM Infrastructure Management integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete a integration, you need to complete the following steps for IBM Cloud Pak for AIOps. Then, you need to complete steps on your Infrastructure Management to remove the integration deployment. The deployment on the cluster is not removed when you delete the integration in the IBM Cloud Pak for AIOps console for IBM Cloud Pak for AIOps.

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Infrastructure Management** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Infrastructure Management integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**. Your integration is deleted.
6. Delete the integration deployment on your Infrastructure Management cluster. To delete the deployment, you must manually run OpenShift CLI (**`oc`**) commands against the cluster.
    1. From a command line, use the **`oc login`** command to log in to your Infrastructure Management cluster.
    2. Switch to the project (namespace) where Infrastructure Management is installed.

       ```
       oc project <project>
       ```

       Where **`<project>`** is the project (namespace) where Infrastructure Management is installed.
    3. Delete the **`grpc-bridge-secret`** secret.

       ```
       oc delete secret grpc-bridge-secret
       ```
    4. Delete the integration deployment

       ```
       oc delete deploy aiops-connector
       ```

       The deployment is now deleted from your Infrastructure Management cluster.
    5. Delete the **`aiops-connector-allow-kafka`** network policy:

       ```
       oc delete networkpolicy aiops-connector-allow-kafka
       ```


- Assigning a Custom Analysis Profile to a Virtual Machine
    - Creating a Virtual Machine Analysis Profile

- Creating an Action to Assign the Virtual Machine Analysis Profile to the Analysis Task
- Creating a Virtual Machine Control Policy
- Creating a Policy Profile and Assigning the Virtual Machine Control Policy
- Assigning the Policy Profile to a Virtual Machine

Assigning a Custom Analysis Profile to a Virtual Machine

You can create and assign different analysis profiles for specific virtual machine configurations using the Infrastructure Automation user interface. This document describes the steps required to create a custom virtual machine analysis profile, and assigning it to a virtual machine for use with SmartState analysis, via a control policy.

1. Creating a Virtual Machine Analysis Profile.
2. Creating an Action to Assign the Virtual Machine Analysis Profile to Analysis Task
3. Creating a Virtual Machine Control Policy.
4. Creating a Policy Profile and Assigning the Virtual Machine Control Policy.
5. Assigning the Policy Profile to a Virtual Machine.

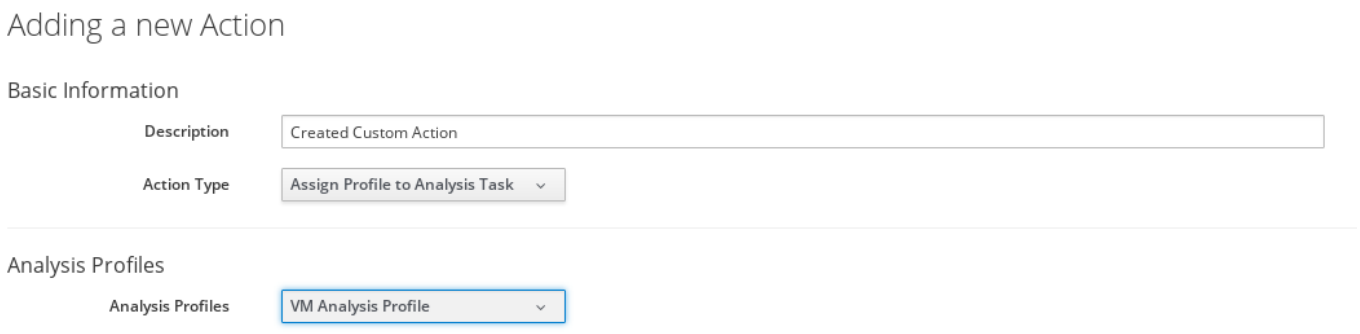
Creating a Virtual Machine Analysis Profile

1. Log into the appliance as the **admin** user.
2. Click **Configuration**.
3. Expand the **Settings** accordion, then click **Analysis Profiles**.
4. Click (**Configuration**), then (**Add VM Analysis Profile**).
5. In **Basic Information**, enter **Name** and **Description** for the analysis profile.
6. In the **Category** tab, select the categories you want to collect information for under **Category Selection**. The **Category** tab is available for virtual machine profiles only.
7. In the **File** tab, click **<New Entry>** and specify the file to be scanned and if you want to collect contents for, then click **Save**. Repeat the step for adding multiple file name entries.
8. In the **Registry** tab, enter the **Registry Key** and **Registry Value**, then click **Save**. To determine whether a registry key exists without providing a value, enter ` ` in *Registry Value</emphasis>. The **Registry** tab is available for virtual machine profiles only.
9. In the **Event Log** tab, specify the event log entries to collect. Enter a **Filter Message** to look for specific text in a message. Enter a **Level** (*info*, *warn* or *error*, for example) to specify the event level. Enter the **Source** for the event log entry and **# (Number) of Days** to specify how far back to check.
10. Click **Add**.

Creating an Action to Assign the Virtual Machine Analysis Profile to the Analysis Task

Actions are performed after the condition is evaluated. You can associate actions with specific events when you create a policy. Infrastructure Automation provides a set of default actions, but you can also create custom actions using the Infrastructure Automation user interface.

Use this procedure to create a custom action by adding the **Assign Profile to Analysis Task** action type to the virtual machine analysis profile (created in Creating a Virtual Machine Analysis Profile).



1. Browse to menu: **Control > Explorer**.
2. Expand the **Actions** accordion and click **Configuration**, then (**Add a new Action**).

3. Enter a **Description** for the new action. This will be the name given to your new action.
4. Select **Assign Profile to Analysis Task** from the **Action Type** list.
5. Select the newly-created virtual machine analysis profile from the **Analysis Profiles** list.
6. Click **Add**.

Note:

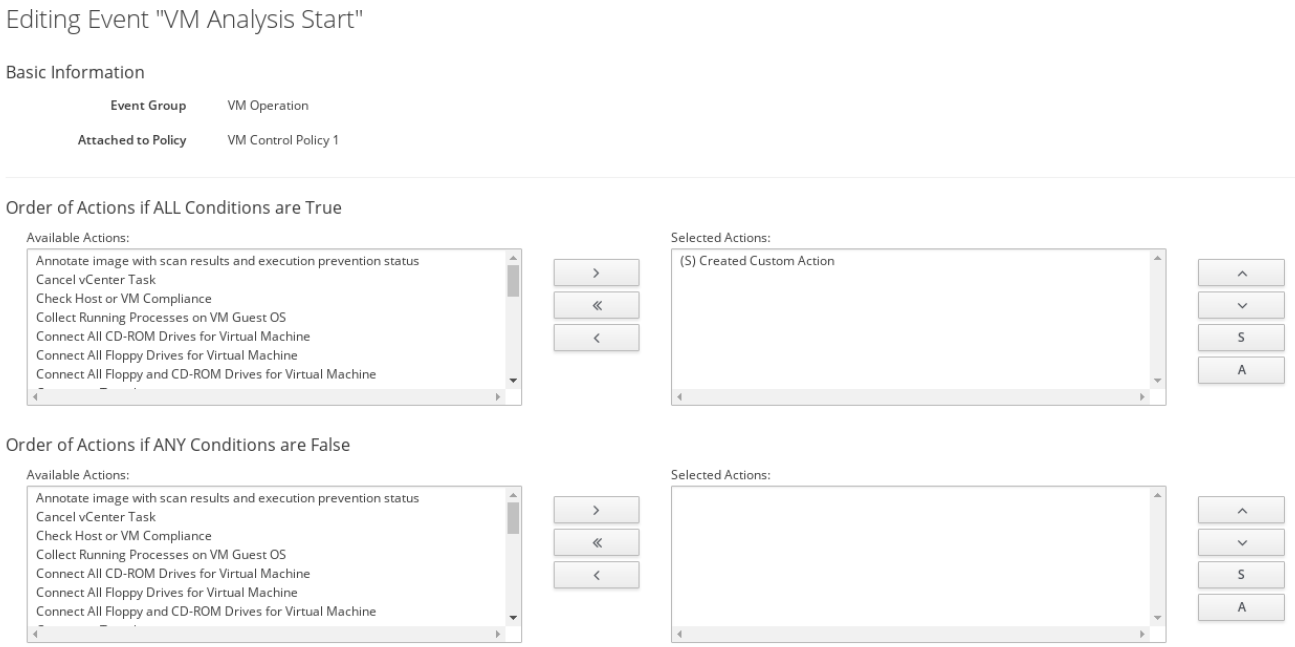
You can only associate this action with an analysis start event.

The action is created and added to the **Available Actions** list. Associate this action with the **VM Analysis Start** event when you create a virtual machine control policy in the next procedure.

Creating a Virtual Machine Control Policy

You can create a control policy by combining an event, a condition, and an action. The procedure below describes how to create a virtual machine control policy to assign the newly-created action to the **VM Analysis Start** event. Optionally, you can use a scope expression that is tested immediately when the policy is triggered by an event. If the item is out of scope, then the policy will not continue on to the conditions, and the assigned action will not run.

1. Browse to menu: **Control > Explorer**.
2. Expand the **Policies** accordion, and click **Control Policies**.
3. Select **Vm Control Policies**.
4. Click **Configuration**, then **Add a New VM and Instance Control Policy**.
5. Enter a **Description**. This will be the name given to your VM control policy.
6. Clear the **Active** box if you do not want this policy processed even when assigned to a resource.
7. Optional: Enter a **Scope** (you can also create a scope as part of a condition, or not use one at all). If the virtual machine is not included in the scope, the assigned action will not run.
   - You can use the drop-down list to create an expression for the **Scope**. Based on what you choose, different options appear. Click **Commit expression element changes** to add the scope.
8. Enter **Notes** if required.
9. Click **Add**. The policy is added and listed under **Vm Control Policies** in the **Policies** accordion.
10. Select the newly-added VM control policy. You can now associate events, conditions, and actions with the policy.
11. Click **Configuration**, then **Edit this Policy's Event assignments**.
12. Under **VM Operation**, set **VM Analysis Start** to **Yes**.
13. Click **Save**.
14. Click the **VM Analysis Start** event to configure actions.
15. Click **Configuration**, then **Edit Actions for this Policy Event**.
16. In **Order of Actions if ALL Conditions are True**, select the action created in Creating an Action to Assign the Virtual Machine Analysis Profile to the Analysis Task from the **Available Actions** list. This action will take place if the resources meet the conditions of the policy.



Note:

Each selected action can be executed synchronously or asynchronously; a synchronous action will not start until the previous synchronous action is completed, while an asynchronous action allows the next action to start whether or not the first action has completed. Also, at least one Infrastructure Automation server in the Infrastructure Automation zone must have the notifier server role enabled for the trap to be sent.

17. Click ( ) which will move the action to **Selected Actions**. The selected action is set to (S) Synchronous by default. From **Selected Actions**, select the action, then:
    - Click **A** (Set selected Actions to Asynchronous) to make it asynchronous.
    - Click **S** (Set selected Actions to Synchronous) to make it synchronous. If creating a synchronous action, use the up and down arrows to identify in what order you want the actions to run.
18. Click **Save**.

Creating a Policy Profile and Assigning the Virtual Machine Control Policy

Add a new policy profile and assign the virtual machine control policy (created in Creating a Virtual Machine Control Policy) to it.

1. Browse to menu: **Control > Explorer**.
2. Expand the **Policy Profiles** accordion, click **Configuration**, then **Add a New Policy Profile**.
3. In the **Basic Information** area, enter a **Description** for the policy profile.
4. Under **Policy Selection**, select the virtual machine control policy created in Creating a Virtual Machine Control Policy from the **Available Policies** list.
5. Click ( ) to move the selected virtual machine control policy into this profile.
6. Click **Add**.

Assigning the Policy Profile to a Virtual Machine

Assign the policy profile you created in Creating a Policy Profile and Assigning the Virtual Machine Control Policy to a virtual machine, and initiate SmartState analysis.

Note:

Policy profiles can be specified at multiple levels. If you assign a policy profile to a provider (Amazon EC2 or OpenStack for example), the profile will apply to all hosts or virtual machines for that provider.

1. Browse to menu: **Compute > Clouds > Instances** or menu: **Compute > Infrastructure > Virtual Machines**. Select the virtual machine or instance.
2. Click **Policy**, then **Manage Policies**.
3. Under **Select Policy Profiles**, select the policy profile created in Creating a Policy Profile and Assigning the Virtual Machine Control Policy. It will turn blue to show the selection. Click the triangle next to the policy profile to see its member policies.

4. Click **Save**.
5. Click **Configuration**, then **Perform SmartState Analysis**. A pop-up window appears to confirm the action.
6. Click **OK**. SmartState analysis is initiated for the selected virtual machine or instance from the Infrastructure Automation database.

SmartState analysis will now report back findings specified by the custom virtual machine analysis profile.

- Scripting Actions in Infrastructure Automation
  - Updates to Rails and Ruby
    - Updates and Changes in Rails
    - Updates and Changes in Ruby
  - Understanding the Automate Model
    - Automate Model
    - Creating a Domain
    - Editing a Domain
    - Deleting a Domain
    - Importing a Domain from a Git Repository
    - Changing Priority Order of Domains
    - Creating a Namespace
    - Creating a Class
    - Creating a Schema for a Class
    - Editing a Field in a Schema
    - Editing Schema Sequence
    - Adding an Instance to a Class
    - Copying a Class or Instance
    - Relationships
    - Methods
      - Creating a Method
      - Creating a Dynamic Content Dialog
      - Creating a Playbook Automate Method
        - Passing variables between successive playbook methods
    - Expression Methods
      - Input Parameters
        - Optional Input Parameters
      - Creating an Expression Method
    - Simulation
      - Simulating an Automate Model
    - Importing, Exporting, and Resetting the Datastore
      - Exporting All Datastore Classes
      - Importing Datastore Classes
      - Resetting Datastores to the Default
  - Invoking Automate
    - Automate Examples
    - Invoking Automate using a Custom Button
    - Creating a Custom Button Group
    - Creating a Custom Button
    - Creating an Ansible Playbook Button
    - Editing a Custom Button
    - Deleting a Custom Button
    - Setting Enablement and Visibility for Custom Buttons
    - Using a Custom Button
    - Initiating Automate from an Event
    - Creating a Policy for Automate
    - Creating a Custom Automate Action
- Objects
  - Virtual Machine Properties
  - Methods for use in Ruby Scripts
  - Host Properties
  - Provider Properties
  - Storage Properties
- FAQs and Flows
  - Phase 1: Create Provision Request
  - Phase 2: Request Approval
  - Phase 3: Quota Validation
  - Phase 4: Provisioning
  - Phase 5: Retirement
- Inline Method to Create a Provision Request
  - Ruby method
- Migrating Custom Buttons
  - Migrating Custom Buttons

Scripting Actions in Infrastructure Automation

The Automate model provides flexibility to not only change parts of the provisioning process, but also to allow you to automate other operational tasks. Below are some scenarios where the Automate model can help accomplish these tasks.

- Intelligent Workload Management - An enterprise had a requirement that when a virtual machine has reached a High CPU Percent Ready for a specified period of time, a vMotion should occur to a more suitable host. For this reason, VMware's Distributed Resource Scheduler (DRS) was not practical, as the CPU Ready metric could not trigger DRS. The solution was to leverage Infrastructure Automation Control and Infrastructure Automation Automate to drive the management of this workflow.
- Power on only during business hours - An organization which gave a group of self-service users Infrastructure Automation access had a requirement to only allow certain virtual machines to be powered during business hours. This was solved with Infrastructure Automation Automate.
- Auto-Tagging virtual machines based on file contents - An IT organization needed a way to consume information from a text file on a virtual machine and dynamically populate vCenter. The data used to auto-tag virtual machines is also used to align unmanaged virtual machines to the business.

Updates to Rails and Ruby

Changes within Ruby and Rails versions can create issues for custom code and automation in Infrastructure Automation environments.

Before proceeding with major upgrades that include changes to versions of Ruby and Rails, review the below resources.

Updates and Changes in Rails

See Rails Releases for information on each release of Rails. To compare versions of Rails, see Comparing changes in Rails.

Updates and Changes in Ruby

See Ruby Releases for information on each release of Ruby by version number. To compare versions of Ruby, see Comparing changes in Ruby.

Understanding the Automate Model

Automate enables real-time, bi-directional process integration. This provides users with a method to implement adaptive automation for management events and administrative or operational activities.

Automate Model

The Automate model is arranged to provide an object oriented hierarchy to control automation functions. The model uses the following organizational units arranged in a hierarchy:

- **Datastore** - The main organization unit that stores the entire model.
- **Domains** - Domains act as collection of automation functions. Functions are executed depending on the order of Domain priority, which means a function in a Domain with a higher priority overrides the same functions specified in a lower-priority Domain. This allows Infrastructure Automation to specify a core Domain (ManageIQ) but allow users to override automate functions with custom Domains. Each Domain contains a set of Namespaces.
- **Namespaces** - Containers that organize and categorize functions of the model. Namespaces can contain child Namespaces as well as Classes.
- **Classes** - Templates for a specific function of the model. Each Class uses a Schema to apply to Instances to populate with default values. Each class also can contain a set of methods.
- **Instances** - An instance is a version of a class populated with initial configuration data. An instance can include a collection of any number of attributes, calls to methods, and relationships.
- **Methods** - Methods are functions within the model. Methods use Ruby code to execute various operations needed for a Class.

Infrastructure Automation contains a preconfigured Domain for users:

- **ManageIQ** - The core domain for Infrastructure Automation Automate operations. This domain is locked with the following Namespaces:
  - **Cloud** - General cloud instance lifecycle from provisioning, retirement, methods, email.
  - **Control** - Control contains email alerts for policy controls.
  - **Infrastructure** - General infrastructure VM lifecycle from provisioning, retirement, methods, email.
  - **Integration** - Used to interface with systems outside of Infrastructure Automation. Use this namespace to integrate with additional systems.
  - **Service** - Service lifecycle from provisioning, retirement, methods, email.
  - **System** - System contains classes that can provide the start points for all Infrastructure Automation Automate activities.

You can copy classes and instances from locked Domains into your own custom domains.

Note:

Changing the existing classes or instances shipped with the product is not recommended because this may hinder the operation of Infrastructure Automation. You can link to these methods using relationships.

To reset the Automate model to default settings, navigate to **Automate > Import/Export** and click the **Reset** option.

Creating a Domain

1. Navigate to **Automation > Automate > Explorer**. The default view is the Datastore.
2. Click (**Configuration**), then (**Add a New Domain**).
3. Type in a unique **Name** and **Description**. Choose if the Domain is **Enabled**.
4. Click **Add**.

The new domain is created.

Editing a Domain

1. Navigate to **Automation > Automate > Explorer**. The default view is the Datastore.
2. Select the Domain you want to edit.
3. Click (**Configuration**), then (**Edit Selected Domain**).
4. Make the required edits.
5. Click **Save**.

You have edited the selected domain.

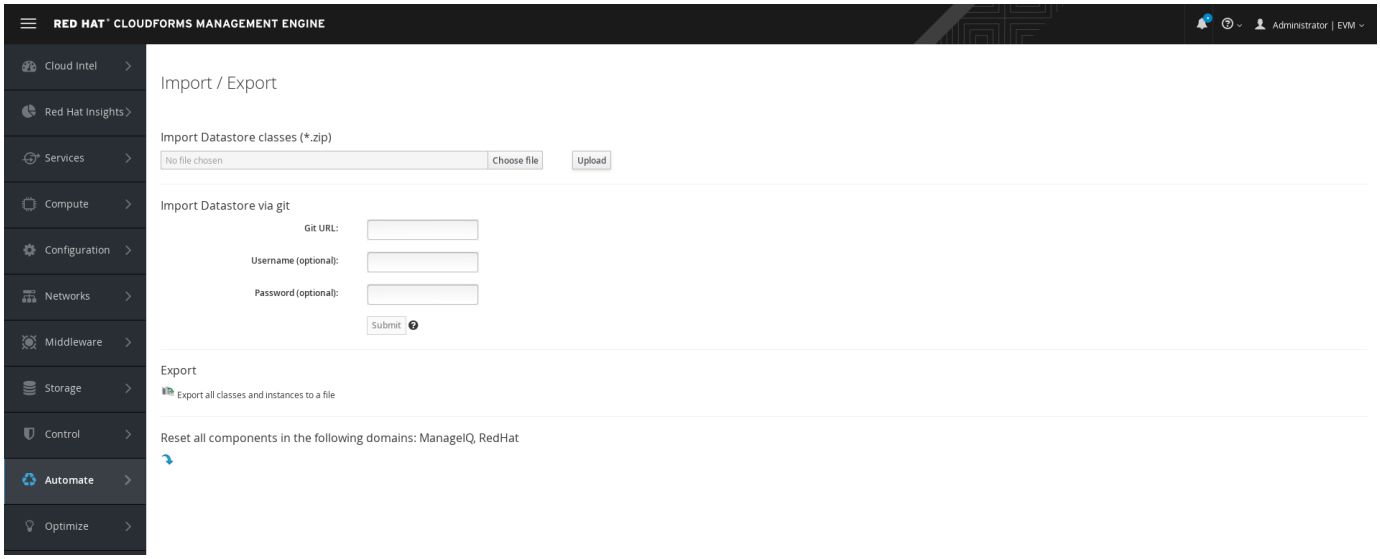
Deleting a Domain

1. Navigate to **Automation > Automate > Explorer**. The default view is the Datastore.
2. Select the **Domain** that you want to delete.
3. Click (**Configuration**), then (**Remove This Domain**).
4. A window to confirm the removal of Domain appears.
5. Click **OK**.

The selected Domain is deleted.

Importing a Domain from a Git Repository

Infrastructure Automation can import an Automate domain from a Git repository by specifying a repository and branch, along with user details. Currently, you can only add git domains via the **Import/Export** area of the user interface.



1. Navigate to **Automation > Automate > Import/Export**.
2. In **Import Datastore via git**, enter the **Git URL**. Select the branch or tag to use.
3. Optionally, enter a **Username** and **Password**.
4. Select **Verify Peer Certificate** if desired.
5. Click **Submit**.

The new domain is imported via Git repository. Note that the domain is validated on import.

Changing Priority Order of Domains

Functions are executed depending on the order of Domain priority. Use this procedure to change the priority order of domains.

1. Navigate to **Automation > Automate > Explorer**. The default view is the Datastore.
2. Select the Domains you want to change the priority order for.
3. Click (**Configuration**), then (**Edit Priority Order of Domains**).
4. The list of Domains selected shows up. Note that you cannot change the priority of locked Domains and therefore locked Domains do not show up on the list.
5. Select one or more consecutive groups to move up or down to change their priority as required.
6. Click **Save**.

Creating a Namespace

1. Navigate to **Automation > Automate > Explorer**. The default view is the Datastore.
2. Navigate through the various Domains and Namespaces until you reach the desired location for your new Namespace.
3. Click (**Configuration**), then (**Add a New Namespace**).
4. Type in a unique **Name** and **Description**.
5. Click **Add**.

The new Namespace is created.

Creating a Class

1. Navigate to **Automation > Automate > Explorer**, navigate to the namespace you want to add a class to.
2. Click (**Configuration**), then (**Add a new Class**).
3. Type in a unique **Name** and **Description**.
4. If you want to use the schema from a class that has already been created, select it from the **Inherits From** dropdown. If the class that the new class inherits from changes, the new class will also change.
5. Click **Add**.

The new class is created and you can create a schema, add instances and methods.

Note:

For each class, create a schema if you did not choose to inherit from an existing class. The schema can include attributes, methods, assertions, and relationships.

Creating a Schema for a Class

This procedure shows you how to create a schema.

1. Navigate to **Automation > Automate > Explorer**, and click the class you want to define a schema for.
2. Click on the **Schema** tab.
3. Click (**Configuration**), then (**Edit selected Schema**).
4. Click (**Click to add a new field**) to create a new field.
5. Type in a **Name** for the new field.
6. From **Type**, select **Assertion**, **Attribute**, **Method**, **Relationship**, or **State**.
7. If applicable, select a **Data Type** and set a **Default Value**.
8. Type in a user friendly **Display Name** and **Description**.
9. Check **Sub** to enable the substitution syntax of **${}**. Uncheck it if you want to use that syntax as a regular string.
10. Fill in **Collect** and **Message** as required. **Collect** is used to roll up values from resolved relationships. For example, a relationship can resolve to a large object tree. Use **collect** to specify how to pull out data from those child objects into the current object. If you give **collect** a name value, it will store the method result in an attribute of the current object with that name.
11. **On Entry**, **On Exit**, **On Error**, **Max Retries**, and **Max Time** are fields used mostly for state machines. Leave blank if not applicable.
12. Click (**Add this entry**) to confirm the fields values.
13. For each new field, repeat steps 4 through 10.
14. When you have created all of the fields, click **Save**.

The class schema is created, and you can now add instances to it.

Note:

You may need to edit a class schema to reorder, add, edit, or remove a field. Classes define the order in which fields are processed and you may need to process some items before others.

Editing a Field in a Schema

This procedure describes how to edit schema fields.

1. Navigate to **Automation > Automate > Explorer**.
2. Click the class you want to define a schema for.
3. Click the **Schema** tab.
4. Click (**Configuration**), then (**Edit selected Schema**).
5. Make required changes to any of the definitions for the field.
6. To remove a field, click (**Click to delete this field from the schema**).
7. Click **Save** when you are finished editing the schema.

Once the schema is created, you can add instances and methods to the class.

Editing Schema Sequence

This procedure shows you how to change schema sequence.

1. Navigate to **Automation > Automate > Explorer**.
2. Click the class you want to change the schema sequence for.
3. Click the **Schema** tab.
4. Click (**Configuration**), then (**Edit Sequence**).
5. In the **Class Schema Sequencing** area, click the field you want to change the sequence for.
   - To move a field up in the order of resolving an instance, click (**Move selected field up**).
   - To move a field down in the order of resolving an instance, click (**Move selected field down**).
6. Click **Save** when you are finished editing the sequence.

Adding an Instance to a Class

This procedure shows you how to create an instance.

1. Navigate to **Automation > Automate > Explorer**.
2. Click the class you want to define a schema for.
3. Click the **Instances** tab.
4. Click (**Configuration**), then (**Add a new Instance**).
5. In the **Main Info** area, type in a **Name**, **Display Name** and **Description**.
6. In the **Fields** area, type in an appropriate value for each field, leave the field blank if no value is required, or use the default value.
7. Click **Add**.

Copying a Class or Instance

1. Navigate to **Automation > Automate > Explorer**. The default view is the Datastore.
2. Navigate through the various Domains and Namespaces until you reach the desired class or instance to copy.
3. Click (**Configuration**), then either (**Copy this Class**) or (**Copy this Instance**) depending on the object chosen.
4. Choose the target Domain in the **To Domain** drop-down menu.
5. The object retains the same path as the **From Domain** and overrides the class in **From Domain** if the **To Domain** has a high priority. You can also untick the **Copy to same path** option to specify a new Namespace.
6. Click **Add**.

Relationships

Relationships are used to connect to other instances in the **Automation Datastore**. Relationships are formed using URI syntax. The following can also be passed through a relationship:

- Use **#** to set the message to send to the item in the relationship.
- To pass an input to the method use **?** followed by the item to pass.
- If you want to use a substitution, the syntax is **$\{}** with the substitution located between the brackets.

Example	Explanation
`/Cloud/VM/Provisioning/Naming/Default#create`	This relationships uses the Default instance of the Naming class, which provides a means for other classes to name virtual machines. The relationship sends the `create` message to the class.
`/Cloud/VM/Provisioning/StateMachines/VMProvision_VM/AcquireMACAddress#$\{#ae_message}`	This relationships substitutes the message to send to the AcquireMACAddress instance of the VMProvision_VM class with the value in `ae_message`.
`/Cloud/VM/Retirement/Email/vm_retirement_emails?event=vm_retired`	Invokes the vm_retirement_emails instance of the Email class. Also sends the value `vm_retired` in the `event` attribute, which is used in the vm_retirement_emails method.
`/Service/Lifecycle/Retirement?service_id=$\{process#service_id}`	Invokes the Retirement instance of the Lifecycle class and send a replacement value in `process#service_id` to the `service_id` attribute.

Methods

Methods are pieces of code associated with a class or object to perform a task. Infrastructure Automation allows for Ruby methods or backing a method using an Ansible playbook. You can create your own methods or use relationships to link to pre-existing ones.

Infrastructure Automation ships with a core set of Ruby gems used by the Infrastructure Automation Rails Application. The Ruby gems in this set are subject to change. If you are calling gems using Automate that are no longer in this release, you can install them by using the `gem install` command.

While gems can be imported into automation methods using `require`, it is recommended that the authors of the automation methods clearly document the use of gems either in the core set or a custom set. It is the responsibility of the author of such custom automation to own the life cycle of any gem being referenced in those methods.

The Release Notes list Ruby gems that have been added, updated, or removed in the latest version of Infrastructure Automation.

Creating a Method

This procedure shows you how to create a method.

1. Navigate to **Automation > Automate > Explorer**, navigate to the class where you want to create a method.
2. Click the **Methods** tab.
3. Click (**Configuration**), (**Add a New Method**).
4. In the **Main Info** area, type in a **Name** and **Display Name**.
5. For **Location**, select **inline**. Once selected, you will be presented with a **Data** area in which to write or copy the script.
6. Click **Validate** to check the syntax.
7. Click **Add**.

Creating a Dynamic Content Dialog

The procedure describes the steps to create a dynamic content dialog.

1. Navigate to **Automation > Automate > Explorer**.
2. From the accordion menu, click **DOMAIN > Cloud > VM > Operations > Methods**.

   Note:

   DOMAIN must be a user-defined Domain and not the locked ManageIQ Domain. If necessary, you can copy the class from the ManageIQ domain into a custom domain.

   This example uses the **Cloud** Namespace but can also use the **Infrastructure** namespace.
3. Click (**Configuration**), then (**Add a new Instance**).
4. In the **Main Info** area, enter **Name** = *dynamic_list*, replacing *dynamic_list* with an appropriate name for the method.
5. Enter a **Display Name** and **Description**.
6. In the **Fields** area, enter **Value** = *dynamic_list*. Leave the other fields blank or use the default values.
7. Click **Add**.
8. Navigate to **Methods** tab.
9. In the **Main Info** area, enter **Name** = *dynamic_list* and populate the **Data** section with the example automate method below.
10. Click **Add**.
11. Set the automate entry point for the dialog control; use the new instance created in step four. You can create a new domain and copy the method to that domain.

```
#  Automate Method

dialog_field = $evm.object

# sort_by: value / description / none
dialog_field["sort_by"] = "value"

# sort_order: ascending / descending
#dialog_field["sort_order"] = "ascending"

# data_type: string / integer
dialog_field["data_type"] = "integer"

# required: true / false
# dialog_field["required"] = "true"

dialog_field["values"] = {1 => "one", 2 => "two", 10 => "ten", 50 => "fifty"}
dialog_field["default_value"] = 2
```

Creating a Playbook Automate Method

Infrastructure Automation can choose an Ansible playbook from a repository and execute it as a method. Each playbook method can take additional input parameters specified by the user.

Important:

- You must first sync your playbook repositories before using them to create a method. See Adding a Playbook Repository in *Managing Providers* for information on initial playbook repository set-up.
- Using Ansible playbooks to populate dynamic dialog fields is not recommended due to delay times caused by the overhead of interaction between systems.
- Only users with administrator privileges can run a service dialog based on a playbook automate method.

To create a playbook automate method:

1. Navigate to **Automation > Automate > Explorer**, then click on a domain under **Datastore**.
2. Under a namespace, select the class for which you want to create a new method.
3. Click the **Methods** tab.
4. Click (**Configuration**) then, (**Add a New Method**).
5. In the **Main Info** area, select **Playbook** from the **Type** dropdown menu.
6. Provide a **Name** and **Display Name**.
7. Select a playbook **Repository** from the list.
    1. Choose a **Playbook** to use.
    2. Select the **Machine Credential** the playbook will use when it runs.
    3. Select the **Vault Credential** to use.
    4. From the **Cloud Type** list, select a cloud provider.
    5. Choose the **Cloud Credential** that corresponds to the selected cloud type.
8. Specify the **Hosts** on which the playbook will run. Choose **Localhost** or provide unique values in the **Specify host values** field.
9. Set the **Max TTL** in minutes. The Time To Live (TTL) field allows you to set the maximum execution time for the playbook to run.
10. Select when to receive **Logging Output** from the options in the drop-down menu.
11. Use the **Escalate Privilege** toggle switch to enable user privilege escalation if credentials are called for during the playbook run.
12. Choose a **Verbosity** value to set the debug level for playbook execution.
13. Add required **Input Parameters** using the fields and values available. Click the to add additional input parameters.

    Note:

    Input parameters become extra vars, with substitution enabled. This overcomes the lack of a dialog which would normally allow for the input of additional information. For more information on extra vars, see the Ansible documentation.
14. Click **Add** when finished.

Once created, the domain including your playbook method can be exported to appliances in your testing or production environments or imported in appliances in multiple regions.

Important:

To import a domain with a playbook method, you must have an existing Ansible playbook on the destination environment with the same name or the import will fail.

See Exporting All Datastore Classes for information.

Passing variables between successive playbook methods

Automate designers can pass variables between successive Ansible playbook methods in a state machine using Infrastructure Automation.

To pass variables between Ansible playbook methods, use the **set_stats** module in your playbooks.

Example.

```
---
# This playbook prints a simple debug message
- name: Echo Hello, Fred and Wilma
  hosts: localhost

  tasks:
  - name: "playbook1 - testing input vars"
    set_stats:
      data:
        pb_var1a: ""
        pb_var1b: ""

  - debug: msg="Hello,  and "
```

For more information on the **set_stats** module, see the Ansible documentation.

Expression Methods

Infrastructure Automation additionally provides support for Expression Methods, that allow you to use advance search filters as Automate Methods, substituting the user input from Automate Objects. Expression methods have several distinct advantages, including: running directly in the worker appliance; removing the overhead of forking a DRb process to run the Automate Methods; no Ruby code required; and prebuilt for Dynamic Dialogs.

Input Parameters

Expression methods allow for substitution of user input through input parameters,

Input Parameter	Explanation
arg	The argument used in the expression. Each argument should employ the prefix **arg**. Example: **arg1**: the first argument in the expression; **arg2**: the second argument in the expression; **argn**: the nth argument in the expression.

Optional Input Parameters

Note:

If attributes and distinct are not specified we try to store the result in a variable called values with a hash consisting of id and name. This makes it compatible with our existing dynamic dialog result set.

Optional Input Parameter	Explanation
attributes	A comma delimited list of attributes to select from the resultant objects. This should me marked as an Array Type in the Input Parameters field.
distinct	A comma delimited list of attributes which are distinct in the resultant objects.This should me marked as an Array Type in the Input Parameters field.
result_obj	The object where the result data should be stored. (default: current object)
result_attr	The name of the attribute which stores the result. (default: values)
result_type	The result type hash or array (default: dialog_hash which matches to our dynamic dialog hash. Valid values are 'hash', 'dialog_hash', 'array', 'simple'
on_empty	The method behavior when the search returns an empty list.
error	Abort. (default: error)
default	The default value in case the result is empty and you select warn.

Creating an Expression Method

Expression methods allow you to use advance search filters as automate methods, substituting user input at runtime, and making them ideal for dynamic dialogs.

To create an expression method:

1. Navigate to **Automation > Automate > Explorer**, then click on a domain under **Datastore**.
2. Under a namespace, select the class for which you want to create a new method.
3. Click the **Methods** tab.
4. Click (**Configuration**) then, (**Add a New Method**).
5. In the **Main Info** area, select **expression** from the drop-down menu.
6. Provide a **Name** and **Display Name**.
7. Select an **Expression Object** from the drop-down menu.
8. In the **Expression** editor, create the expression by setting the controls and values used at runtime:

1. Using the drop-down menu, select the value to use. Based on your selection, choose or input additional values from the drop-down menus or text fields that appear.
2. In the **Contains** field, input a value or click **User will input the value**.
3. Click ✓ to complete the expression.

9. Add **Input Parameters** for each of the user input fields required.
    1. Click + to add a new parameter.
    2. Provide a **Name**, **Default Value** and select a **Data Type** for each parameter.
    3. Click ✓ to add the parameter.

    Note:

    If **User will input the value** is checked, arguments for each input parameter names using the prefix “arg”.

    For example, if there are 3 fields then the input parameter names should be arg1, arg2, and arg3. If there are two runtime parameters arg1 and arg2 must be defined in the input parameters. In the default value for these fields values can be substituted from other objects in the Automate Workspace.
10. Click **Add**.

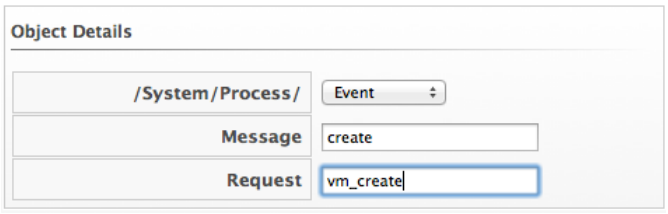
Simulation

After your model is designed, use the simulate page to test it. It allows you to see the results in tree and XML view.

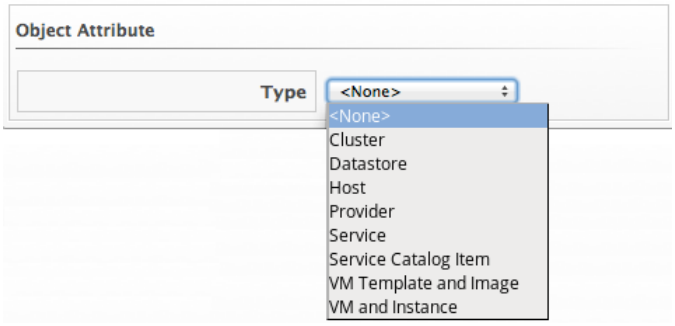
Simulating an Automate Model

This procedure shows you how to simulate an automate model.

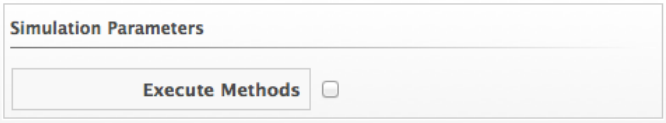
1. Navigate to **Automation > Automate > Simulation**.
2. In **Object Details**, select a type of object from **`/System/Process/`** that will initiate the model. The **Message** should be **`create`**. Type in the name of the **Request** where you are starting from.



3. Select the **Type** of item you want to run the simulation on. Then, select a specific one to use as the example.



4. Check **Execute Methods** if you want to perform the model and not just simulate it.



5. Type in the **Attribute/Value Pairs** fields if applicable.
6. Click **Submit**.

Click on the **Tree View** or **XML View** tabs to see results.

Importing, Exporting, and Resetting the Datastore

The **Automate Model** can be exported and imported as a YAML file. Infrastructure Automation allows you to back up your model by export. Red Hat may provide you with new or updated classes, and provides an import function for either a class or the entire model. Finally, you can reset the datastore to its default. Always be sure to export the current datastore before importing or resetting.

Note:

The datastores you are exporting or importing between must use the same Infrastructure Automation version.

Exporting All Datastore Classes

This procedure shows you how export datastore classes as a YAML file.

1. Navigate to **Automation > Automate > Import/Export**.
2. Click  (**Export all Datastore classes and instances to a file**).
3. Follow your browser's prompts to save the file.

The datastore is zipped and exported as a **`YAML`** file.

Importing Datastore Classes

This procedure shows you how to import datastore classes.

1. Navigate to **Automation > Automate > Import/Export**.
2. Export the datastore so that you have a backup.
3. Click **Choose file** to navigate to the location of the **.zip* file to import.
4. Click **Upload**.
5. After the file has uploaded, specify the following using the dropdown menus:
   - The domain to import into
   - The domain to import from
   - All namespaces you wish to import
6. Click **Commit** to finish importing the selected domain.

The datastore is imported from the uploaded **`YAML`** files.

Resetting Datastores to the Default

This procedure shows you how reset the datastores to the default.

1. Navigate to **Automation > Automate > Import/Export**.
2. Export the datastore so that you have a backup.
3. Click  (**Reset all components in the built-in domains**).
4. Click **OK**.

Invoking Automate

Automate Examples

This chapter describes the ways to invoke an Automate workflow. Automation can be initiated through an alert, an event, a Infrastructure Automation application, or a custom button. The same automation process can be re-used across more than one of these. For example, using automation to remove orphaned virtual machines and instances could be initiated by:

- An administrator request from the Infrastructure Automation console (from a custom button)
- An alert indicating the datastore has less than 20% free-space
- A virtual machine or instance unregistered event is detected

All invocations of an automate model must enter through the **`/System/Process`** namespace.

You can use automation to create a Provisioning Dialog that modifies the provisioning parameters of a VM service. For more information, see Example: Provisioning a VM service

Invoking Automate using a Custom Button

Invoke an Automate model by mapping an Ansible playbook or instance from the **`/System/Process/Request`** class to a custom button. Before creating the button, you need to have an Ansible playbook service catalog item or an instance in the **`/System/Process/Request`** class to map to it and a button group to assign it to.

Create buttons for a cluster, host, datastore, provider, virtual machines or cloud instances. When the button is clicked, the model or playbook will be invoked for the selected item. For each of these, you can have up to 15 buttons.

Creating a Custom Button Group

This procedure shows you how to create a custom button group.

1. Browse to menu: **Automation > Automate > Customization**.
2. Click the **Buttons** accordion.
3. From the **Object Types** tree, select the type of object you want to create the button group for.
4. Click **Configuration**, **Add a new Button Group**.
5. Type in a **Button Group Text** and **Button Group Hover Text**, and select the **Button Group Image** that you want to use.
6. If custom buttons have already been created, assign them to the button group. If not, see Creating a Custom Button.
7. Click **Add**.

The button group will show in the object type you added the button to.

Creating a Custom Button

This procedure shows you how to create a custom button.

Note:

Custom buttons can be migrated to other Infrastructure Automation appliances. See Migrating Custom Buttons for guidance on migrating custom buttons to a new Infrastructure Automation appliance.

1. Browse to menu: **Automation > Automate > Customization**.
2. Click the **Buttons** accordion.
3. From the **Object Types** tree, select the type of object you want to create the button for.
4. Click **Unassigned Buttons**.
5. Click **Configuration**, then **Add a new Button**.

   Note:

   If **Add a new Button** is not available, that means you have not created a button group for that object. To continue, create a button group first. See Creating a Custom Button.
6. Under the **Options** tab:
   1. Select the **Button** type from the list.
   2. Enter button **Text** and **Hover Text**, and select the **Icon** and **Icon Color** to use.
   3. Select a **Dialog** if applicable.
   4. Check **Open URL** to open a browser window for the custom URL returned when the button is executed.
   5. Choose a **Display For** option for the button.
   6. Select a **Submit** parameter to choose how to submit objects to automate. Selecting *Submit All* will pass all objects at once when the button is executed, while choosing *One by one* will run execute the button action each time per object.
7. Under the **Advanced** tab:
   1. Set button **Enablement**. Click **Define Expression** to access the expression editor tool. Enter **Disabled Button Text** to display when the custom button is disabled.
   2. Use **Visibility** to determine button appearance based on a custom expression. Click **Define Expression** to access the expression editor tool.

      Note:

      For more about setting visibility and enablement for a custom button, see Filtering.
   3. In **Object Details**, select **Request** from the **`/System/Process/`** dropdown. By default, the message is **`create`**. Do not change it.
   4. Enter a **Request** name for the **`/System/Process/Request`** instance.
   5. Enter the **Attribute/Value Pairs** fields if applicable.
   6. Set **Role Access**. Selecting *<By Role>* will display available roles. Check applicable roles.
8. Click **Add** when you have confirmed that the button accomplishes the task you want.

The button will show in the object type you added the button to.

Please note that there are two ways of creating custom buttons, and creation through the “Automation > Automate > Generic Objects” screen will necessitate adding an attribute/value pair that is not required when adding through the “Automation > Customization > Buttons” screen.

Creating an Ansible Playbook Button

Infrastructure Automation includes an option to create an Ansible Playbook custom button. This feature allows users to select a playbook to run as well as an inventory target to run it against. An Ansible playbook type button can be defined for any object type available.

Note:

An Ansible Playbook catalog item must exist in order to create an Ansible Playbook custom button. For more information, see Creating an Ansible Playbook Service Catalog Item in the Provisioning Virtual Machines and Hosts guide.

1. Navigate to **Automation > Automate > Customization**.
2. Click the **Buttons** accordion.
3. From the **Object Types** tree, select the type of object you want to create the button for.
4. Click **Unassigned Buttons**.
5. Click (**Configuration**), then (**Add a new Button**).

   Note:

   If (**Add a new Button**) is not available, that means you have not created a button group for that object. To continue, see Creating a Custom Button Group.
6. Select **Ansible Playbook** from the **Button Type** drop-down menu.
7. From the **Playbook Catalog Item** choose a playbook-backed catalog item to run.
8. Choose a host from the **Inventory** against which to run the playbook. If **Specific Hosts** is selected, input the IP address or DNS names for each host in the text field, separating each with a comma.

   Note:

   Infrastructure Automation supports two configurations for host value input:

   - To allow user-provided host values, set the custom button to **Specific Hosts** and leave the associated text field blank.
   - To use admin-specified host values, remove the **Hosts** field when creating the dialog the service uses. In this configuration, the field will not appear to users. See Service Dialogs for information on generating a service dialog.
9. Type in a **Text** and **Hover Text**, and select the **Icon** you want to use.
10. Select an **Icon Color** from the color selection palette that pops up.
11. Check **Open URL** to open a browser window for the custom URL returned when the playbook is run.
12. Select display options for the Ansible Playbook button from the **Display for** drop-down menu. Choose for the button to display in the list view, for single entities, or both.
13. Choose how to submit objects to automate by selecting an option from the **Submit** drop-down menu. Selecting **Submit all** will pass all objects at once when the playbook is executed, while choosing **One by one** will run the the playbook each time per object.
14. Click **Add** when you have confirmed that the button accomplishes the task you want.

Editing a Custom Button

This procedure shows you how to edit a custom button.

1. Navigate to **Automation > Automate > Customization**.
2. From the **Object Types** dropdown, select the type of object you want to edit the button for.
3. Click the button you want to edit.
4. Click (**Configuration**), (**Edit this Button**).
5. Modify as required.
6. Click **Save**.

Deleting a Custom Button

This procedure shows you how to delete a custom button.

1. Navigate to **Automation > Automate > Customization**, then select the **Buttons** accordion.
2. From the **Object Type** tree in the accordion menu, select the type of object you want to remove the button from.
3. Click (**Configuration**). then click (**Remove this button**).
4. Click **OK** to confirm that you want to delete this button.

Setting Enablement and Visibility for Custom Buttons

Infrastructure Automation adds methods for evaluating an expression to determine whether a custom button is visible and enabled. Each method has a target object, for example, a virtual machine or host, and expressions can set a custom button to visible, hidden, or disabled.

Note:

Filtering works on single objects and is not applicable to lists.

To apply filtering actions to a custom button:

1. Navigate to **Automation > Automate > Customization**.
2. Click the **Buttons** accordion. Select the custom button to use.
3. Click (**Configuration**), then (**Edit this Button**).
4. Click the **Advanced** tab.
5. To set enablement filtering on a custom button:
    1. Under **Enablement**, click on **Define Expression**.
    2. Create a visibility expression using the expression editor tools.
    3. Click (**Confirm**) when finished defining the expression.
    4. Provide **Disabled Button Text** in the field.
6. To set visibility filtering on a custom button:
    1. Under **Visibility**, click on **Define Expression**.
    2. Create a visibility expression using the expression editor tools.
    3. Click (**Confirm**) when finished defining the expression.
7. Click **Save**.

Using a Custom Button

This procedure shows you how to use custom buttons to invoke a cluster, host, datastore, provider, virtual machine or instance.

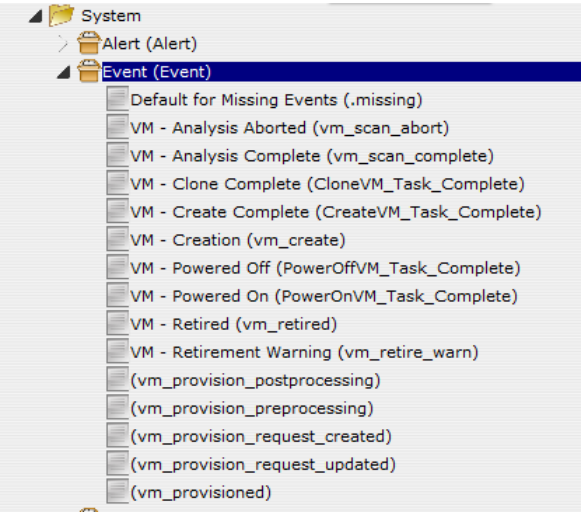
1. Go to the page for the item that you created a button for.
2. Click the custom button group from the taskbar, and then your custom button.

The automate model is invoked for the specified item.

Initiating Automate from an Event

You can also use a Infrastructure Automation Policy Event to initiate automation. You can either use the provided Raise Automation Event action or create a custom automation action. The first case will start in the **`/System/Process`** class, but then go to the Event that initiated the Automate model in the **`/System/Process/Event Class`**. If you create your own custom action, it will start from the **`/System/Process`** class and then go to the **`/System/Process/Request Class`** instead.

For example, suppose that you always want the same Automate model to occur when a virtual machine is created. You would use the Raise Automation Event Action. There are instances in the **`/System/Process/Event Class`** for the following Events that you can select as part of a Policy:



Creating a Policy for Automate

This procedure shows you how to create a policy for automate.

1. Navigate to **Control > Explorer**.
2. Click the **Policies** accordion, and select **Control Policies**.
3. Select **Vm Control Policies**.

4. Click (**Configuration**), then (**Add a New Control Vm Policy**).
5. Type in a **Description**.
6. Uncheck **Active** if you do not want this policy processed even when assigned to a resource.
7. Click **Add**. You are brought to the page where you add conditions and events to your new policy.
8. Click (**Configuration**), then (**Edit this Policy's Event assignments**).
    - Check the events you want to use to send to an **Automate Model**.
    - Click **Save**.
    - From the **Events** area, click on the **Description of the Event** you want to assign an action to.
    - Click (**Edit Actions for this Policy Event**).
9. Select **Raise Automation Event**, and click (**Move selected Actions into this Event**).
10. Click **Save**.

You can now assign this policy to a **Policy Profile**. Then, assign the policy profile to the virtual machines. Every time this event happens on the virtual machine the appropriate Automate Model will be initiated.

Note:

If you want the policy to initiate an Automate Model from the `/System/Process/Request` class, then you can create your own custom action. Be sure to have an instance in the `/System/Process/Request` class for it to map to.

Creating a Custom Automate Action

This procedure shows you how to create a custom Automate action.

1. Navigate to **Control > Explorer** accordion.
2. Click the **Actions** accordion.
3. Click (**Configuration**), then (**Add a new Action**).
4. Type in a **Description** for the Action.
5. Select **Invoke a Custom Automation** from **Action Type**.
6. In **Custom Automation**,
    - For **Message**, type `create`.
    - For **Request**, type in the name of the instance of the `/System/Process/Request Class` in the second.
7. Type in the **Attribute/Value Pairs** fields if applicable.
8. Click **Add**.

The action is created and can be added to a policy.

Objects

Virtual Machine Properties

When using these items in a method, prefix them with `vm.`. For example: `vm.storage_id`.

Friendly Name or Description	Raw Column Name
Allocated Disk Storage	allocated_disk_storage
Autostart	autostart
Blackbox Exists	blackbox_exists
Blackbox Validated	blackbox_validated
Boot Time	boot_time
Busy	busy
Cluster	ems_cluster_name
Configuration XML	config_xml
Connection State	connection_state
CPU Affinity	cpu_affinity
CPU Limit	cpu_limit
CPU Reserve	cpu_reserve
CPU Reserve Expand	cpu_reserve_expand

Friendly Name or Description	Raw Column Name
CPU Shares	cpu_shares
CPU Shares Level	cpu_shares_level
Created on Time	ems_created_on
Currently Used Space	used_storage_by_state
Datastore Path	v_datastore_path
Date Created	created_on
Date Updated	updated_on
Description	description
Ems	ems_id
Evm Owner	evm_owner_id
Evm Owner Email	evm_owner_email
Evm Owner Name	evm_owner_name
Infrastructure Automation Unique ID (Guid)	guid
Format	format
Host	host_id
Host Name	host_name
Id	id
Is a Template	v_is_a_template
Last Analysis Attempt On	last_scan_attempt_on
Last Analysis Time	last_scan_on
Last Compliance Status	last_compliance_status
Last Compliance Timestamp	last_compliance_timestamp
Last Perf Capture On	last_perf_capture_on
Last Sync Time	last_sync_on
Location	location
Memory Limit	memory_limit
Memory Reserve	memory_reserve
Memory Reserve Expand	memory_reserve_expand
Memory Shares	memory_shares
Memory Shares Level	memory_shares_level
Name	name
OS Name	os_image_name
Owner	owner
Paravirtualization	paravirtualization
Parent Cluster	v_owning_cluster
Parent Datacenter	v_owning_datacenter
Parent Folder (Hosts & Clusters)	v_owning_folder
Parent Folder (VMs & Templates)	v_owning_blue_folder
Parent Folder Path (Hosts & Clusters)	v_owning_folder_path
Parent Folder Path (VMs & Templates)	v_owning_blue_folder_path
Parent Host Platform	v_host_vmm_product
Parent Resource Pool	v_owning_resource_pool
Pct Free Disk	v_pct_free_disk_space
Platform	platform
Power State	power_state
Previous State	previous_state
Registered	registered
Reserved	reserved
Retired	retired
Retirement	retirement
Retires On	retires_on
Service	service_id
Smart	smart
Standby Action	standby_action
State Changed On	state_changed_on
Storage	storage_id
Storage Name	storage_name
Template	template
Thin Provisioned	thin_provisioned
Tools Status	tools_status
Total Provisioned Space	provisioned_storage
Total Snapshots	v_total_snapshots
Total Used Disk Space	used_disk_storage
Uid Ems	uid_ems

Friendly Name or Description	Raw Column Name
Uncommitted Space	uncommitted_storage
Used Storage	used_storage
V Pct Used Disk Space	v_pct_used_disk_space
VDI Available	vdi_available
VDI Connection DNS Name	vdi_connection_dns_name
VDI Connection Logon Server	vdi_connection_logon_server
VDI Connection Name	vdi_connection_name
VDI Connection Remote IP Address	vdi_connection_remote_ip_address
VDI Connection Session Name	vdi_connection_session_name
VDI Connection Session Type	vdi_connection_session_type
VDI Connection URL	vdi_connection_url
VDI Endpoint IP Address	vdi_endpoint_ip_address
VDI Endpoint MAC Address	vdi_endpoint_mac_address
VDI Endpoint Name	vdi_endpoint_name
VDI Endpoint Type	vdi_endpoint_type
VDI User Appdata	vdi_user_appdata
VDI User DNS Domain	vdi_user_dns_domain
VDI User Domain	vdi_user_domain
VDI User Home Drive	vdi_user_home_drive
VDI User Home Path	vdi_user_home_path
VDI User Home Share	vdi_user_home_share
VDI User Logon Time	vdi_user_logon_time
VDI User Name	vdi_user_name
Vendor	vendor
Version	version
VMsafe Agent Address	vmsafe_agent_address
VMsafe Agent Port	vmsafe_agent_port
VMsafe Enable	vmsafe_enable
VMsafe Fail Open	vmsafe_fail_open
VMsafe Immutable VM	vmsafe_immutable_vm
VMsafe Timeout (ms)	vmsafe_timeout_ms

Virtual Machine Properties

Methods for use in Ruby Scripts

To use one of these in one of your own Ruby methods, use the syntax of **`vm.method`**. For example, to reboot the guest operating system, use **`vm.rebootGuest`**.

Method	Description
start	Start Virtual Machine container.
stop	Stop Virtual Machine container.
suspend	Suspend Virtual Machine container.
unregister	Unregister Virtual Machine.
collect_running_processes	Collect the running processes from a started Virtual Machine.
shutdownGuest	Shutdown the guest operating system of the VM container. Requires VMware tools (or vendors tools) installed on the guest.
standbyGuest	Put the guest operating system into standby. Requires VMware tools (or vendors tools) installed on the guest.
rebootGuest	Reboot the guest operating system. Requires VMware tools (or vendors tools) installed on the guest.

Host Properties

When using these items in a method, prefix them with host, such as **`host.ems_id`**.

Friendly Name or Description	Raw Column Name
All Enabled Ports	all_enabled_ports
Annotation	v_annotation
Authentication Status	Authentication_status
Connection State	connection_state
CPU usage MHz rate average over time period	cpu_usagemhz_rate_average_avg_over_time_period
CPU usage MHz rate high over time period	cpu_usagemhz_rate_average_high_over_time_period
CPU usage MHz rate low over time period	cpu_usagemhz_rate_average_low_over_time_period
Custom Attribute 1	custom_1
Custom Attribute 2	custom_2
Custom Attribute 3	custom_3
Custom Attribute 4	custom_4
Custom Attribute 5	custom_5
Custom Attribute 6	custom_6
Custom Attribute 7	custom_7
Custom Attribute 8	custom_8

Friendly Name or Description	Raw Column Name
Custom Attribute 9	custom_9
Date Created	created_on
Derived memory usage average over time period	derived_memory_used_avg_over_time_period
Derived memory usage high over time period	derived_memory_used_high_over_time_period
Derived memory usage low over time period	derived_memory_used_low_over_time_period
Ems	ems_id
Enabled Inbound Ports	enabled_inbound_ports
Enabled Outbound Ports	enabled_outbound_ports
Enabled Run Level 0 Services	enabled_run_level_0_services
Enabled Run Level 1 Services	enabled_run_level_1_services
Enabled Run Level 2 Services	enabled_run_level_2_services
Enabled Run Level 3 Services	enabled_run_level_3_services
Enabled Run Level 4 Services	enabled_run_level_4_services
Enabled Run Level 5 Services	enabled_run_level_5_services
Enabled Run Level 6 Services	enabled_run_level_6_services
Enabled TCP Inbound Ports	enabled_tcp_inbound_ports
Enabled TCP Outbound Ports	enabled_tcp_outbound_ports
Enabled UDP Inbound Ports	enabled_udp_inbound_ports
Enabled UDP Outbound Ports	enabled_udp_outbound_ports
EVM Unique ID (Guid)	guid
Hostname	hostname
Id	id
IP Address	ipaddress
Last Compliance Status	last_compliance_status
Last Compliance Timestamp	last_compliance_timestamp
Last Perf Capture On	last_perf_capture_on
Last Analysis Time	last_scan_on
Name	name
OS Name	os_image_name
Platform	platform
Power State	power_state
Region Description	region_description
Region Number	region_number
Reserved	reserved
Service Names	service_names
Settings	settings
Smart	smart
SSH Root Access	ssh_permit_root_login
Uid Ems	uid_ems
Date Updated	updated_on
User Assigned Os	user_assigned_os
Parent Cluster	v_owning_cluster
Parent Datacenter	v_owning_datacenter
Parent Folder (Hosts & Clusters)	v_owning_folder
Total Datastores	v_total_storages
Total VMs	v_total_vms
VMM Build Number	vmm_buildnumber
VMM Platform	vmm_product
VMM Vendor	vmm_vendor
VMM Version	vmm_version

Provider Properties

When using these items in a method, prefix them with **`ems`**, such as **`ems.ems_id`**.

Friendly Name or Description	Raw Column Name
Aggregate VM CPUs	aggregate_vm_cpus
Aggregate VM Memory	aggregate_vm_memory
CPU Ratio	v_cpu_vr_ratio
CPU Usage MHZ Rate Average High Over Time Period	cpu_usagemhz_rate_average_high_over_time_period”
CPU Usage MHZ Rate Average Low Over Time Period	cpu_usagemhz_rate_average_low_over_time_period
CPU Usage MHZ Rate Average Over Time Period	cpu_usagemhz_rate_average_avg_over_time_period
Date Created	created_on
Date Updated	updated_on
Derived Memory Usage Rate Average High Over Time Period	derived_memory_used_high_over_time_period

Friendly Name or Description	Raw Column Name
Derived Memory Usage Rate Average Low Time Period	derived_memory_used_low_over_time_period
Derived Memory Usage Rate Average Over Time Period	derived_memory_used_avg_over_time_period
Distributed Resource Scheduler Automation Level	drs_automation_level
Distributed Resource Scheduler Enabled	drs_enabled
Distributed Resource Scheduler Migration Threshold	drs_migration_threshold
EMS ID	ems_id
EVM Zone	zone_name
High-Availability Admission Control	ha_admit_control
High-Availability Enabled	ha_enabled
High-Availability Max Failures	ha_max_failures
Id	id
Last Performance Data Captured	last_perf_capture_on
Last Smart State Analysis	last_scan_on
Memory Ratio	v_ram_vr_ratio
Name	name
Parent Datacenter	v_parent_datacenter
Qualified Description	v_qualified_desc
Region Description	region_description
Region Number	region_number
Reserved	reserved
Total CPU Speed	aggregate_cpu_speed
Total Hosts	total_hosts
Total Memory	aggregate_memory
Total Number of Logical CPUs	aggregate_logical_cpus
Total Number of Physical CPUs	aggregate_physical_cpus
Total Vms	total_vms
Unique Identifier	uid_ems

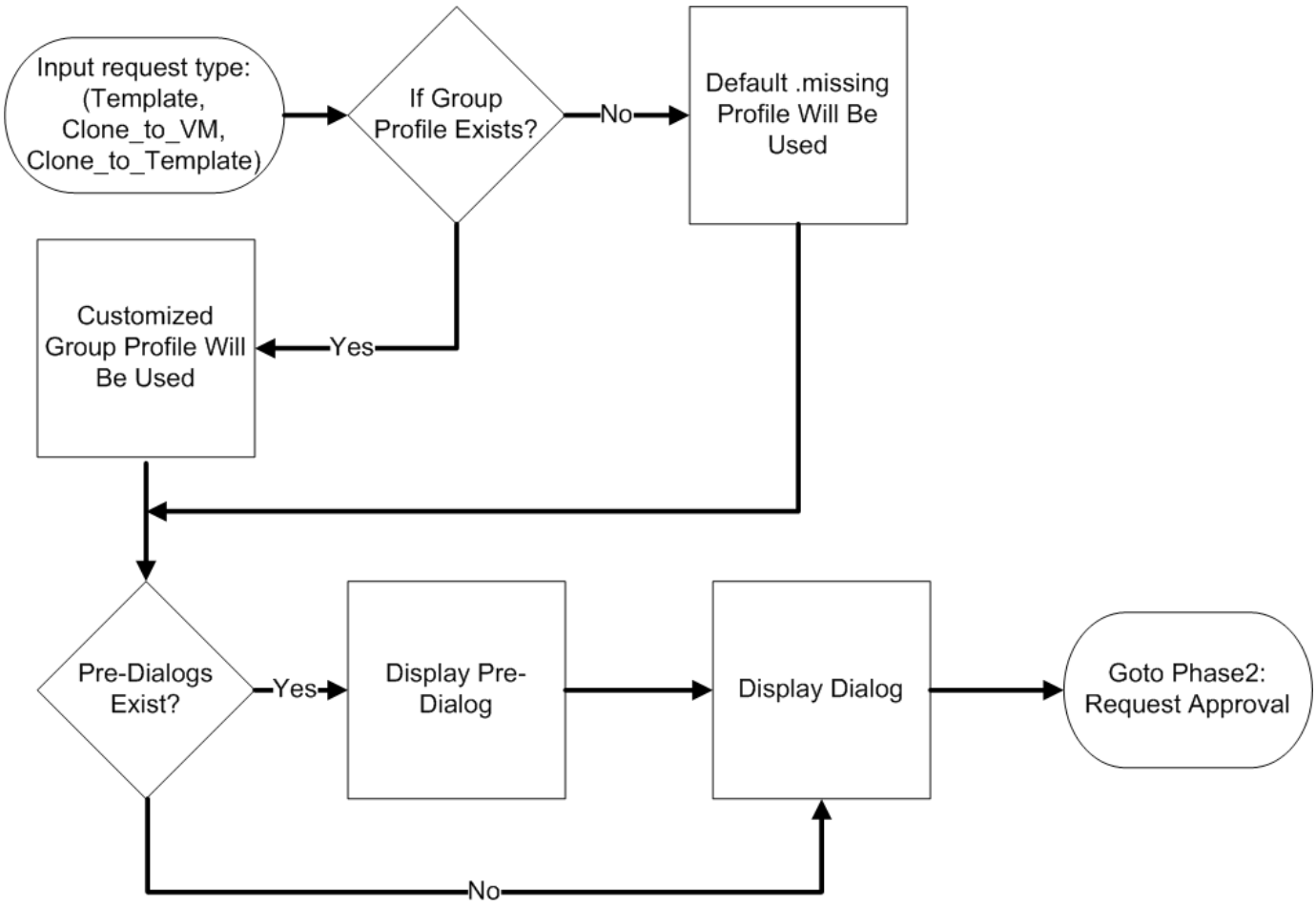
Storage Properties

When using these items in a method, prefix them with **`storage`**, such as **`storage.name`**.

Friendly Name or Description	Raw Column Name
Date Created	created_on
Date Updated	updated_on
Disk Files Percent of Used	v_disk_percent_of_used
Free Space	free_space
Free Space Percent of Total	v_free_space_percent_of_total
Id	id
Last Analysis Time	last_scan_on
Last Perf Capture On	last_perf_capture_on
Location	location
Multiple Host Access	multiplehostaccess
Name	name
Non-VM Files Percent of Used	v_debris_percent_of_used
Other VM Files Percent of Used	v_vm_misc_percent_of_used
Provisioned Space Percent of Total	v_provisioned_percent_of_total
Reserved	reserved
Size of Non-VM Files	v_total_debris_size
Size of Other VM Files	v_total_vm_misc_size
Size of VM Memory Files	v_total_vm_ram_size
Size of VM Snapshot Files	v_total_snapshot_size
Snapshot Files Percent of Used	v_snapshot_percent_of_used
Store Type	store_type
Total Hosts	v_total_hosts
Total Managed Registered Vms	total_managed_registered_vms
Total Managed Unregistered Vms	total_managed_unregistered_vms
Total Provisioned Space	v_total_provisioned
Total Space	total_space
Total Unmanaged Vms	total_unmanaged_vms
Total VMs	v_total_vms
Uncommitted	uncommitted
Used Space	v_used_space
Used Space Percent of Total	v_used_space_percent_of_total
VM Memory Files Percent of Used	v_memory_percent_of_used

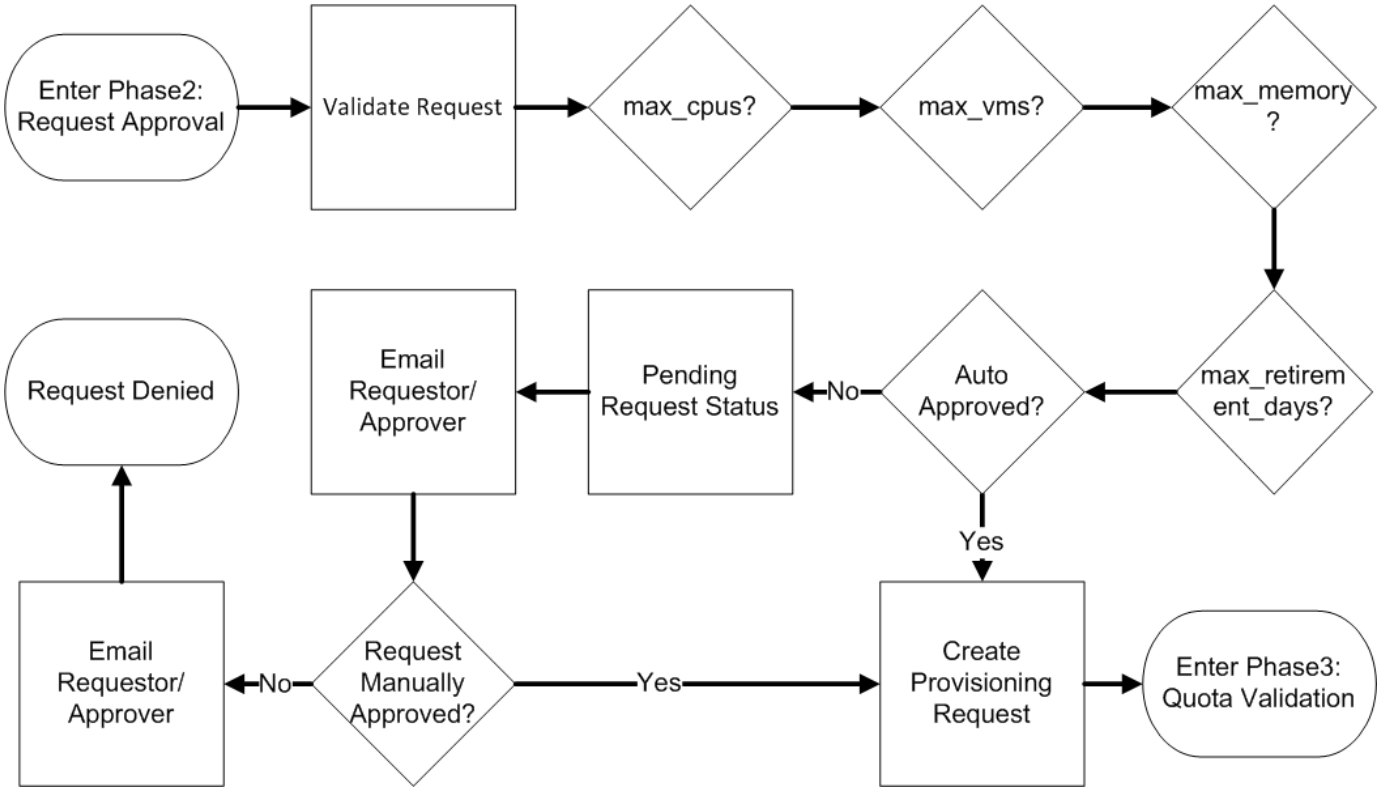
FAQs and Flows

Phase 1: Create Provision Request



Question	Answer
Where do I create a new provisioning profile based on a users LDAP group?	Navigate to [`VM / Provisioning / Profile`] of either the **Cloud** or **Infrastructure** namespace in your domain.
Where can I specify a pre-dialog to present to a Requester in their LDAP group?	Custom pre-dialogs can be defined in [`VM / Provisioning /
Profile / `\] of either the **Cloud** or **Infrastructure** namespace in your domain.	
I would like to customize our dialogs. Where are all the dialogs kept?	All dialogs are located on each Infrastructure Automation appliance in the `/var/www/miq/vmdb/db/fixtures` directory.
What happens if I do not specify any profiles for provisioning?	Infrastructure Automation searches for a matching LDAP group in the [`VM / Provisioning / Profile`] class of either the **Cloud** or **Infrastructure** namespace in your domain. If an LDAP profile is NOT found then Infrastructure Automation will use the `missing class instance`.

Phase 2: Request Approval



Question	Answer
Where can I specify auto-approval values on a per virtual machine template basis?	Tags can be assigned to templates in the form of [**prov_max_vms**, **prov_max_cpus**, **prov_max_memory**, **prov_max_retirement_days**].
Where can I modify the default Auto-Approval values?	These values can be set in the [`Service / Provisioning / StateMachines /
ServiceProvisionRequestApproval / Default`] class instance in your domain.	
How can I customize the email that is sent when a request is approved?	The Request Approved email message can be modified in [**`VM / Provisioning / Email / MiqProvisionRequest_Approved`**] in either the **Cloud** or **Infrastructure** namespace of your domain.
How can I customize the email that is sent when a request is denied?	The Request Denied email message can be modified in [**`VM / Provisioning / Email / MiqProvisionRequest_Denied`**] in either the **Cloud** or **Infrastructure** namespace of your domain.
How can I customize the email that is sent when a request is not Auto-approved?	The Request Pending email message can be modified in [**`VM / Provisioning / Email / MiqProvisionRequest_Denied`**] in either the **Cloud** or **Infrastructure** namespace of your domain.
If a Request Approval requires manual approval, how does an Approver approve the request?	Log into Infrastructure Automation as an approver/admin and navigate to **Virtual Machines > Requests** and then click on the request.

Phase 3: Quota Validation

Question	Answer
Where in Infrastructure Automation can I set default quota thresholds for users and groups?	These values can be set in the [`VM
Provisioning / StateMachines / ProvisionRequestQuotaVerification`] class instance of either the **Cloud** or **Infrastructure** namespace in your domain.	
Where in Infrastructure Automation can I set individual and group quota thresholds?	Tags can be assigned to groups or users by navigating to **Configuration > Access Control**. The following are valid tags that can be assigned to group or individual users: [**quota_max_cpu**, **quota_max_memory**, **quota_max_storage**].
Where can I customize the way our virtual machines are named?	Virtual machine naming conventions can be altered using the methods in the [`VM /
Provisioning / Naming`] class of either the **Cloud** or **Infrastructure** namespace in your domain.	
How can I customize the email that is sent when a request is denied?	The Request Denied email message can be modified in the [`VM /
Provisioning / Email / MiqProvisionRequest_Denied`] in either the **Cloud** or **Infrastructure** namespace of your domain.	

Phase 4: Provisioning

Target Type: Cloning a Template to a Virtual Machine.

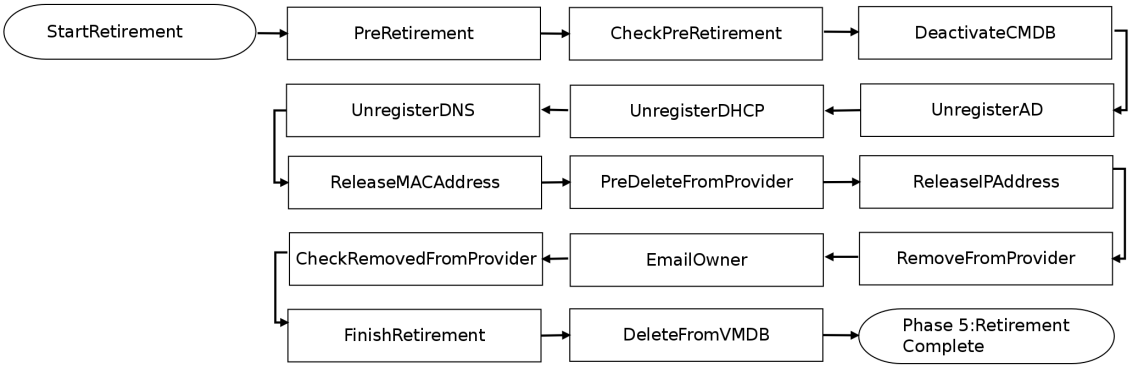
Target Type: Cloning a Template to a Virtual Machine

Target Type: Clone to Template.

Target Type: Clone to Template

Question	Answer
Where can I customize the email message that is sent upon provisioning completion?	This can be customized using the [`VM / Provisioning / Email
/ MiqProvision_Complete`] in either the **Cloud** or **Infrastructure** namespace of your domain.	
Where can I change what is put into the virtual machines Annotation after provisioning?	These settings can be modified by leveraging the `_PreProvision Ruby` methods in [`VM / Provisioning / StateMachines /
Methods`] in either the *Cloud</emphasis> or **Infrastructure** namespace of your domain.	
Where do I set the default VC folder location for provisioning virtual machines?	This can be modified by leveraging by leveraging the `_PreProvision Ruby` methods in [`VM / Provisioning / StateMachines /
Methods`] in either the *Cloud</emphasis> or **Infrastructure** namespace of your domain.	
Where can I modify the virtual machine customization spec mapping?	This can be modified by leveraging by leveraging the `_PreProvision
Ruby `methods in \[`VM / Provisioning / StateMachines / Methods`] in either the *Cloud</emphasis> or **Infrastructure** namespace of your domain.	
Where can I modify the Clone_to_Template state_machine?	Navigate to [`VM / Provisioning / StateMachines / VMProvision_VM / template`] in either the **Cloud** or **Infrastructure** namespace of your domain.
Where can I modify the Clone_to_VM state_machine?	Navigate to [`VM /
Provisioning / StateMachines / VMProvision_VM / clone_to_vm`] in either the **Cloud** or **Infrastructure** namespace of your domain.	

Phase 5: Retirement



Question	Answer
Where can I customize the email message that is sent upon completion of virtual machine retirement?	This can be customized using the [`VM /
Retirement / Email / vm_retirement_emails`] in either the **Cloud** or **Infrastructure** namespace of your domain.	
Where can I customize the email message that is sent during virtual machine retirement warnings?	This can be customized using the [`VM /
Retirement / Email / vm_retirement_emails`] in either the **Cloud** or **Infrastructure** namespace of your domain.	
If I want to customize what gets called during the retirement phase where should I look?	This can be customized using the [`VM / Retirement
/ StateMachines / VMRetirement`] in either the **Cloud** or **Infrastructure** namespace of your domain.	
How can I extend the virtual machine retirement date an additional number of days?	Create a custom button for virtual machines that launches [`/System/Request/vm_retire_extend`]. Then navigate to the [`VM /
Retirement / Email / vm_retire_extend`] Ruby method in the **Cloud** or **Infrastructure** namespaces and set the` vm_retire_extend_days` value.	

Inline Method to Create a Provision Request

Ruby method

```
# Infrastructure Automation Automate Method
#
$evm.log("info", "Infrastructure Automation Automate Method  Building VM Provisioning Request Started")
#
```

```
prov= $evm.root['miq_provision']

  # arg1 = version
  args = ['1.1']
  # arg2 = templateFields
  args << {'name' => 'App'}
  # arg3 = vmFields
  args << {'vm_name' => 'CRM_APP', 'request_type' => 'template'}
  # arg4 =  requester
  args << {'owner_email' => 'admin@asd.com', 'owner_last_name' => 'Admin', 'owner_first_name' => 'Admin', 'user_name' =>
'admin'}
  # arg5 =  tags
  args << {'crm' => 'true'}
  # arg6 =  WS Values
  args << nil
  # arg7 = emsCustomAttributes
  args << nil
  # arg8 = miqCustomAttributes
  args << nil

$evm.log("info","Building provisioning request with the following arguments: <#{args.inspect}>")
result = $evm.execute('create_provision_request', *args)
```

Migrating Custom Buttons

Migrating Custom Buttons

This workflow demonstrates how to export custom buttons from one Infrastructure Automation appliance and import them in another Infrastructure Automation appliance.

1. Export custom buttons from the source Infrastructure Automation appliance:
    1. SSH into the Infrastructure Automation appliance as **root**.
    2. Create a temporary directory:

       ```
       # mkdir /tmp/custom_buttons
       ```

    3. Browse to the Virtual Management Database (VMDB) directory:

       ```
       # vmdb
       ```

    4. Export custom buttons using the following **rake** command:

       ```
       # rake evm:export:custom_buttons -- --directory /tmp/custom_buttons
       ```

    5. Confirm the **yaml** file was created by navigating to the new directory:

       ```
       # cd /tmp/custom_buttons
       ```

2. Copy the directory to the target Infrastructure Automation appliance:

   ```
   scp -r /tmp/custom_buttons/ hostname:/tmp/custom_buttons
   ```

3. Import custom buttons on the target Infrastructure Automation appliance:
    1. SSH to the target Infrastructure Automation appliance as **root**.
    2. Browse to the VMDB directory:

       ```
       # vmdb
       ```

    3. Import custom buttons using the following **rake** command:

       ```
       # rake evm:import:custom_buttons -- --source /tmp/custom_buttons
       ```

To verify successful import of custom buttons:

1. Log in to the target Infrastructure Automation appliance.
2. Browse to menu: **Automation > Automate > Customization**.
3. Click **Buttons** in the accordion menu.

Imported buttons will appear under the parent **Object Type**.

Infrastructure Management API Documentation

In this guide you will find a detailed overview of REST API, reference material, appendices, and examples. The guides provide information on how to integrate Infrastructure Automation – Infrastructure Management with external applications. It details the specification of the Infrastructure Management REST API, which is implemented as standard REST HTTP requests and responses of content type JSON.

- API Overview
- API Reference
- API Appendices
- API Examples
- Prerequisites

Infrastructure Management API Overview

In this guide you will find a detailed overview of Infrastructure Automation – Infrastructure Management REST API.

- REST API Authorization
- Password Updates
- CRUD
- HTTP Basics
- JSON Specification
- Query Specification
- Bulk Querying
- Collection Filtering
- Pagination

Prerequisites for Infrastructure Management APIs

Infrastructure Management can be installed as either a VM aplliance or as a containerized deployment. To issue API calls to the Infrastructure Management that is installed as a containerized deployment, obtain the authorization token and the URL hostname as described below.

Set the values for the following variables:

Variable	Description
IM_USER	IM_USER is the user you want to use. You can use either the default admin user or any user in the user group that you have configured in the im-iminstall CR. See https://ibm.biz/IM_containerized_pod.
IM_PASSWORD	Set the user password variable.
CP_CONSOLE_HOSTNAME	Set the CP console hostname with the command `oc -n <IA_installation_namespace> get routes cp-console -o=jsonpath='{.spec.host}'`.

Where **<IA_installation_namespace>** is the namespace that Infrastructure Automation is installed in.

1. Run the command to get the bearer token for the user

   ```
   export IM_BEARER_TOKEN=$(curl -ks -X POST  -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"  -d "grant_type=password&username=${IM_USER}&password=${IM_PASSWORD}&scope=openid" https://${CP_CONSOLE_HOSTNAME}/idprovider/v1/auth/identitytoken|jq -r .access_token)
   ```

2. Run the command to get the URL hostname,

   export IM_HOSTNAME=$(oc -n cp4aiops get route | grep inframgmt | awk '{print $2}')

3. After the variables IM_BEARER_TOKEN and IM_HOSTNAME are defined, use them to issue the API calls. A curl command should be in this format,

```
curl  -H "Authorization: Bearer ${IM_BEARER_TOKEN}" \
     ..............................  \
     "https://${IM_HOSTNAME}/api/......"
```

For example, to issue a GET request to the Infrastructure Management that is installed as a containerized deployment, specify,

```
curl  -H "Authorization: Bearer ${IM_BEARER_TOKEN}" \
     -k -i -X GET -H "Accept: application/json"   \
     "https://${IM_HOSTNAME}/api/providers"
```

Note: To issue API calls to the Infrastructure Management that is installed as a VM appliance, follow the instructions in the subsequent sections in Infrastructure Management API Guide. To issue API calls to the Infrastructure Management that is installed as a containerized deployment, replace the HTTP authorization header and the URL segment (http://localhost:3000) with those described above.

- REST API Authorization
  - Specifying Authorization Group
  - Specifying an Invalid Authorization Group

REST API Authorization

When authenticating to the appliance, the current group of the authenticated user will be used for determining the authorization of what the user can see and do using the REST API. While this is acceptable when users only belong to a single group, this falls short when users are authenticating using their External Authentication credentials (IPA, LDAP or AD) where they may belong to multiple groups.

Specifying Authorization Group

This section describes how to specify an authorization group for the user. The mechanism described here is honored for both methods of authentication:

- Basic Authentication
- Token based Authentication

The user's group is to be specified in the new HTTP Header:

```
X-MIQ-Group
```

and is to contain the user's group name.

Example request using Basic Authentication:

```
$ curl --user username:password
       -i -X GET -H "Accept: application/json"
       -H "X-MIQ-Group: auditors"
       http://localhost:3000/api/vms
```

Example request using Token based Authentication:

```
$ curl -i -X GET -H "Accept: application/json"
      -H "X-Auth-Token: 93d98fd44cc5a401ff385ba87b21fd13"
      -H "X-MIQ-Group: auditors"
      http://localhost:3000/api/vms
```

The validated users's group will be in effect for the duration of the REST API request. Multiple requests can be made as a particular user but with different authorization groups concurrently.

Specifying an Invalid Authorization Group

In case an invalid authorization group is specified, a 401 Unauthorized is returned:

```
HTTP/1.1 401 Unauthorized
X-Frame-Options: SAMEORIGIN
...

{
  "error" : {
    "kind" : "unauthorized",
    "message" : "Invalid Authorization Group <invalid_group_name> specified",
    "klass" : "ApiController::AuthenticationError"
  }
}
```


- Password Updates

Password Updates

When the appliance is configured for Database authentication, users can update their passwords via the REST API.

Note that if an authenticated REST API user is entitled to edit users, they can edit other users' passwords, e.g. admin account.

Passwords can be updated via the **edit** post action on the user's resource as follows:

```
POST /api/users/:id

{
  "action" : "edit",
  "resource" : { "password" : "<new_password>" }
}
```

So for updating one's password, the */api/users/:id* being updated must be the authenticated REST API user's object.

- CRUD Request Examples
  - Show a collection of Resources
  - Return a single Resource
  - Create a Resource
  - Update a Resource
  - Delete a Resource

CRUD Request Examples

The following examples show the basic CRUD operations (Create, Read, Update, Delete) using the REST API.

For Authentication, we are showing here basic authentication via the *–user admin:smartvm* credentials argument. For multiple API calls it is recommended to access the REST API via token based authentication, please refer the the Authentication Page for details.

Show a collection of Resources

Get a collection of services: **GET /api/services**

```
curl --user admin:smartvm
     -i -X GET -H "Accept: application/json"
     http://localhost:3000/api/services
```

Return a single Resource

Return a single service: **GET /api/services/:id**

```
curl --user admin:smartvm
     -i -X GET -H "Accept: application/json"
     http://localhost:3000/api/services/1
```

Create a Resource

Create a new provider: **POST /api/providers**

```
curl --user admin:smartvm
     -i -X POST -H "Accept: application/json"
     -d '{
           "type"       : "ManageIQ::Providers::Redhat::InfraManager",
           "name"       : "RHEVM Provider",
           "hostname"   : "rhevm.local.com",
           "ipaddress"  : "192.168.5.1",
           "credentials" : {
             "userid"   : "admin",
             "password" : "12345"
           }
     }'
     http://localhost:3000/api/providers
```

Update a Resource

Update the name of a service: **PUT /api/services/:id**

```
curl --user admin:smartvm
     -i -X PUT -H "Accept: application/json"
     -d '{ "name" : "updated service name" }'
     http://localhost:3000/api/services/1
```

Delete a Resource

Delete a service: **DELETE /api/services/:id**

```
curl --user admin:smartvm
     -i -X DELETE -H "Accept: application/json"
     http://localhost:3000/api/services/1
```


- HTTP Basics
- REST API Entry Point
  - Response:
  - Entry Point Details
- Supported Content Types
- URL Paths
- Methods and related URLs
- Updating resources
- Modifying Resource attributes
- Return Codes
  - Success
  - Client Errors
  - Server Errors

HTTP Basics

- REST Entry Point
  - Entry Point Details
- Supported Content Types
- URL Paths
- Methods and related URLs
- Updating resources
- Return Codes

REST API Entry Point

The REST API is made available via the /api URL prefix. It can be accessed on the Appliance server directly via HTTP using the default port as follows:

```
http://localhost:3000/api
```

Or external to the Appliance as follows:

```
https://<applianceHost_fqdn>/api
```

Response:

```
{
  "name" : "API",
  "description" : "REST API",
  "version" : "2.3.0",
  "versions" : [
    {
      "name" : "2.3.0",
      "href" : "http://localhost:3000/api/v2.3.0"
    }
  ],
  ...
  "collections" : [
    ...
    {
      "name" : "providers",
      "href" : "http://localhost:3000/api/providers",
      "description" : "Providers"
    },
    ...
    {
      "name" : "vms",
      "href" : "http://localhost:3000/api/vms",
      "description" : "Virtual Machines"
    },
    ...
  ]
}
```

- Version is the Current API version, accessible via either of the following:
- /api/
- /api/v2.0.0/
- Versions lists current and any earlier API versions that are still exposed via their respective entry points:
- /api/v**Version**/

Entry Point Details

An authenticated request of the entry point:

```
GET /api
```

returns the following structure:

```
{
  "name": "API",
  "description": "REST API",
  "version": "2.3.0",
  "versions": [
    {
      "name": "2.3.0",
      "href": "http://localhost:3000/api/v2.3.0"
    }
  ],
  "settings": {
  },
  "identity": {
    "userid": "admin",
    "name": "Administrator",
    "user_href": "http://localhost:3000/api/users/1",
    "group": "EvmGroup-super_administrator",
    "group_href": "http://localhost:3000/api/groups/2",
    "role": "EvmRole-super_administrator",
    "role_href": "http://localhost:3000/api/roles/1",
    "tenant": "My Company",
    "groups": [
      "EvmGroup-super_administrator"
    ]
  },
  "server_info": {
    "version": "master",
    "build": "20161104133757_01d7244",
    "appliance": "EVM"
  },
  "product_info": {
    "name": "ManageIQ",
    "name_full": "ManageIQ",
    "copyright": "Copyright (c) 2014-present ManageIQ Authors.",
    "support_website": "http://www.manageiq.org",
    "support_website_text": "ManageIQ.org"
  },
  "collections": [
    {
      "name": "actions",
      "href": "http://localhost:3000/api/actions",
      "description": "Actions"
    },
    ...
```

```
    {
      "name": "vms",
      "href": "http://localhost:3000/api/vms",
      "description": "Virtual Machines"
    },
    {
      "name": "zones",
      "href": "http://localhost:3000/api/zones",
      "description": "Zones"
    }
  ]
}
```

Where **settings** and **identity** are specific to the authenticated user.

Supported Content Types

Requests:

```
Accept: application/json
```

Responses:

```
Content-Type: application/json
```

URL Paths

The recommended convention for URLs is to use alternate collection / resource path segments, relative to the API entry point as described in the following example:

URL	Description
/api	The REST API entry point
/api/v**Version**	The REST entry point for a specific version of the REST API
/api/:collection	A top-level collection
/api/:collection/:id	A specific resource of that collection
/api/:collection/:id/:subcollection	Sub-collection under the specific resource

Methods and related URLs

The basic HTTP Methods used for the API are GET, POST, PUT, PATCH, OPTIONS and DELETE.

URL	Semantic
GET /api/:collection	Return all resources of the collection
GET /api/:collection/:id	Return the specific resource
POST /api/:collection	Create a resource in the collection
POST /api/:collection/:id	Perform an Action on a resource in the collection
PUT /api/:collection/:id	Update a specific resource
PATCH /api/:collection/:id	Update a specific resource
OPTIONS /api/:collection	Return metadata about the collection Please refer to Collection Metadata for details
DELETE /api/:collection/:id	Delete a specific resource

There :collection represent specific Appliance entities like services, hosts, vms, etc.

Updating resources

As shown in the above table, there are a couple of way to update attributes in a resource. These are:

- Update a resource via the PUT HTTP Method
- Update a resource via a POST Method with an *edit* action.
- Update a resource via the PATCH HTTP Method

While the PUT is the common method, the PATCH mechanism gives better control on which attribute to edit, add as well as remove which is not available with the other two methods.

Modifying Resource attributes

```
PUT /api/services/42
```

```
{
  "name" : "The new Service name",
  "description" : "A Description for the new Service"
}
```

```
POST /api/services/42
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "A new Service name",
    "description" : "A Description for the new Service"
  }
}
```

```
PATCH /api/services/42
```

```
[
  { "action": "edit", "path": "name", "value": "A new Service name" },
  { "action": "add", "path": "description", "value": "A Description for the new Service" },
  { "action": "remove", "path": "display" }
]
```

Return Codes

- Success
- Client Errors
- Server Errors

Success

- **200 OK** - The request has succeeded without errors, this code should be returned for example when retrieving a collection or a single resource.
- **201 Created** - The request has been fulfilled and resulted in a **new resource being created**. The resource is available before this status code is returned. The response includes the HTTP body of the newly created resource.
- **202 Accepted** - The request has been accepted for processing, but the processing has not been completed. Like, resource is not fully available yet. This status code is usually returned when the resource creation happens asynchronously. In this case the HTTP response includes a pointer to *monitor* or a *job* where the client can query to get the current status of the request and the estimate on when the request will be actually fulfilled.
- **204 No Content** - The server has fulfilled the request but does not need to return an entity-body, and might want to return updated meta information. This HTTP response is commonly used for the DELETE requests, as the resource that was deleted does not exists anymore.

Client Errors

- **400 Bad Request** - The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications. In REST API this status code should be returned to client when the client use the wrong combination of attributes, like expanding the non-existing collection, or using the pagination parameter incorrectly. Another use-case could be creating or performing actions on the resource, when the wrong JSON serialization of the resource or action is used.
- **401 Unauthorized** - The request requires user authentication. The response MUST include a *Authenticate* header field containing a challenge applicable to the requested resource. If the request include *Authenticate* header, then this HTTP status code might indicate that the current user is **not authorized** to perform given action or to access given resource.
- **403 Forbidden** - The server understood the request, but is refusing to fulfill it. Authorization will not help in this case. This HTTP status code might indicate that the action performed is not supported for this resource or collection.
- **404 Not Found** - In this case, the server has not found anything that matches with the URL.
- **415 Unsupported Media Type** - The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method. This error must be returned, when the client is explicitly asking for format other than JSON (application/json).

Server Errors

- **500 Internal Server Error** - The server encountered an unexpected condition which prevented it from fulfilling the request. This error code must be used when an exception is raised in the application and the exception has nothing to do with the client request.

- JSON Specification
  - General Conventions
  - Basic types
  - Common Attributes and Actions
    - Attributes
  - Actions
    - About permissions and security:
  - Collections
  - Action Specification
  - Collection actions
  - Resource actions
  - Action Responses

JSON Specification

General Conventions

The API uses JSON throughout; the Content-Type for all requests and responses is **application/json**.

As is general practice with REST, clients should not make assumptions about the server's URL space. Clients are expected to discover all URL's by navigating the API. To keep this document readable, we still mention specific URL's, generally in the form of an absolute path. Clients should not use these, or assume that the actual URL structure follows these examples, and instead use discovered URL's. Any client should start its discovery with the API entry point, here denoted with **/api**.

- Basic Types
- Common Attributes and Actions
  - Attributes
- Actions
- Collections
- Action Specification
- Collection Actions
- Resource Actions
- Action Responses

Basic types

The following are basic data types and type combinators that are used throughout:

Name	Explanation	Example serialization
Integer	Integer value	{ "memsize" : 2048 }
String	JSON string	{ "state" : "running" }
URL	Absolute URL	{ "href" : "http://SERVER/vms/1/start" }
Timestamp	Timestamp in ISO8601 format	{ "created" : "2013-12-05T08:15:30Z" }
Array[T]	Array where each entry has type T	{ "vms" : [{ "id" : "1" }, { "id" : "2" }] }
Ref[T]	A reference to a T, used to model relations, the T is a valid Resource identifier	{ "vm" : { "href" : URL } }
Collection	Array[T] where T represents a Ref[T], this might allow actions to be executed on all members as a single unit	{ "vms" : { "count" : 2, "resources" : [{ "href" : URL}, { "href" : URL }], "actions" : [] }
Struct	A structure with sub-attributes	"power_state": {"state" : "ON", "last_boot_time" : "2013-05-29T15:28Z", "state_change_time" : "2013-05-29T15:28Z"}

Common Attributes and Actions

The following describes attributes and actions that are shared by all resources and collections defined in this API.

Attributes

Attribute	Type	Description
id	String	A string identifier for the referenced resource
href	Ref(self)	A unique self reference
name	String	A human name of the resource

```
{
  "id" : "10",
  "href" : "http://localhost:3000/api/vms/10",
  "name" : "first_vm"
}
```

Actions

Action	HTTP method	Description
create	POST	Create new resource in the collection
edit	PUT/PATCH/POST	Edit attributes in resource
delete	DELETE	Delete resource

Note:

About permissions and security:

Advertising of the common actions depends purely on the role and permissions of that the current API user does have for the particular resource.

Collections

Resources can be grouped into collections. Each collection is homogeneous so that it contains only one type of resource, and is unordered. Resources can also exist outside any collection. In this case, we refer to these resources as singleton resources. Collections are themselves resources as well.

Collections can exist globally, at the top level of an API, but can also be contained inside a single resource. In the latter case, we refer to these collections as sub-collections. Sub-collections are usually used to express some kind of "contained in" relationship

Collections are serialized in JSON in the following way:

```
{
  "name" : String,
  "count": Integer,
```

```
  "subcount": Integer,
  "resources": [ ... ],
  "actions": [ ... ]
}
```

Where the **name** attribute is the collection name. The **count** attribute in a collection always denotes the total number of items in the collection, not the number of items returned. **subcount** attribute in a collection depicts the number of items returned. Then the **resources** attribute is an Array[T] where T might be a list of references to the T or, if expanded a list of resources with all attributes. The **actions** then contains an Array of actions that can be performed against the collection resources.

Action Specification

The representation of each resource will only contain an action and its URL if the current user is presently allowed to perform that action against that resource. Actions will be contained in the **actions** attribute of a resource; that attribute contains an array of action definition, where each action definition has a rel, method and a href attribute.

- **name** attribute contains the action name
- **method** attribute states the HTTP method that must be used in a client HTTP request in order to perform the given action (eg. GET, POST, PUT, DELETE)
- **href** attribute contains the absolute URL that the HTTP request should be performed against
- **form** an optional attribute that references a JSON document which describes the resource attributes that can be provided in the message body when performing this action. This description will indicate which of those attributes are mandatory and which are optional.

Collection actions

The actions performed against a collection of resources are in most cases batch operations against multiple resources. The action request must include an HTTP body with the action name and the list of resource representations that the action will be performed against.

The resource representation might include the resource attributes as they can change the way how the action is actually performed. In the example below, the first service is retired immediately, versus the second being retired at a later date with a retirement warning of 3 days.

Sample JSON request body for collection action:

```
POST /api/services
```

```
{
  "action": "request_retire",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/101" },
    { "href" : "http://localhost:3000/api/services/102",
      "date" : "10/30/2015",
      "warn" : 3
    }
  ]
}
```

Actions in collection:

```
{
  "name" : String,
  "count": Integer,
  "resources": [ ... ],
  "actions": [
    {
      "name"   : "start",
      "method" : "post",
      "href"   : URL
    },
    {
      "name"   : "stop",
      "method" : "post",
      "href"   : URL
    },
    {
      "name"   : "suspend",
      "method" : "post",
      "href"   : URL
    },
    {
      "name"    : "edit",
      "method" : "post",
      "href"   : URL
    },
    {
      "name"    : "delete",
      "method" : "post",
      "href"   : URL
    },
    {
      "name"   : "delete",
      "method" : "delete",
      "href"   : URL
    }
  ]
}
```

Resource actions

An action performed against a given resource is always described in the body of the HTTP request. The HTTP body could contain a list of resource attributes that dictate how the state of the receiving resource is to be changed once the action is performed. At minimum the JSON document in the message body must contain the name of the action to be performed.

In cases where no attributes are required to perform an action the HTTP body will contain an empty JSON document, in which case default values will be assigned to the corresponding attributes.

Sample JSON request body for resource action:

```
POST /api/services/101

{
  "action"   : "request_retire",
  "resource" : { "date" : "10/30/2015", "warn" : 5 }
}

POST /api/vms/321

{
  "action"   : "start",
  "resource" : {}
}
```

or Simply:

```
{
  "action"   : "start"
}
```

Actions in a resource:

```
{
  "id"     : String,
  "href"   : Ref(self),
  "name"   : "resource human name",
  "actions" : [
    {
      "name"   : "edit",
      "method" : "post",
      "href"   : URL
    }
  ]
}
```

Action Responses

When performing actions on resources, there are two types of responses that one is to expect.

1. For actions that operate on the resource itself like a *create* or *edit*, the response is usually the updated resource. This includes creation of Provision or Automate requests where the created /api/provision_requests and /api/automation_requests gets returned.
2. For others like a *start* or *stop* action, the response includes an action result for each targetted resource. An action result has the following construct in the response:

```
{
  "results" : [
    {
      "success" : true | false,
      "message" : String,
      "href" : Ref[resource],
      "result" : Struct,
      "task_id" : Id,
      "task_href" : Ref[task]
    },
    ...
  ]
}
```

results being an array of action results as one or more resources could be targeted in a request.

NOTE:

- success and message are always there.
- result is optional and would exist when an action results in data, i.e. policy resolve.
- href is populated for the resource being targeted by the action
- task_id and task_href are optional. They are defined when actions are run asynchronously and a task is created, i.e. http://localhost:3000/api/tasks/:id which can be monitored for action completion.

Other action specific attributes could also be returned in the the action result.

For tagging actions:

```
{
  "tag_category" : String,
```

```
  "tag_name" : String,
  "tag_href" : Ref[tags]
}
```

When executing actions on subcollections, like *policies*, *policy_profiles* and *service_templates*, the following is also provided in the action result:

```
{
  "<subcollection>_id" : Id,
  "<subcollection>_href" : Ref[subcollection]
}
```


- Query Specification
  - Control Attributes
  - Filtering
  - Expanding Collection
  - Virtual Attributes and Relationships

Query Specification

This specification identifies the controls available when querying collections.

The controls are specified in the GET URL as attribute value pairs as follows:

```
GET /api/:collection?attr1=val1&attr2=val2&...
```

Control Attributes

Category	Attribute	Semantics
Pagination		
	offset	0-based offset of first item to return
	limit	number of items to return. If 0 is specified then the remaining items are returned
	Please refer to Pagination for details on default pagination and pagination links provided by the API	
Scope		
	filter[]	One or more filters to search on. See Filtering below.
	attributes=atr1,atr2,...	Which attributes in addition to id and href to return. If not specified or *all* (default is attributes=*all*), then all attributes are returned
	expand=resources	To expand the resources returned in the collection and not just the href. See Expanding Collection below
Sorting		
	sort_by=atr1,atr2,...	By which attribute(s) to sort the result by. SQL friendly virtual attributes are also supported.
	sort_order=asc or desc	Order of the sort
	sort_options=ignore_case	Option for case insensitive sort

- The **count** attribute in a collection always denotes the total number of items in the collection, not the number of items returned.
- The **subcount** attribute in a collection denotes the number of items from the collection that were returned. For example, as a result of a paged request.

Filtering

GET requests against collections support the following query parameters to enable filtering:

- **filter**: The filter to use for querying the collection, i.e. filter[]=name='myservice%'.

```
GET /api/:collection?filter[]=name='myservice%'
```

For more information on filter[] please reference the Filtering page

Expanding Collection

While in the JSON serialization example the description says that the resource might be a list of references to these resource, using the **expand** parameter they can be expanded to return a full JSON serialization of the resource instead:

GET */api/vms*

```
{
  "name" : "vms",
  "count" : 2,
  "subcount" : 2,
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/1" },
    { "href" : "http://localhost:3000/api/vms/2" }
  ],
  "actions" : []
}
```

GET */api/vms?expand=resources*

```
{
  "name" : "vms",
```

```
  "count" : 2,
  "subcount" : 2,
  "resources" : [
    {
      "id" : "1",
      "href" : "http://localhost:3000/api/vms/1",
      "name" : "My First VM",
      ...
    },
    {
      "id" : "2",
      "href" : "http://localhost:3000/api/vms/2",
      "name" : "My Second VM",
      ...
    }
  ],
  "actions" : []
}
```

Virtual Attributes and Relationships

When querying resources, virtual attributes (not database columns) as well as relationships can be queried via the attributes parameter.

For example, while accounts and software are defined subcollections of vms

```
GET /api/vms/166?expand=software&attributes=ipaddresses,lans,storage
```

```
{
  "href": "http://localhost:3000/api/vms/166",
  "id": "166",
  "vendor": "vmware",
  "name": "Dev Nightly Appl",
  "location": "Nightly Appl 201.vmx",
  "host_id": "4",
  "created_on": "2014-11-20T19:37:28Z",
  "updated_on": "2015-03-07T04:32:33Z",
  "storage_id": "12",
  "guid": "a9409862-70ec-11e4-90c6-b8e85646e742",
  "uid_ems": "422f50c6-2ba6-1059-338d-423cc3daf5b4",
  "tools_status": "toolsNotRunning",
  "standby_action": "checkpoint",
  "power_state": "off",
  "state_changed_on": "2014-11-20T19:37:28Z",
  "connection_state": "connected",
  "memory_reserve": 0,
  "memory_reserve_expand": false,
  "memory_limit": -1,
  "memory_shares": 40960,
  "memory_shares_level": "normal",
  "raw_power_state": "poweredOff",
  ...
  "ipaddresses": [
    "192.168.100.1"
  ],
  "lans": [
    {
      "id": "8",
      "switch_id": "6",
      "name": "VM Network",
      "tag": "0",
      "created_on": "2014-11-20T19:37:23Z",
      "updated_on": "2014-11-20T19:37:23Z",
      "uid_ems": "VM Network",
      "computed_allow_promiscuous": false,
      "computed_forged_transmits": true,
      "computed_mac_changes": true
    }
  ],
  "storage": {
    "id": "12",
    "name": "StarM1-Dev",
    "store_type": "VMFS",
    "total_space": 2134061875200,
    "free_space": 385020329984,
    "created_on": "2014-11-20T19:37:22Z",
    "updated_on": "2015-03-09T13:36:05Z",
    "multiplehostaccess": 0,
    "location": "4e43dd32-c6b7543a-32bf-0010187f038c",
    "uncommitted": 845539212800,
    "ems_ref_obj": "--- !ruby/string:VimString\nstr: datastore-15624\nxsiType: :ManagedObjectReference\nvimType:
:Datastore\n",
    "directory_hierarchy_supported": true,
    "thin_provisioning_supported": true,
    "raw_disk_mappings_supported": true,
    "master": false,
    "ems_ref": "datastore-15624"
  }
  "software": [
    {
      "href": "http://localhost:3000/api/vms/320/software/1",
      "id": "1",
```

```
      "name": "OpenOffice",
      "vendor": "OpenOffice.org",
      "vm_or_template_id": "166"
    }
  ]
}
```

As another example, one can query good details on hosts:

```
GET /api/hosts/8?attributes=custom_attributes,ext_management_system,resource_pools,storages,vms,hardware
```

of course, one needs to be careful with queries like these as list of vms for a host could be quite large.

Virtual attributes can also be queried from one-to-one relationships via the dot notation as follows:

```
GET /api/hosts/8?attributes=ext_management_system.id,ext_management_system.guid,ext_management_system.name
```

```
{
  "href": "http://localhost:3000/api/hosts/8",
  "id": "8",
  "name": "test1.sample.com",
  "hostname": "test1.sample.com",
  "ipaddress": "test1.sample.com",
  "vmm_vendor": "vmware",
  "vmm_version": "5.0.0",
  "vmm_product": "ESXi",
  "vmm_buildnumber": "515841",
  ...
  "ext_management_system": {
    "name": "vcenter50",
    "guid": "e84e8c58-bdbd-11e4-8983-b8e85646e742",
    "id": "6"
  }
}
```

With attributes, database attributes, virtual attributes, and relationships can be specified together as in the following example:

```
GET /api/vms/166?attributes=name,raw_power_state,ipaddresses,storage.name
```

```
{
  "href": "http://localhost:3000/api/vms/166",
  "id": "166",
  "name": "Dev Nightly Appl",
  "raw_power_state": "poweredOff",
  "ipaddresses": [
    "192.168.253.1"
  ],
  "storage": {
    "name": "StarM1-Dev"
  }
}
```

This is helpful when specific information is needed out of resources and helps with response time when querying large number of resources as in the following example:

```
GET /api/vms?limit=1000&offset=1000&expand=resources&
    attributes=name,raw_power_state,ipaddresses,storage.name
```

- Bulk Querying

Bulk Querying

While the capability to fetch an individual resource via /api/:collection/:id is available, fetching multiple resources via a filtered search is possible but has limitation on GET url limits and is cumbersome on the filtering syntax.

The Bulk Queries provides this capability by implementing a *query* POST action on the primary collections.

```
POST /api/vms
```

```
{
  "action" : "query",
  "resources" : [
    { "guid" : "...." },
    { "guid" : "...." },
    ...
  ]
}
```

The *query* interface is flexible in that it allows resources to be specified by different relevant attributes. These include:

Identifying Attributes
href
id
guid (*if applicable*)

Some other attributes can also be used as identifying attributes as shown in the following table:

Collection	Optional Identifying Attributes

Collection	Optional Identifying Attributes
/api/groups	description
/api/roles	name
/api/tags	name
/api/users	name, userid

As with regular GETs, queries return expanded resources, but also support other query parameters as in the following example:

```
POST /api/vms?attributes=name,raw_power_state
```

```
{
  "action" : "query",
  "resources" : [
    { "id" : 10 },
    { "id" : 20 },
    { "id" : 23 },
    { "id" : 28 },
    { "id" : 30 }
  ]
}
```

Sample response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/vms/10",
      "id": "10",
      "name": "test_vm_1",
      "raw_power_state": "down"
    },
    {
      "href": "http://localhost:3000/api/vms/20",
      "id": "20",
      "name": "demo_server_A",
      "raw_power_state": "down"
    },
    {
      "href": "http://localhost:3000/api/vms/23",
      "id": "23",
      "name": "demo_server_B",
      "raw_power_state": "down"
    },
    {
      "href": "http://localhost:3000/api/vms/28",
      "id": "28",
      "name": "aab_dev_vm",
      "raw_power_state": "up"
    },
    {
      "href": "http://localhost:3000/api/vms/30",
      "id": "30",
      "name": "ipa_dev",
      "raw_power_state": "down"
    }
  ]
}
```

- Collection Filtering

Collection Filtering

When querying collections, filters can be specified by providing one or more filter[]=... parameter.

```
GET /api/:collection?filter[]=...&filter[]=...&filter[]=...
```

Syntax for a filter is as follows:

```
filter[]=attribute op value
```

By default, filters are logically *anded* together, for a logical or, prefix the attribute with *or* as follows:

```
filter[]=or attribute op value
```

Where op supports the following operators:

Operator	Description	Operands
=	Equality	Numeric or string. For String, supports % or * as wildcard NULL or nil can also be specified instead of quoted string
!=	Inequality	
<	Less Than	Numeric
<=	Less Than or Equal	Numeric
>=	Greater Than or Equal	Numeric
>	Greater Than	Numeric

When strings, value to be quoted in single or double quotes. When datetimes, only strict comparison operators are supported.

Example Query of VMs named sample*, created_on > 2019-09-01, and return name and vendor

```
GET /api/vms?expand=resources&attributes=name,vendor&filter[]=created_on>2019-09-01&filter[]=name='sample%'
```

Example Query looking for services that are retired but have an unspecified service_id

```
GET /api/services?expand=resources&attributes=retired&filter[]=service_id=nil&filter[]=retired=true
```

Example Query of VMs with power_state of on or off:

```
GET /api/vms?expand=resources&filter[]=power_state=[on,off]
```

Alternately, you can use the following more verbose syntax using OR:

```
GET /api/vms?expand=resources&filter[]=power_state=off&filter[]=or%20power_state=on
```

Example Query of VMs with more than 4 CPU sockets or more than 16,000 MB of RAM (OR):

```
GET /api/vms?expand=resources&filter[]=num_cpu>4&filter[]=or%20ram_size>16000
```

Example Query of VMs without a specific value (attribute negation):

```
GET /api/vms?expand=resources&filter[]=power_state!=unknown
```

Example Query of VMs where a field is NOT NULL (vms that are either retired or not):

```
GET /api/vms?expand=resources&filter[]=retired!=nil
GET /api/vms?expand=resources&filter[]=retired!=NULL
GET /api/vms?expand=resources&filter[]=retired!=null
```

Example Query of VMs using a boolean:

```
GET /api/vms?expand=resources&filter[]=retired=true
```

Example Query of VMs created before a specific time:

```
GET /api/vms?expand=resources&filter[]=created_on<2023-01-01
```

Attributes specified via the filter[] parameter can include virtual attributes, including one level association of the virtual attribute as follows:

The HTTP **OPTIONS** request can be used on a collection to display all of the collection specific metadata, including the available virtual attributes. For more information please refer to the Collection Metadata page.

```
GET /api/vms?attributes=name,vendor,hardware&filter[]=vendor='vmware'&filter[]=hardware.memory_mb>=8192
```

Here filtering on the virtual attribute *hardware* including the one level association *memory_mb* off hardware.

- Pagination
  - About max_results_per_page

Pagination

Pagination is enabled by default. When querying collections and the *limit* query parameter is not specified, only 1000 resources are returned by default. This is driven by the new advanced setting:

```
:api:
  :max_results_per_page: 1000
```

Example GET with offset and limit specified.

```
GET /api/vms?offset=2&limit=2&sort_by=name&expand=resources
```

```
{
  "name"      : "vms",
  "count"     : 7,
  "subcount"  : 2,
  "pages"     : 4,
  "resources" : [
    {
      "href"    : "http://localhost:3000/api/vms/7254",
      "id"      :  "7254",
      "vendor"  : "vmware",
      "name"    : "test-vm1",
      ...
    },
    {
      "href"    : "http://localhost:3000/api/vms/7257",
      "id"      : "7257",
      "vendor"  : "vmware",
      "name"    : "test-vm2",
      ...
    }
  ],
 "actions" : [
    ...
  ],
 "links" : {
```

```
   "self"     : "http://localhost:3000/api/vms?offset=2&limit=2&sort_by=name&expand=resources",
   "next"     : "http://localhost:3000/api/vms?offset=4&limit=2&sort_by=name&expand=resources",
   "previous" : "http://localhost:3000/api/vms?offset=0&limit=2&sort_by=name&expand=resources",
   "first"    : "http://localhost:3000/api/vms?offset=0&limit=2&sort_by=name&expand=resources",
   "last"     : "http://localhost:3000/api/vms?offset=6&limit=2&sort_by=name&expand=resources"
 }
}
```

With pagination enabled by default, collection queries now return the following:

- “pages” metadata which is the total number of pages for the *limit* specified (or the default 1000).
- “links” metadata section which includes the necessary URL’s for fetching the current page, next and previous pages, as well as the first and last pages.

Notes:

- When the page being returned is the first page (i.e. offset = 0), the *previous* link is not returned.
- When the page being returned is the last page, the *next* page link is not returned.
- For small result sets where *limit* is less than max_results_per_page, the *limit* will be honored.
- For large result sets where the *limit* is greater than max_results_per_page, *limit* will be ignored and the result will be limited to the first max_results_per_page resources.

About max_results_per_page

If the default value for max_results_per_page does not meet your needs, it can be changed using the Advanced Settings feature in the UI. However, it is *NOT* recommended to set it to some arbitrarily large number in an effort to circumvent the pagination feature. The pagination feature is designed to prevent excessive load on the API workers, which is easy to do with large collections such as metric rollups and event streams that may have millions of records.

Infrastructure Management API Reference

In this guide you will find a detailed reference for the Infrastructure Automation – Infrastructure Management REST API.

- Primary Collections
- Actions
- Alert Management
- Authentication Management
- Automate Model
- Automate Domains
- Automation Requests
- Automate Workspaces
- Categories and Tags
- Chargebacks, Rates and Assignments
- Cloud Volumes
- Conditions
- Configuration Script Source Management
- Conversion Hosts
- Custom Actions
- Custom Attributes
- Custom Button Management
- Event Streams
- Flavor Management
- Generic Object Management
- Group Management
- Host Management
- Instance Management
- Metric Rollups
- Network Routers
- Notifications Support
- Orchestration Support
- Physical Server Management
- Picture Management
- Policy Management
- Policy Collections
- Provider Support
- Provision Requests
- Request Support
- Reports Support
- Role Management
- Security Groups
- Service Management
- Service Queries
- Service Custom Actions
- Service Dialogs
- Service Orders

- Setting Ownership
- Settings Management
- Snapshot Management
- Tags and Tagging
- Tasks
- Tenant Management
- Transformation Mappings
- User Management
- Virtual Machine Management

- Collections

Collections

The following tables shows the currently available primary collections accessible via:

```
GET /api/:collection
```

Where :collection is one of the following:

Collection	Description
actions	Actions
alert_definition_profiles	Alert Definition Profiles
alert_definitions	Alert Definitions
alerts	Alerts
auth_key_pairs	Auth Key Pairs
authentications	Authentications
automate	Automate Model
automate_classes	Automate Classes
automate_domains	Automate Domains
automate_workspaces	Automate Workspaces
automation_requests	Automation Requests
availability_zones	Availability Zones
categories	Categories
chargebacks	Chargebacks
cloud_networks	Cloud Networks
cloud_object_store_containers	Cloud Object Store Containers
cloud_subnets	Cloud Subnets
cloud_templates	Cloud Templates
cloud_tenants	Cloud Tenants
cloud_volume_types	Cloud Volume Types
cloud_volumes	Cloud Volumes
clusters	Clusters
conditions	Conditions
configuration_profiles	Configuration Profiles
configuration_script_payloads	Configuration Script Payloads
configuration_script_sources	Configuration Script Source
configuration_scripts	Configuration Scripts
configured_systems	Configured Systems
container_groups	Container Groups
container_images	Container Images
container_nodes	Container Nodes
container_projects	Container Projects
container_templates	Container Templates
container_volumes	Container Volumes
containers	Containers
conversion_hosts	Conversion Hosts
currencies	Currencies
custom_button_sets	Custom Button Sets
custom_buttons	Custom Buttons
customization_scripts	Customization Scripts
customization_templates	Customization Templates
data_stores	Datastores
enterprises	Enterprises
event_streams	Event Streams
events	Events
features	Product Features
firmware_registries	Firmware Registries

Collection	Description
firmwares	Firmwares
flavors	Flavors
floating_ips	Floating IPs
generic_object_definitions	Generic Object Definitions
generic_objects	Generic Objects
groups	Groups
guest_devices	Guest Devices
hosts	Hosts
instances	Instances
lans	Lans
load_balancers	Load Balancers
measures	Measures
metric_rollups	Metric Rollups
network_routers	Network Routers
notifications	User's past notifications
orchestration_stacks	Orchestration Stacks
orchestration_templates	Orchestration Template
physical_chassis	Physical Chassis
physical_racks	Physical Racks
physical_servers	Physical Servers
physical_storages	Physical Storages
physical_switches	Physical Switches
pictures	Pictures
policies	Policies
policy_actions	Actions
policy_profiles	Policy Profiles
providers	Providers
provision_dialogs	Provisioning Dialogs
provision_requests	Provision Requests
pxe_images	PXE Images
pxe_menus	PXE Menus
pxe_servers	PXE Servers
rates	Chargeback Rates
regions	Regions
reports	Reports
request_tasks	Request Tasks
requests	Requests
resource_pools	Resource Pools
results	Report Results
roles	Roles
search_filters	Search Filters
security_groups	Security Groups
servers	EVM Servers
service_catalogs	Service Catalogs
service_dialogs	Service Dialogs
service_offerings	Service Offerings
service_orders	Service Orders
service_parameters_sets	Service Parameters Sets
service_requests	Service Requests
service_templates	Service Templates
services	Services
settings	Settings
switches	Switches
tags	Tags
tasks	Tasks
templates	Templates
tenant_groups	Tenant Groups
tenants	Tenants
transformation_mappings	Transformation Mappings
users	Users
vms	Virtual Machines
widgets	Miq Widgets
zones	Zones

- Actions
  - Querying Actions
  - Creating Actions
  - Editing Actions
  - Deleting Actions

Actions

Management of actions is provided via the following collection:

`/api/actions`

Full CRUD actions on resources is available:

- Querying Actions
- Creating Actions
- Editing Actions
- Deleting Actions

Querying Actions

Querying all actions in the system is simply:

`GET /api/actions`

Getting details on a specific Action:

`GET /api/actions/:id`

Creating Actions

Actions can be created via a POST to the actions collection or via the **create** action signature which also allows creation of multiple actions in a single request.

`POST /api/actions`

```
{
  "name" : "sample_action",
  "description" : "Sample Action",
  "action_type" : "custom_automation",
  "options" : {
    "ae_message" : "message",
    "ae_request" : "request",
    "ae_hash" : { "sample_key" : "sample_value" }
  }
}
```

or creating multiple actions:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "action_1", "description" : "Action 1", ... },
    { "name" : "action_2", "description" : "Action 2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating actions.

Editing Actions

`POST /api/actions/:id`

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Description"
  }
}
```

or editing multiple actions:

`POST /api/actions`

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/actions/101",
      "description" : "Updated Description 1"
    },
    {
```

```
      "href" : "http://localhost:3000/api/actions/102",
      "description" : "Updated Description 2"
    },
    ...
  ]
}
```

Deleting Actions

Actions can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/actions/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/actions/101
```

Deleting multiple actions can be done as follows:

```
POST /api/actions

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/actions/101" },
    { "href" : "http://localhost:3000/api/actions/102" },
    ...
  ]
}
```


- Alert Management
  - Querying Alerts
  - Querying Alert Actions
  - Creating Alert Actions
  - Querying Alert Definitions
  - Creating Alert Definitions
  - Editing Alert Definitions
  - Deleting Alert Definitions

Alert Management

Management of alerts, alert actions and alert definitions is provided via the following collections and subcollections:

```
/api/alerts
/api/alerts/:id/alert_actions

/api/alert_definitions
```

Queries of alerts:

- Querying Alerts

Queries and creation of alert actions:

- Querying Alert Actions
- Creating Alert Actions

Full CRUD actions on alert definitions:

- Querying Alert Definitions
- Creating Alert Definitions
- Editing Alert Definitions
- Deleting Alert Definitions

Querying Alerts

Querying all alerts in the system is simply:

```
GET /api/alerts
```

Getting details on a specific alert:

```
GET /api/alerts/:id
```

Querying Alert Actions

Querying all alert actions of a specific alert is done via the subcollection as follows:

```
GET /api/alerts/:alert_id/alert_actions
```

Querying an alert action of a specific alert is done via the subcollection resource as follows:

```
GET /api/alerts/:alert_id/alert_actions/:alert_action_id
```

Creating Alert Actions

Creating an alert action for a specific alert can be done by the following POST:

```
POST /api/alerts/:alert_id/alert_actions

{
  "action_type" : "custom_automation",
  "comment" : "Comment about the Alert Action"
}
```

Querying Alert Definitions

Querying all alert definitions in the system is simply:

```
GET /api/alert_definitions
```

Getting details on a specific alert definition:

```
GET /api/alert_definitions/:id
```

Creating Alert Definitions

Alert definitions can be created via a POST to the alert definitions collection or via the **create** action signature which also allows creation of multiple alert definitions in a single request.

```
POST /api/alert_definitions

{
  "description" : "Test Alert Definition",
  "db" : "ContainerNode",
  "expression" : { "eval_method" : "dwh_generic", "mode" : "internal", "options" : {} },
  "options"    : { "notifications" : { "delay_next_evaluation" : 600, "evm_event" : {} } },
  "enabled"    : true
}
```

or creating multiple alert definitions:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "Sample Alert Definition 1", ... },
    { "description" : "Sample Alert Definition 2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating alert definitions.

Editing Alert Definitions

```
POST /api/alert_definitions/:id

{
  "action" : "edit",
  "resource" : {
    "options" : { "notifications" : { "delay_next_evaluation" : 1200, "evm:event" : {} } }
  }
}
```

or editing multiple alert definitions:

```
POST /api/alert_definitions

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/alert_definitions/101",
      "options" : { "notifications" : { "delay_next_evaluation" : 1200, "evm:event" : {} } }
    },
    {
      "href" : "http://localhost:3000/api/alert_definitions/102",
      "options" : { "notifications" : { "delay_next_evaluation" : 1200, "evm:event" : {} } }
    },
    ...
  ]
}
```

Deleting Alert Definitions

Alert definitions can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/alert_definitions/101
```

```
{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/alert_definitions/101
```

Deleting multiple alert definitions can be done as follows:

```
POST /api/alert_definitions
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/alert_definitions/101" },
    { "href" : "http://localhost:3000/api/alert_definitions/102" },
    ...
  ]
}
```


- Authentications
  - Querying Authentications
  - Creating Authentications
  - Editing Authentications
  - Refreshing Authentications
  - Deleting Authentications

Authentications

Management of authentications is provided via the following collection:

```
/api/authentications
```

Full CRUD actions on authentications is available:

- Querying Authentications
- Creating Authentications
- Editing Authentications
- Refreshing Authentications
- Deleting Authentications

Querying Authentications

Querying all authentications in the system is simply:

```
GET /api/authentications
```

Getting details on a specific authentication:

```
GET /api/authentications/:id
```

One can also query authentications of a configuration script payload's manager as follows:

```
GET /api/configuration_script_payloads/:id/authentications
```

Or getting details on a specific authentication:

```
GET /api/configuration_script_payloads/:id/authentications/:authentication_id
```

Creating Authentications

Authentications can be created via a POST to the authentications collection or via the **create** action signature which also allows creation of multiple authentications in a single request.

```
POST /api/authentications
```

```
{
  "description" : "Authentication Description",
  "name" : "SomeCredentials",
  "related" : {},
  "type" : "ManageIQ::Providers::AnsibleTower::AutomationManager::Credential",
  "manager_resource" : { "href" : "http://localhost:3000/api/providers/7" }
}
```

or creating multiple authentications:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "System Credentials", "name" : "SystemCreds", ... },
```

```
    { "description" : "Admin Credentials",  "name" : "AdminCreds", ... },
    ...
  ]
}
```

Optionally, authentications can be created for a provider of a specific configuration script payload, thus eliminating the need of specifying a provider.

```
POST /api/configuration_script_payloads/:id/authentications
```

```
{
  "description" : "Authentication Description",
  "name" : "SomeCredentials",
  "related" : {},
  "type" : "ManageIQ::Providers::AnsibleTower::AutomationManager::Credential"
}
```

Also supporting bulk creates:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "System Credentials", "name" : "SystemCreds", ... },
    { "description" : "Admin Credentials",  "name" : "AdminCreds", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating authentications.

Editing Authentications

```
POST /api/authentications/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "UpdatedCredentials"
  }
}
```

or editing multiple authentications:

```
POST /api/authentications
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/authentications/101",
      "description" : "Updated Sample Credentials 1",
      "name" : "UpdatedCredentials1"
    },
    {
      "href" : "http://localhost:3000/api/authentications/102",
      "description" : "Updated Sample Credentials 2",
      "name" : "UpdatedCredentials2"
    },
    ...
  ]
}
```

Refreshing Authentications

Authentications can be refreshed by posting a **refresh** action to a single resource or to multiple resources in bulk by targeting the collection.

```
POST /api/authentications/:id
```

```
{
  "action" : "refresh"
}
```

or refreshing multiple authentications:

```
POST /api/authentications
```

```
{
  "action" : "refresh",
  "resources" : [
    { "id" : "51" },
    { "id" : "52" }
  ]
}
```

Example:

```
POST /api/authentications/51
```

```
{
  "action" : "refresh"
```

```
}
```

Response:

```
{
  "success" : true,
  "message" : "Refreshing Authentication id:51 name:'SampleCredentials'",
  "task_id" : "8",
  "task_href" : "http://localhost:3000/api/tasks/8",
  "tasks" : [
    {
      "id" : "8",
      "href" : "http://localhost:3000/api/tasks/8"
    }
  ]
}
```

Deleting Authentications

Authentications can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/authentications/101
```

```
{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/authentications/101
```

Deleting multiple authentications can be done as follows:

```
POST /api/authentications
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/authentications/101" },
    { "href" : "http://localhost:3000/api/authentications/102" },
    ...
  ]
}
```


- Automate
  - Usage
  - Querying Domains
  - Querying By Fully Qualified Name
  - Querying First Level Children Nodes
  - Querying Subtree
  - Querying State Machines Subtree

Automate

Querying the Automate model is provided via the following entrypoint:

```
/api/automate
```


- Usage
- Querying Domains
- Querying By Fully Qualified Name
- Querying First Level Children Nodes
- Querying Subtree
- Querying State Machines Subtree

Usage

Querying the Automate Model via /api/automate allows queries of:

- top level Automate Domains
- any object in the Automate Model
- arbitrary depth via the **depth** parameter

depth value	data returned
0	Just node itself
1	Node itself plus its first level children nodes
#	Node itself plus # levels of children nodes
-1	Node itself plus the whole subtree underneath it

- Automate model object tree to include state machine endpoints via the **search_options** parameter with value **state_machines**

Example queries:

```
GET /api/automate
GET /api/automate/manageiq
GET /api/automate/manageiq?depth=2
GET /api/automate/manageiq?depth=-1&search_options=state_machines
```

Querying Domains

Querying the Automate Domains can be done via GET on the /api/automate entrypoint as follows:

```
GET /api/automate
```

```
{
  "name": "automate",
  "subcount": 4,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain",
      "id": "2352",
      "name": "ManageIQ",
      "created_on": "2016-10-31T14:58:09Z",
      "updated_on": "2016-10-31T14:58:09Z",
      "description": "Base domain",
      "updated_by": "system",
      "priority": 0,
      "enabled": true,
      "tenant_id": "1",
      "source": "system"
    },
    ...
  ]
}
```

Note:

The fqname is always returned by this query

If desired a sub-set of attributes can be asked via as follows:

```
GET /api/automate?attributes=name,source
```

```
{
  "name": "automate",
  "subcount": 4,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "name": "ManageIQ",
      "source" : "system"
    },
    ...
  ]
}
```

Querying By Fully Qualified Name

Querying any object in the Automate Model by its fully qualified name as follows:

Note:

The fqname in the query is *not* case sensitive

```
GET /api/automate/manageiq
```

```
{
  "name": "automate",
  "subcount": 1,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain",
      "id": "2352",
      "name": "ManageIQ",
      "created_on": "2016-10-31T14:58:09Z",
      "updated_on": "2016-10-31T14:58:09Z",
      "description": "Base domain",
      "updated_by": "system",
      "priority": 0,
      "enabled": true,
      "tenant_id": "1",
      "source": "system"
    }
  ]
}
```

Another example showing fetching an automate class object:

```
GET /api/automate/manageiq/system/request
```


```
{
  "name": "automate",
  "subcount": 1,
  "resources": [
    {
      "fqname": "/ManageIQ/System/Request",
      "domain_fqname": "/System/Request",
      "klass": "MiqAeClass",
      "id": "75",
      "description": "Automation Requests",
      "name": "Request",
      "created_on": "2015-12-09T20:56:44Z",
      "updated_on": "2015-12-09T20:56:44Z",
      "namespace_id": "41",
      "updated_by": "system"
    }
  ]
}
```


Automate model nodes returned include both:

- **fqname**: the fully qualified name of the node which is rooted at /.
    - i.e. /ManageIQ/System/Request
- **domain_fqname**: the fully qualified name of the node rooted at the domain, which is **fqname** without the Domain prefix
    - i.e. /System/Request

Querying First Level Children Nodes

When specifying a depth of 1, the Automate model node and its first level children nodes are returned.

In this example, we are asking for the /ManageIQ Domain and its direct children nodes.

```
GET /api/automate/manageiq?depth=1&attributes=domain_fqname,klass
```


```
{
  "name": "automate",
  "subcount": 9,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain"
    },
    {
      "fqname": "/ManageIQ/System",
      "domain_fqname": "/System",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Service",
      "domain_fqname": "/Service",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Infrastructure",
      "domain_fqname": "/Infrastructure",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/GenericObject",
      "domain_fqname": "/GenericObject",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Deployment",
      "domain_fqname": "/Deployment",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Control",
      "domain_fqname": "/Control",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/ConfigurationManagement",
      "domain_fqname": "/ConfigurationManagement",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Cloud",
      "domain_fqname": "/Cloud",
      "klass": "MiqAeNamespace"
    }
```

```
  ]
}
```

Querying Subtree

When specifying a depth of -1, the Automate model node and all of the nodes underneath it are returned.

```
GET /api/automate/manageiq?depth=-1&attributes=klass
```

```
{
  "name": "automate",
  "subcount": 935,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "klass": "MiqAeDomain"
    },
    {
      "fqname": "/ManageIQ/System",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/System/Event",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/System/Event/RequestEvent",
      "klass": "MiqAeNamespace"
    },
    ...
  ]
}
```

Querying State Machines Subtree

In this example, we are querying the ManageIQ Automate Domain model tree but only asking for branches that include state machine nodes.

```
GET /api/automate/manageiq?depth=-1&attributes=domain_fqname,klass&search_options=state_machines
```

```
{
  "name": "automate",
  "subcount": 104,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain"
    },
    {
      "fqname": "/ManageIQ/System",
      "domain_fqname": "/System",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/System/CommonMethods",
      "domain_fqname": "/System/CommonMethods",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/System/CommonMethods/QuotaStateMachine",
      "domain_fqname": "/System/CommonMethods/QuotaStateMachine",
      "klass": "MiqAeClass"
    },
    {
      "fqname": "/ManageIQ/System/CommonMethods/QuotaStateMachine/quota",
      "domain_fqname": "/System/CommonMethods/QuotaStateMachine/quota",
      "klass": "MiqAeInstance"
    },
    ...
  ]
}
```


- Automate Domains
  - Querying Domains
  - Accessing Domains
  - Refresh From Source

Automate Domains

Querying the Automate Domains and triggering actions on domains is provided via the following entrypoint:

```
/api/automate_domains
```


- Querying Domains
- Accessing Domains
- Refresh From Source

Querying Domains

Querying the Automate Domains can be done via GET on the /api/automate_domains entrypoint and can be queried like any other collection:

```
GET /api/automate_domains
```

```
{
  "name": "automate_domains",
  "count": 9,
  "subcount": 9,
  "resources": [
    {
      "href": "http://localhost:3000/api/automate_domains/2352"
    },
    {
      "href": "http://localhost:3000/api/automate_domains/2347"
    },
    ...
  ],
  "actions": [
    {
      "name": "query",
      "method": "post",
      "href": "http://localhost:3000/api/automate_domains"
    },
    {
      "name": "refresh_from_source",
      "method": "post",
      "href": "http://localhost:3000/api/automate_domains"
    }
  ]
}
```

Accessing Domains

Accessing individual domains can be done via their numeric id or their name (*case sensitive*) as follows:

```
GET /api/automate_domains/2352
```

```
GET /api/automate_domains/ManageIQ
```

Refresh From Source

When Automate domains are imported from external sources, i.e. Git, the *refresh_from_source* will re-fetch the Automate domain from external source and refresh the automate domain. This will refetch the Domain from the same reference, i.e. branch or tag and name.

```
POST /api/automate_domains/SampleDomain
```

```
{
  "action": "refresh_from_source"
}
```

All refresh_from_source actions are done asynchronously and the related Task id and href are returned as follows:

```
{
  "success": true,
  "message": "Refreshing Automate Domain id:2347 name:'SampleDomain' from git repository",
  "task_id": "61",
  "task_href": "http://localhost:3000/api/tasks/61",
  "href": "http://localhost:3000/api/automate_domains/SampleDomain"
}
```

If the Domain needs to be refreshed from a different branch or tag and name, the following can be passed in with the *refresh_from_source* action:

```
{
  "action": "refresh_from_source",
  "resource" : {
    "ref_type" : "branch" | "tag",
    "ref" : "other_branch_or_tag_name"
  }
}
```


- Automation Requests
  - Creating Requests
  - Editing Requests
  - Approving or Denying Requests
    - Approving a Request
    - Denying a Request
  - Monitoring Requests
    - Response:
  - Editing Request Tasks

Automation Requests

Automation requests are made available via the following entrypoint:

```
/api/automation_requests
```

- Creating Requests
- Editing Requests
- Approving or Denying Requests
- Monitoring Requests
- Editing Request Tasks

Creating Requests

Either by specifying a *create* action or by posting the automation request directly to **/api/automation_requests**.

As an example:

```
{
  "uri_parts" : {
    "namespace" : "System",
    "class"     : "Request",
    "instance"  : "InspectME",
    "message"   : "create"
  },
  "parameters" : {
    "var1" : "value 1",
    "var2" : "value 2",
    "minimum_memory" : 2048
  },
  "requester" : {
    "auto_approve" : true
  }
}
```

Editing Requests

Updating requests can be performed by posting an *edit* action as follows:

```
POST /api/automation_requests/11
```

```
{
  "action" : "edit",
  "resource" : {
    "options" : {
      "attrs" : {
        "var1" : "updated value1",
        "var2" : "updated value2",
        "minimum_memory" : 4096
      }
    }
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/automation_requests/11",
  "id": "11",
  "options": {
    "message": "create",
    "namespace": "System",
    "class_name": "Request",
    "instance_name": "InspectME",
    "user_id": "1",
    "attrs": {
      "var1" : "updated value1",
      "var2" : "updated value2",
      "minimum_memory" : 4096
    }
  },
  "tenant_id": "1",
  "approval_state": "approved",
  "status": "Ok",
  "request_type": "automation",
  "description": "Automation Task",
  "type": "AutomationRequest",
  "created_on": "2018-01-08T13:48:00Z",
  "updated_on": "2018-01-08T13:56:15Z",
  "requester_id": "1",
  "requester_name": "Administrator",
  "request_state": "pending",
  "message": "Automation Request - Request Created",
  "userid": "admin",
  "process": true
}
```

Approving or Denying Requests

If an automation request is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the automation request resource.

Approving a Request

```
POST /api/automation_requests/:id

{
  "action" : "approve",
  "reason" : "reason for approval"
}
```

Denying a Request

```
POST /api/automation_requests/:id

{
  "action" : "deny",
  "reason" : "reason for denial"
}
```

Monitoring Requests

Once an automation request is created, the response will be the queryable automation request itself, i.e. **/api/automation_request/:id**

Response:

```
{
  "results": [
    {
      "id": "10",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-14T18:12:12Z",
      "updated_on": "2015-04-14T18:12:12Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",
        "instance_name": "InspectME",
        "user_id": "1",
        "attrs": {
          "var1": "value 1",
          "var2": "value 2",
          "minimum_memory": 2048,
          "userid": "admin"
        }
      },
      "userid": "admin"
    }
  ]
}
```

In the above example, the request could be queried periodically until the **request_state** reaches the **finished** state.

Note:

The request tasks of an automation request can also be queried by expanding the request_tasks subcollection as follows:

```
GET /api/automation_requests/:id?expand=request_tasks
```

or by fetching them directly as follows:

```
GET /api/automation_requests/:id/request_tasks?expand=resources
```

Editing Request Tasks

Request tasks for Automation Requests can be updated as follows:

```
POST /api/automation_requests/:id/request_tasks/:request_task_id

{
  "action" : "edit",
  "resource" : {
    "options" : {
      "request_param_a" : "value_a",
      "request_param_b" : "value_b"
    }
  }
}
```


- Automate Workspaces
  - Querying Automate Workspaces
    - Request:
    - Response:

- Editing Automate Workspaces
- Encrypting Automate Workspace Passwords
  - Request:
  - Response:
- Decrypting Automate Workspace Passwords
  - Request:
  - Response:

Automate Workspaces

Management of Automate Workspaces is provided with the following collection:

`/api/automate_workspaces`

- Querying Automate Workspaces
- Editing Automate Workspaces
- Encrypting Automate Workspace Passwords
- Decrypting Automate Workspace Passwords

Querying Automate Workspaces

It is not allowed to query the automate workspace collection, only individual automate workspaces can be fetched by guid.

`GET /api/automate_workspaces/:guid`

Example GET Request and Response:

Request:

`GET /api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f`

Response:

```
{
  "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f",
  "id": "2",
  "guid": "430a66a0-b564-4341-b9b2-85d6f467f26f",
  "user_id": "1",
  "tenant_id": "1",
  "input": {
    "objects": {
      "root": {
        "var1": "a",
        "var2": "b"
      }
    },
    "method_parameters": {
      "arg1": "first_argument"
    }
  },
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f"
    },
    {
      "name": "encrypt",
      "method": "post",
      "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f"
    },
    {
      "name": "decrypt",
      "method": "post",
      "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f"
    }
  ]
}
```

Editing Automate Workspaces

Updating Automate Workspaces can be done via the *edit* action as follows:

`POST /api/automate_workspaces/:guid`

```
{
  "action" : "edit",
  "resource" : {
    "objects" : {
      "root" : {
        "var3" : "c"
      }
    }
```

```
  }
}
```

Encrypting Automate Workspace Passwords

Encrypting passwords in the automate workspace can be done as follows:

Request:

```
POST /api/automate_workspaces/:guid

{
  "action" : "encrypt",
  "resource" : {
    "object" : "method_parameters",
    "attribute" : "arg2",
    "value" : "unencrypted_password"
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f",
  "id": "2",
  "guid": "430a66a0-b564-4341-b9b2-85d6f467f26f",
  "user_id": "1",
  "tenant_id": "1",
  "input": {
    "objects": {
      "root": {
        "var1": "a",
        "var2": "b"
      }
    },
    "method_parameters": {
      "arg1": "first_argument",
      "arg2": "password::********"
    }
  }
}
```

Decrypting Automate Workspace Passwords

Encrypted passwords in the automate workspace can be decrypted individually as follows:

Request:

```
POST /api/automate_workspaces/:guid

{
  "action" : "decrypt",
  "resource" : {
    "object" : "method_parameters",
    "attribute" : "arg2"
  }
}
```

Response:

```
{
  "object" : "method_parameters",
  "attribtue" : "arg2",
  "value" : "unencrypted_password"
}
```


- Categories & Tags
  - Querying Categories
  - Querying Tags
    - Request:
    - Response:
    - Request:
    - Response:
  - Creating Categories
    - Response:
  - Creating Tags
  - Editing Categories & Tags
  - Deleting Categories & Tags
    - Request:
    - Response:

Categories & Tags

Management of Categories and Tags is provided via the following collections:

`/api/categories`

`/api/tags`

Full CRUD actions on both cateogries and tags are made available.

- Querying Categories
- Querying Tags
- Creating Categories
- Creating Tags
- Editing Categories & Tags
- Deleting Categories & Tags

Querying Categories

Querying all categories in the system is simply:

```
GET /api/categories
```

Or getting attributes of interest like *name*, *description* or *ns* for the namespace:

```
GET /api/categories?expand=resources&attributes=ns,name,description
```

Querying Tags

Querying all tags in the system is simply:

```
GET /api/tags
```

Querying tags of a specific category is done as follows:

```
GET /api/categories/:id/tags
```

Example Query for Location Category:

Request:

```
GET /api/categories/1
```

Response:

```
{
  "href": "http://localhost:3000/api/categories/1",
  "id": "1",
  "description": "Location",
  "read_only": false,
  "syntax": "string",
  "single_value": true,
  "example_text": "The geographic location of the resource, such as New York, Chicago, or London.",
  "tag_id": "4",
  "parent_id": "0",
  "show": true,
  "default": true,
  "name": "location",
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/categories/1"
    }
  ]
}
```

Querying all tags of the Location category:

Request:

```
GET /api/categories/1/tags?expand=resources&attributes=name
```

Response:

```
{
  "name": "tags",
  "count": 1389,
  "subcount": 4,
  "resources": [
    {
      "href": "http://localhost:3000/api/categories/1/tags/8",
      "id": "8",
      "name": "/managed/location/paris"
    },
    {
      "href": "http://localhost:3000/api/categories/1/tags/7",
      "id": "7",
      "name": "/managed/location/london"
    },
    {
      "href": "http://localhost:3000/api/categories/1/tags/6",
      "id": "6",
      "name": "/managed/location/chicago"
    },
    {
      "href": "http://localhost:3000/api/categories/1/tags/5",
      "id": "5",
      "name": "/managed/location/ny"
    }
  ],
  "actions": [
    {
      "name": "create",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1/tags"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1/tags"
    }
  ]
}
```

For additional details on tags, the **category** and **classification** attributes can be queried on tags as follows:

```
GET /api/tags/5?attributes=category,classification
```

Additionally, a compound attribute called **categorization** is provided to return key information from both **category** and **classification** of tags as follows:

```
GET /api/tags/5?attributes=categorization
```

```
{
  "href": "http://localhost:3000/api/tags/5",
  "id": "5",
  "name": "/managed/location/ny",
  "categorization": {
    "name": "ny",
    "description": "New York",
    "category": {
      "name": "location",
      "description": "Location"
    },
    "display_name": "Location: New York"
  }
}
```

Creating Categories

Creating Categories can be done by posting the Category JSON to the categories collection directly, or via the **create** action signature as follows:

```
POST /api/categories
```

```
{
  "name" : "test",
  "description" : "Test Category"
}
```

or via the **create** action:

```
{
  "action" : "create",
  "resource" : {
    "name" : "test",
    "description" : "Test Category"
  }
}
```

Response:

```
{
  "results": [
    {
      "id": "1430",
      "description": "Test Category",
```

```
      "read_only": false,
      "syntax": "string",
      "single_value": false,
      "tag_id": "1439",
      "parent_id": "0",
      "show": true
    }
  ]
}
```

Attributes which can also be specified when creating categories include:

example_text
show
single_value
syntax

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Categories.

Creating Tags

New tags for a category can be created either as a tag collection post or via a tag subcollection **create** action to a specific category.

```
POST /api/tags

{
  "name" : "test_tag",
  "description" : "Test Tag",
  "category" : { "href" : "http://localhost:3000/api/categories/1430" }
}
```

The **category** above can identify a category by specifying either one of the **href**, **id** or **name** attributes.

Tags can optionally be created via the **create** action on the tags subcollection of a category as follows:

```
POST /api/categories/1430/tags

{
  "action" : "create",
  "resource" : {
    "name" : "test_tag",
    "description" : "Test Tag"
  }
}
```

or simply:

```
POST /api/categories/1430/tags

{
  "name" : "test_tag",
  "description" : "Test Tag"
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Tags.

Editing Categories & Tags

Editing Categories and Tags can be done via the **edit** action:

Example editing a category:

```
POST /api/categories/1430

{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Category Description"
  }
}
```

Example editing a tag:

```
POST /api/tags/1441

{
  "action" : "edit",
  "resource" : {
    "name" : "updated_test_tag"
  }
}
```

Deleting Categories & Tags

Deleting Categories and Tags can be done via either the **delete** post action or the DELETE HTTP method.

```
POST /api/categories/1430

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/categories/1430
```

Deleting associated tag example:

```
POST /api/tags/1441

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/tags/1441
```

One can also delete the tag when accessed via the subcollection

Request:

```
POST /api/categories/1430/tags

{
  "action" : "delete",
  "resources" : [
    { "id" : "1441" },
    { "id" : "1442" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "tags id: 1441 deleting"
    },
    {
      "success": true,
      "message": "tags id: 1442 deleting"
    }
  ]
}
```

Tags to be deleted can be specified via the **href**, **id** or **name** attribute.

- Chargebacks, Rates & Assignments
  - Querying Chargebacks
    - Request:
    - Response:
    - Request:
    - Response:
  - Creating Chargeback Rates
    - Request:
    - Response:
  - Updating Chargeback Rates
    - Request:
    - Response:
  - Deleting Chargeback Rates
    - Request:
    - Response:
  - Querying Assigned Chargeback Rates
    - Assigned chargeback rates on single rate:
    - Response:
    - Assigned chargeback rates on collection:
    - Response:
  - Assign action for Chargeback Rates
  - “Compute” Rate Type
  - “Storage” Rate Type
    - Response:
  - Basic Template for single chargeback request
  - Basic Template for chargebacks collection request
  - Enterprise

- Selected Providers
- Selected Clusters
- Tagged VMs and Instances
- Tagged Container Images
- Labeled Container Images
- Tenants
- Enterprise
- Selected Datastores
- Tagged Datastores
- Tenants
- Unassign action for Chargeback Rates
- Example of Chargeback Flow
- Create Chargeback Report
  - Chargeback Report Options
- Filtering according to resource
  - Chargeback for VMs (db=ChargebackVm)
  - Chargeback for Projects (db=ChargebackContainerProject)
  - Chargeback for Container Images (db=ChargebackContainerImage)
- Grouping Report Options
  - Chargeback for VMs (db=ChargebackVm)
  - Chargeback for Projects (db=ChargebackContainerProject)
  - Chargeback for Container Images (db=ChargebackContainerImage)
  - Example of request to create chargeback report
- Create Chargeback Rates
- Create Chargebacks (top-level rate entity)
- Assign Chargeback Rate
- Run Chargeback Report
- Results of Chargeback Report

Chargebacks, Rates & Assignments

Queries of Chargebacks and Management of Rates is provided via the following collections:

`/api/chargebacks`

`/api/rates`

- Querying Chargebacks
- Creating Chargeback Rates
- Updating Chargeback Rates
- Deleting Chargeback Rates
- Querying Assigned Chargeback Rates
- Assign action for Chargeback Rates
- Unassign action for Chargeback Rates
- Example of Chargeback Flow

Querying Chargebacks

Querying all chargebacks in the system is simply:

```
GET /api/chargebacks
```

or Query a specific chargeback, for example:

```
GET /api/chargebacks/1
```

```
{
  "href": "http://localhost:3000/api/chargebacks/1",
  "id": "1",
  "guid": "b47a0ef0-4335-11df-aba8-001d09066d98",
  "description": "Default",
  "rate_type": "Compute",
  "created_on": "2014-11-20T19:10:03Z",
  "updated_on": "2014-11-20T19:10:03Z",
  "default": true
}
```

Optionally, one can query the rates details for a chargeback as follows:

```
GET /api/chargebacks/:id/rates
```

Example getting additional rates details(attributes)

Request:

```
GET /api/chargebacks/1/rates?expand=resources&attributes=description,metric,group,source,friendly_rate
```

Response:


```
{
  "name": "rates",
  "count": 10,
  "subcount": 8,
  "resources": [
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/8",
      "id": "8",
      "description": "Fixed Compute Cost 2",
      "group": "fixed",
      "source": "compute_2",
      "friendly_rate": "0.0 Hourly"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/7",
      "id": "7",
      "description": "Fixed Compute Cost 1",
      "group": "fixed",
      "source": "compute_1",
      "friendly_rate": "1.0 Hourly"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/6",
      "id": "6",
      "description": "Used Disk I/O in KBps",
      "group": "disk_io",
      "metric": "disk_usage_rate_average",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 1.0 per Kbps from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/5",
      "id": "5",
      "description": "Used Network I/O in KBps",
      "group": "net_io",
      "metric": "net_usage_rate_average",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 0.0 per Kbps from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/4",
      "id": "4",
      "description": "Allocated Memory in MB",
      "group": "memory",
      "metric": "derived_memory_available",
      "source": "allocated",
      "friendly_rate": "Hourly @ 0.0 + 1.0 per Megabytes from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/3",
      "id": "3",
      "description": "Used Memory in MB",
      "group": "memory",
      "metric": "derived_memory_used",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 0.0 per Megabytes from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/2",
      "id": "2",
      "description": "Allocated CPU Count",
      "group": "cpu",
      "metric": "derived_vm_numvcpus",
      "source": "allocated",
      "friendly_rate": "Hourly @ 0.0 + 1.0 per Cpu from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/1",
      "id": "1",
      "description": "Used CPU in MHz",
      "group": "cpu",
      "metric": "cpu_usagemhz_rate_average",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 0.0 per Megahertz from 0.0 to Infinity"
    }
  ]
}
```


Example getting additional of rates details(relations)

Request:

```
GET /api/rates?
expand=resources&attributes=chargeback_rate,detail_measure,detail_currency,chargeable_field,chargeback_tiers
```

Response:


```
{
    "name": "rates",
    "count": 138,
    "subcount": 138,
    "pages": 1,
    "resources": [
        {
            "href": "http://localhost:3090/api/rates/544",
            "id": "544",
            "enabled": true,
            "description": "Allocated CPU Count",
            "group": "cpu",
            "source": null,
            "metric": "derived_vm_numvcpus",
            "per_time": "hourly",
            "per_unit": "cpu",
            "friendly_rate": "Hourly @ 1.0 + 0.0 per Cpu from 0.0 to Infinity",
            "chargeback_rate_id": "8",
            "created_on": "2018-06-01T11:24:52Z",
            "updated_on": "2018-06-01T11:24:52Z",
            "chargeback_rate_detail_measure_id": null,
            "chargeback_rate_detail_currency_id": "1",
            "chargeable_field_id": "3",
            "sub_metric": null,
            "chargeback_rate": {
                "id": "8",
                "guid": "52a70f67-7ab8-47a0-ba36-7808ac2449e0",
                "description": "solinux1",
                "rate_type": "Compute",
                "created_on": "2018-06-01T11:24:52Z",
                "updated_on": "2018-06-01T11:24:52Z",
                "default": false
            },
            "detail_currency": {
                "id": "1",
                "code": "USD",
                "name": "United States Dollar",
                "full_name": "United States Dollar",
                "symbol": "$",
                "unicode_hex": "36",
                "created_at": "2019-06-13T18:10:32Z",
                "updated_at": "2019-07-09T13:42:36Z"
            },
            "chargeable_field": {
                "id": "3",
                "chargeback_rate_detail_measure_id": null,
                "metric": "derived_vm_numvcpus",
                "group": "cpu",
                "source": "allocated",
                "description": "Allocated CPU Count"
            },
            "chargeback_tiers": [
                {
                    "id": "11010",
                    "chargeback_rate_detail_id": "544",
                    "start": 0.0,
                    "finish": null,
                    "fixed_rate": 1.0,
                    "variable_rate": 0.0
                }
            ],
            "actions": [
                {
                    "name": "edit",
                    "method": "post",
                    "href": "http://localhost:3090/api/rates/544"
                },
                {
                    "name": "edit",
                    "method": "patch",
                    "href": "http://localhost:3090/api/rates/544"
                },
                {
                    "name": "edit",
                    "method": "put",
                    "href": "http://localhost:3090/api/rates/544"
                },
                {
                    "name": "delete",
                    "method": "post",
                    "href": "http://localhost:3090/api/rates/544"
                },
                {
                    "name": "delete",
                    "method": "delete",
```

```
                    "href": "http://localhost:3090/api/rates/544"
                }
            ]
        },
        {
            "href": "http://localhost:3090/api/rates/510",
            "id": "510",
            "enabled": true,
            "description": "Allocated CPU Count",
            "group": "cpu",
            "source": null,
            "metric": null,
            "per_time": "hourly",
            "per_unit": "cpu",
            "friendly_rate": "Hourly @ 1.0 + 0.0 per Cpu from 0.0 to Infinity",
            "chargeback_rate_id": "7",
            "created_on": "2018-03-28T10:00:13Z",
            "updated_on": "2018-03-28T10:00:13Z",
            "chargeback_rate_detail_measure_id": null,
            "chargeback_rate_detail_currency_id": "3",
            "chargeable_field_id": "3",
            "sub_metric": null,
            "chargeback_rate": {
                "id": "7",
                "guid": "a72fd416-26ec-4aa0-bc94-0b741379dfcd",
                "description": "teamA_compute_all",
                "rate_type": "Compute",
                "created_on": "2018-03-28T10:00:13Z",
                "updated_on": "2018-03-28T10:00:13Z",
                "default": false
            },
            "detail_currency": {
                "id": "3",
                "code": "GBP",
                "name": "British Pound",
                "full_name": "British Pound",
                "symbol": "£",
                "unicode_hex": "163",
                "created_at": "2019-06-13T18:10:32Z",
                "updated_at": "2019-07-09T13:42:35Z"
            },
            "chargeable_field": {
                "id": "3",
                "chargeback_rate_detail_measure_id": null,
                "metric": "derived_vm_numvcpus",
                "group": "cpu",
                "source": "allocated",
                "description": "Allocated CPU Count"
            },
            "chargeback_tiers": [
                {
                    "id": "10115",
                    "chargeback_rate_detail_id": "510",
                    "start": 0.0,
                    "finish": 'infinity',
                    "fixed_rate": 1.0,
                    "variable_rate": 0.0
                }
            ]
        },
        {
            "href": "http://localhost:3090/api/rates/1859",
            "id": "1859",
            "enabled": true,
            "description": "My CPU allocation rate",
            "group": "cpu",
            "source": "allocated",
            "metric": null,
            "per_time": "daily",
            "per_unit": "megahertz",
            "friendly_rate": "Daily @ 2.0 + 0.0 per Megahertz from 0.0 to 1.0\nDaily @ 2.0 + 0.0 per Megahertz from 1.0 to
Infinity",
            "chargeback_rate_id": "1",
            "created_on": "2020-05-26T10:34:39Z",
            "updated_on": "2020-05-26T10:34:39Z",
            "chargeback_rate_detail_measure_id": "1",
            "chargeback_rate_detail_currency_id": "5",
            "chargeable_field_id": "3",
            "sub_metric": null,
            "chargeback_rate": {
                "id": "1",
                "guid": "b47a0ef0-4335-11df-aba8-001d09066d98",
                "description": "Default",
                "rate_type": "Compute",
                "created_on": "2020-04-23T14:43:50Z",
                "updated_on": "2020-04-23T15:00:29Z",
                "default": true
            },
            "detail_measure": {
                "id": "1",
                "name": "Bytes Units",
```

```
                "units": [
                    "bytes",
                    "kilobytes",
                    "megabytes",
                    "gigabytes",
                    "terabytes"
                ],
                "units_display": [
                    "B",
                    "KB",
                    "MB",
                    "GB",
                    "TB"
                ],
                "step": 1024,
                "created_at": "2020-04-23T14:43:50Z",
                "updated_at": "2020-04-23T15:00:28Z"
            },
            "detail_currency": {
                "id": "5",
                "code": "AUD",
                "name": "Australian Dollar",
                "full_name": "Australian Dollar",
                "symbol": "$",
                "unicode_hex": null,
                "created_at": "2019-06-13T18:10:32Z",
                "updated_at": "2019-07-09T13:42:35Z"
            },
            "chargeable_field": {
                "id": "3",
                "chargeback_rate_detail_measure_id": null,
                "metric": "derived_vm_numvcpus",
                "group": "cpu",
                "source": "allocated",
                "description": "Allocated CPU Count"
            },
            "chargeback_tiers": [
                {
                    "id": "13114",
                    "chargeback_rate_detail_id": "1859",
                    "start": 0.0,
                    "finish": 1.0,
                    "fixed_rate": 2.0,
                    "variable_rate": 0.0
                },
                {
                    "id": "13115",
                    "chargeback_rate_detail_id": "1859",
                    "start": 1.0,
                    "finish": 'infinity',
                    "fixed_rate": 2.0,
                    "variable_rate": 0.0
                }
            ]
        },
        ...
```

Creating Chargeback Rates

Creating a Chargeback Rate is done by posting a new resource or *create* action to the rates collection.

Example follows:

Request:

```
POST /api/rates
```

```
{
  "per_time" : "daily",
  "chargeback_rate_id" : "1",
  "chargeable_field_id" : "1",
  "description": "My CPU allocation rate",
  "group" : "cpu",
  "per_unit" : "megahertz",
  "source" : "allocated",
  "chargeback_tiers": [
       {
          "start": 0.0,
          "finish": "infinity",
          "fixed_rate": 2.0,
          "variable_rate": 0.0
        }
   ]
}
```

Response:

```
{
    "results": [
        {
```

```
            "href": "http://localhost:3090/api/rates/1866",
            "id": "1866",
            "enabled": true,
            "description": "My CPU allocation rate",
            "group": "cpu",
            "source": "allocated",
            "metric": null,
            "per_time": "daily",
            "per_unit": "megahertz",
            "friendly_rate": "Daily @ 2.0 + 0.0 per MHz from 0.0 to Infinity",
            "chargeback_rate_id": "1",
            "created_on": "2020-05-28T13:28:15Z",
            "updated_on": "2020-05-28T13:28:15Z",
            "chargeback_rate_detail_measure_id": null,
            "chargeback_rate_detail_currency_id": null,
            "chargeable_field_id": "1",
            "sub_metric": null
        }
    ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Chargeback Rates.

Updating Chargeback Rates

Updating rates can be done by posting **edit** actions on the rates resource.

Request:

```
POST /api/rates/16

{
  "action" : "edit",
  "resource" : { "description" : "Allocated NICs", "per_time": "hourly" }
}
```

Response:

```
{
    "href": "http://localhost:3090/api/rates/16",
    "id": "16",
    "description": "Allocated NICs",
    "per_time": "hourly",
    "chargeback_rate_id": "1",
    "chargeable_field_id": "3",
    "enabled": true,
    "group": "net_io",
    "source": null,
    "metric": null,
    "per_unit": "cpu",
    "friendly_rate": "Hourly @ 2.0 + 0.0 per Cpu from 0.0 to Infinity",
    "created_on": "2020-04-23T15:00:29Z",
    "updated_on": "2020-05-28T13:24:33Z",
    "chargeback_rate_detail_measure_id": "1",
    "chargeback_rate_detail_currency_id": "5",
    "sub_metric": null
}
```

Deleting Chargeback Rates

Deleting Chargeback Rates can be done via either the **delete** post action or the DELETE HTTP method.

Request:

```
POST /api/rates/16

{
  "action" : "delete"
}
```

Response:

```
{
  "success": true,
  "message": "rates id: 16 deleting",
  "href": "http://localhost:3000/api/rates/16"
}
```

or simply:

```
DELETE /api/rates/16
```

Querying Assigned Chargeback Rates

Assigned chargeback rates on single rate:

```
GET /api/chargebacks/16?attributes=assigned_to
```

Response:

```
{
    "href": "http://localhost:3090/api/chargebacks/2",
    "id": "2",
    "guid": "7d7aaf20-5214-11df-a888-001d09066d98",
    "description": "Default",
    "rate_type": "Storage",
    "created_on": "2019-08-07T16:48:04Z",
    "updated_on": "2020-04-20T13:27:07Z",
    "default": true,
    "assigned_to": [
        {
            "tag": {
                "href": "http://localhost:3090/api/tags/48",
                "name": "finance",
                "category": "department"
            }
        },
        {
            "tag": {
                "href": "http://localhost:3090/api/tags/26",
                "name": "test",
                "category": "environment"
            }
        }
    ]
}
```

Assigned chargeback rates on collection:

```
GET /api/chargebacks?expand=resources&attributes=assigned_to
```

Response:

```
{
    "name": "chargebacks",
    "count": 10,
    "subcount": 2,
    "subquery_count": 2,
    "pages": 1,
    "resources": [
        {
            "href": "http://localhost:3090/api/chargebacks/1",
            "id": "1",
            "guid": "b47a0ef0-4335-11df-aba8-001d09066d98",
            "description": "Default",
            "rate_type": "Compute",
            "created_on": "2019-08-07T16:48:04Z",
            "updated_on": "2020-05-13T09:36:21Z",
            "default": true,
            "assigned_to": [
                {
                    "object": {
                        "id": "1",
                        "name": "Enterprise",
                        "description": "Enterprise Root Object",
                        "created_on": "2017-10-06T16:35:32Z",
                        "updated_on": "2017-10-06T16:35:32Z",
                        "settings": null
                    }
                }
            ]
        },
        {
            "href": "http://localhost:3090/api/chargebacks/2",
            "id": "2",
            "guid": "7d7aaf20-5214-11df-a888-001d09066d98",
            "description": "Default",
            "rate_type": "Storage",
            "created_on": "2019-08-07T16:48:04Z",
            "updated_on": "2020-05-13T09:36:21Z",
            "default": true,
            "assigned_to": [
                {
                    "object": {
                        "id": "31",
                        "type": "ManageIQ::Providers::Vmware::InfraManager::Storage",
                        ...

                    }
                },
                {
                    "object": {
                        "id": "19",
                        "type": "ManageIQ::Providers::Vmware::InfraManager::Storage"
                         ...
```

```
                    }
                },
                {
                    "object": {
                        "id": "27",
                        "type": "ManageIQ::Providers::Vmware::InfraManager::Storage"
                         ...
                    }
                },
                {
                    "object": {
                        "id": "14",
                        "type": "ManageIQ::Providers::Vmware::InfraManager::Storage"
                        ...
                    }
                }
            ]
        }
    ]
}
```

Assign action for Chargeback Rates

Assignments could be done for two rate types: “Compute” and “Storage”. There are few assignments type under each rates type:

“Compute” Rate Type

- Enterprise
- Selected Providers
- Selected Clusters
- Tagged VMs and Instances
- Tagged Container Images
- Labeled Container Images
- Tenants

“Storage” Rate Type

- Enterprise
- Selected Datastores
- Tagged Datastores
- Tenants

Example (chargebacks collection) - assign storage to chargeback rate with id 2 :

```
{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback":
            {
                "id" : 2
            },
            "resource":
            {
                "href": "http://localhost:3090/api/data_stores/12"
            }
        }
    ]
}
```

Response:

```
{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/data_stores/12",
                "name": "EXPORT"
            }
        }
    ]
}
```

Basic Template for single chargeback request

```
POST /api/chargebacks/<chargeback_id>

{
    "action" : "assign",
    "assignments" : [
        {
            <resource>
        }
    ]
}
```

Basic Template for chargebacks collection request

```
POST /api/chargebacks

{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback":
            {
                <chargeback>
            },
            <resource>
        }
    ]
}
```

<chargeback> is same for all types of assignments. For **<chargeback>** can be used **href** or **id**:

- `{"href": "http://localhost:3090/api/chargebacks/1"}`
- `{"id": "1"}`

<resource> can be different across types of assignments - formats are described in next section.

Enterprise

<resource>:

```
"href": "http://localhost:3090/api/enterprises/1"
```

```
POST /api/chargebacks

{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback":
            {
                "id" : 2
            },
            "resource":
            {
                "href": "http://localhost:3090/api/enterprises/1"
            }
        }
    ]
}
```

Response

```
{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/enterprises/1",
                "name": "Enterprise"
            }
        }
    ]
}
```

Selected Providers

<resource>:

```
"href": "http://localhost:3090/api/enterprises/1"
```

Selected Clusters

<resource>:

```
"href": "http://localhost:3090/api/clusters/1"
```

Tagged VMs and Instances

Identification can be done by tag's name and tag's category, by **id** or by **href**.

assignment_prefix is string which partly determines type of assignment. In this case for "VMs and Instances" value for **assignment_prefix** is **vm**.

<resource>:

```
- "tag": {
            "category" : "department",
            "name" : "finance",
            "assignment_prefix": "vm"
         }
- "tag": {
            "id" : 28,
            "assignment_prefix": "vm"
         }
- "tag": {
            "href" : 'http://localhost:3090/api/tags/28,
            "assignment_prefix": "vm"
         }
```

Request

```
POST /api/chargebacks
```

```
{
  "action" : "assign",
  "assignments" : [
   {
       "chargeback": { "id" : 1},
       "tag": { "category" : "department", "name" : "finance", "assignment_prefix": "vm" }
     },
     {
       "chargeback": { "id" : 1},
       "tag": { "id" : 2}
     },
     {
       "chargeback": { "id" : 3},
       "tag": { "href" : "http://localhost:3090/api/tags/28", "assignment_prefix": "vm" }
     }
  ]
}
```

Tagged Container Images

Identification can be done by tag's name and tag's category, by **id** or by **href**.

assignment_prefix is string which partly determines type of assignment. - In this case for "Container Images" value for **assignment_prefix** is **container_image**

<resource>:

```
"tag":
{
    "category" : "department",
    "name" : "finance",
    "assignment_prefix": "container_image"
}
```

or

```
"tag":
{
    "id" : 28,
    "assignment_prefix": "container_image"
}
```

or

```
"tag":
{
    "href" : 'http://localhost:3090/api/tags/28,
    "assignment_prefix": "container_image"
}
```

Request

```
POST /api/chargebacks
```

```
{
  "action" : "assign",
  "assignments" : [
   {
       "chargeback": { "id" : 1},
       "tag": { "category" : "department", "name" : "finance", "assignment_prefix": "vm" }
     },
     {
       "chargeback": { "id" : 1},
```

```
      "tag": { "id" : 2}
    },
    {
      "chargeback": { "id" : 3},
      "tag": { "href" : "http://localhost:3090/api/tags/28", "assignment_prefix": "vm" }
    }
  ]
}
```

Labeled Container Images

<resource>:

```
"href": "http://localhost:3090/api/container_images/934/custom_attributes/30331"
```

This resource represents custom attributes on container images, so called “labels”.

Request

```
POST /api/chargebacks
```

```
{
  "action" : "assign",
    "assignments" :
    [
      {
        "chargeback": { "id" : 1},
        "resource": { "href": "http://localhost:3090/api/container_images/934/custom_attributes/30331" }
      },
      {
        "chargeback": { "id" : 1},
        "resource": { "href": "http://localhost:3090/api/container_images/934/custom_attributes/30324" }
      }
    ]
}
```

Tenants

<resource>:

```
"href": "http://localhost:3090/api/tenants/1"
```

Example

Request

```
POST /api/chargebacks
```

```
{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback": {
                "id" : 2
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/1"
            }
        },
        {
            "chargeback": {
                "id" : 2
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/2"
            }
        }
    ]
}
```

Response

```
{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/1",
                "name": "My Company"
            }
        },
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
```

```
            "resource": {
                "href": "http://localhost:3090/api/tenants/2",
                "name": "OpenStack Cloud Provider OpenStack"
            }
        }
    ]
}
```

Enterprise

<resource>:

```
"resource" :
{
    href": "http://localhost:3090/api/enterprises/1"
}
```

Request

POST /api/chargebacks

```
{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback":
            {
                "id" : 2
            },
            "resource":
            {
                "href": "http://localhost:3090/api/enterprises/1"
            }
        }
    ]
}
```

Selected Datastores

<resource>:

```
"resource":
{
    "href": "http://localhost:3090/api/enterprises/1"
}
```

Example

Request

POST /api/chargebacks

```
{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback" :
            {
                "id" : 2
            },
            "resource" :
            {
                "href" : "/api/data_stores/1"
            }
        }
    ]
}
```

Response

```
{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/data_stores/1",
                "name": "Shared-NFS"
            }
        }
    ]
}
```

Tagged Datastores

<resource>:

```
"tag":
{
    "href": "http://localhost:3090/api/tags/26",
    "assignment_prefix": "storage"
}
```

or

```
"tag":
{
    "category" : "department",
    "name" : "finance",
    "assignment_prefix": "storage"
}
```

or

```
"tag":
{
    "id" : "26",
    "assignment_prefix": "storage"
}
```

Example

Request

```
POST /api/chargebacks
```

```
{
    "action" : "assign",
    "assignments" : [
        {
            "tag": {
                "category" : "department",
                "name" : "finance",
                "assignment_prefix": "storage"
            }
        }
    ]
}
```

Response

```
{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "tag": {
                "href": "http://localhost:3090/api/tags/48",
                "name": "finance",
                "category": "department",
                "assignment_prefix": "storage"
            }
        }
    ]
}
```

Tenants

```
<resource>:
```

```
"href": "http://localhost:3090/api/tenants/1"
```

Request

```
POST /api/chargebacks
```

```
{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback": {
                "id" : 2
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/1"
            }
        },
        {
            "chargeback": {
                "id" : 2
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/2"
            }
```

```
        }
    ]
}
```

Response

```
{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/1",
                "name": "My Company"
            }
        },
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/2",
                "name": "OpenStack Cloud Provider OpenStack"
            }
        }
    ]
}
```

Unassign action for Chargeback Rates

Unassign action has same signature as assign action and action is ‘unassign’.

Example

Request

```
POST
```

```
{
  "action" : "unassign",
  "assignments" : [
     {
       "chargeback": { "id" : 3},
       "tag": { "category" : "prov_max_retirement_days", "name" : "30", "assignment_prefix": "vm" }
     }
  ]
}
```

Example of Chargeback Flow

- Create Chargeback Report
- Create Chargeback Rates
- Assign Chargeback Rates
- Run Chargeback Report
- Results of Chargeback Report

Create Chargeback Report

```
POST api/reports
```

```
{
  "action" : "import",
  "resource" : {
    "report" : {
        "rpt_group"   : "Custom",
        "rpt_type"    : "Custom",
        "menu_name"   : "Chargeback Flow Example",
        "col_order"   : ["vm_name", "display_range", ..],
        "cols"        : ["start_date", "display_range", "vm_name", ..],
        "headers"     : ["VM Name", "Date Range", ..],
        "title"       : "Chargeback Flow Example",
        "db"          : "ChargebackVm",
        "group"       : "y",
        "sortby"      : ["vm_name", "start_date"],
        "template_type" : "report",
        "order"         : "Ascending",
        "db_options" : {
            "rpt_type" : "ChargebackVm",
            "options"  : {
               <report_options>
            }
```

```
        }
    },
    "options" : { "save" : true }
  }
}
```

Attribute **"report"** contains JSON with report definition which can be obtained by exporting a chargeback report.

- **db** is type of report for chargeback purposes. Possible values are **ChargebackVm, ChargebackContainerProject, ChargebackContainerImage**.

Chargeback Report Options

Chargeback Report Options are present on index **options** in request json. There are also some options about filtering and grouping which are described in this section. Default attributes are described first and then there is explanation of attributes according to chargeback type of report(attribute **db**).

Example Request with **options**

POST /api/chargebacks

```
"db_options" : {
        "rpt_type" : "ChargebackVm",
        "options"  : {
            "interval" : "monthly",
            "interval_size" : 12,
            "end_interval_offset" : 1,
            "tenant_id" : "1",
            "method_for_allocated_metrics" : "max",
            "include_metrics:" : true,
            "cumulative_rate_calculation" : false,
            "groupby" : "date",
            "groupby_tag" : null,
            "groupby_label" : null
        }
}
...,
```

- **method_for_allocated_metrics** - **max** or **avg**
- **include_metrics** - **true** or **false** - enable/disable whether to include metrics data, when disable it uses current values of metric data or charged resources
- **cumulative_rate_calculation** - **true** or **false** enable/disable cumulative calculation for tagged resources
- **interval** is which time detail is used for the report. Possible values are **monthly, weekly, daily**

Metric data for chargeback report are taken from this interval:

`<Time.now-interval_size, Time.now-end_interval_offset>`

- **interval_size, end_interval_offset** - attributes in **options**. there is used unit according to interval parameter, for example: when interval is “daily” then **interval_size=3** and **end_interval_offset=2** mean that generated report will be based on metrics starting 3 days ago to 1 day ago.

Request

POST /api/chargebacks

```
"report" : {
    "db_options" : {
        "options" : {
            ...
        }
    },
    ...
}
```

Filtering according to resource

This is part of json used in request below.

Each section lists possible options to filter according to resource’s type. There are used **ids** for identification each type of resource and and those **ids** are used in attributes according to type of resources. Name of attributes can be find in following sections. Attributes are located in json for request(example with tenant) on indexes **report,**
db_options and options:

Request

POST /api/chargebacks

```
"report" : {
    "db_options" : {
        "options" : {
            "tenant_id" : 1,
            ...
        }
    },
    ...
}
```

Chargeback for VMs (db=ChargebackVm)

tenant_id - Tenant id

or

owner - **userid** of user which owns VMs.

Request

```
POST /api/chargebacks

"report" : {
    "db_options" : {
        "options" : {
            "owner" : "admin",
            ...
        }
    },
    ...
}
```

or

tag - tag_path of tag

Request

```
POST /api/chargebacks

"report" : {
    "db_options" : {
        "options" : {
            "tag" : "/managed/prov_max_cpu/2",
            ...
        }
    },
    ...
}
```

Chargeback for Projects (db=ChargebackContainerProject)

tag - tag_path of tag

Request

```
POST /api/chargebacks

"report" : {
    "db_options" : {
        "options" : {
            "tag" : "/managed/prov_max_cpu/2",
            ...
        }
    },
    ...
}
```

or

provider_id - “all” or Openshift Provider id

entity_id - “all” or Container Project id

```
"report" : {
    "db_options" : {
        "options" : {
            "provider_id" : "41",
            "entity_id" : "21",
            ...
        }
    },
    ...
}
```

Chargeback for Container Images (db=ChargebackContainerImage)

provider_id - “all” or Openshift Provider id

entity_id - “all” or Container Image id

Request

```
POST /api/chargebacks

"report" : {
    "db_options" : {
        "options" : {
            "provider_id" : "all",
```

```
            "entity_id" : "all",
            ...
        }
    }
}
```

Grouping Report Options

Chargeback for VMs (db=ChargebackVm)

groupby - specifies a grouping option

Possible values for **groupby**:

date - grouping by date and Vms

date-only - grouping only by date

tenant - grouping by tenants

tag - name of tag category - attribute **groupby_tag** needs to be also specified. Values for **groupby_tag** are names of tag category which can be retrieved in **resources**

```
GET
/api/categories?expand=resources&attributes=name
```

Chargeback for Projects (db=ChargebackContainerProject)

groupby - specifies a grouping option

Possible values for **groupby**:

date - grouping by date and container projects

date-only - grouping only by date

project - grouping by container projects

tag - - name of tag category - attribute **groupby_tag** needs to be also specified. Values for **groupby_tag** are names of tag category which can be retrieved in **resources**

```
GET
/api/categories?expand=resources&attributes=name
```

Chargeback for Container Images (db=ChargebackContainerImage)

groupby - specifies a grouping option

Possible values for **groupby**:

date - grouping by date and containers

date-only - grouping only by date

project - grouping by container images

label - grouping by container labels and when this value is set - attribute **groupby_tag** needs to be also specified. Values for **groupby_tag** are names of labels which can be retrieved by **api/container_images/:id/custom_attributes** endpoint (attribute=name) for single container image.

Example request

```
GET
/api/container_images/934/custom_attributes?expand=resources&attributes=name&filter[]=section=docker_labels
```

Response

```
{
    "name": "custom_attributes",
    "count": 2041860,
    "subcount": 11,
    "subquery_count": 11,
    "pages": 1,
    "resources": [
        {
            "href": "http://localhost:3090/api/container_images/934/custom_attributes/30322",
            "name": "build-date",
            "id": "30322"
        },
        {
            "href": "http://localhost:3090/api/container_images/934/custom_attributes/30324",
            "name": "architecture",
            "id": "30324"
        }
    ],
    ...
}
```

Example of request to create chargeback report

Request

```
POST /api/chargebacks
```

```
{
  "action" : "import",
  "resource" : {
    "report" : {
        "rpt_group"  : "Custom",
        "rpt_type"   : "Custom",
        "menu_name"  : "Chargeback Flow Example 1",
        "col_order"  : ["vm_name", "display_range", "chargeback_rates", "memory_allocated_metric",
"memory_allocated_cost", "memory_allocated_rate", "memory_cost"],
        "cols"       : ["start_date", "display_range", "vm_name", "chargeback_rates", "memory_allocated_metric",
"memory_allocated_cost", "memory_allocated_rate", "memory_cost"],
        "headers"    : ["VM Name", "Date Range", "Chargeback Rates", "Memory Allocated over Time Period", "Memory
Allocated Cost" , "Memory Allocated Rate","Memory Total Cost"],
        "title"      : "Chargeback Flow Example 1",
        "db"         : "ChargebackVm",
        "group"      : "y",
        "sortby"     : ["vm_name", "start_date"],
        "template_type" : "report",
        "order"         : "Ascending",
        "db_options" : {
            "rpt_type" : "ChargebackVm",
            "options"  : {
                "interval" : "monthly",
                "interval_size" : 12,
                "end_interval_offset" : 1,
                "tenant_id" : "1",
                "method_for_allocated_metrics" : "max",
                "include_metrics:" : true,
                "cumulative_rate_calculation" : false,
                "groupby" : "date",
                "groupby_tag" : null,
                "groupby_label" : null
            }
        }
    },
    "options" : { "save" : true }
  }
}
```

Response

```
{
    "results": [
        {
            "success": true,
            "message": "Imported Report: [Chargeback Flow Example 1]",
            "result": {
                "id": "553",
                "name": "Chargeback Flow Example 1",
                "title": "Chargeback Flow Example 1",
                "rpt_group": "Custom",
                "rpt_type": "Custom",
                "priority": null,
                "db": "ChargebackVm",
                "cols": [
                    "start_date",
                    "display_range",
                    "vm_name",
                    "chargeback_rates",
                    "memory_allocated_metric",
                    "memory_allocated_cost",
                    "memory_allocated_rate",
                    "memory_cost"
                ],
                "include": null,
                "col_order": [
                    "vm_name",
                    "display_range",
                    "chargeback_rates",
                    "memory_allocated_metric",
                    "memory_allocated_cost",
                    "memory_allocated_rate",
                    "memory_cost"
                ],
                "headers": [
                    "VM Name",
                    "Date Range",
                    "Chargeback Rates",
                    "Memory Allocated over Time Period",
                    "Memory Allocated Cost",
                    "Memory Allocated Rate",
                    "Memory Total Cost"
                ],
                "conditions": null,
                "order": "Ascending",
```

```
                "sortby": [
                    "vm_name",
                    "start_date"
                ],
                "group": "y",
                "graph": null,
                "dims": null,
                "created_on": "2020-05-19T12:16:31Z",
                "updated_on": "2020-05-19T12:16:31Z",
                "filename": null,
                "file_mtime": null,
                "categories": null,
                "timeline": null,
                "template_type": "report",
                "where_clause": null,
                "db_options": {
                    "rpt_type": "ChargebackVm",
                    "options": {
                        "interval": "monthly",
                        "interval_size": 12,
                        "end_interval_offset": 1,
                        "tenant_id": "1",
                        "method_for_allocated_metrics": "max",
                        "include_metrics:": true,
                        "cumulative_rate_calculation": false,
                        "groupby": "date",
                        "groupby_tag": null,
                        "groupby_label": null
                    }
                },
                "generate_cols": null,
                "generate_rows": null,
                "col_formats": null,
                "tz": null,
                "time_profile_id": null,
                "display_filter": null,
                "col_options": null,
                "rpt_options": null,
                "miq_group_id": "2",
                "user_id": "1"
            }
        }
    ]
}
```


Create Chargeback Rates

Chargeback rate is complex structure. Create chargeback rate requires two endpoints: **`/api/chargebacks`** and **`/api/rates`**. Endpoint **`/api/chargebacks`** manage top level rate entity which aggregate other relations like rate details(**`api/rates`**), currencies(**`api/currencies`**), measures(**`api/measures`**), chargeable_fields(**`api/chargeable_fields`**) and chargeback tiers. This entity is also used to assigning rates to any resource.

Create Chargebacks (top-level rate entity)

Request

```
POST /api/chargebacks
```


```
{
  "guid" : "daily",
  "description" : "1",
  "rate_type" : "Compute"
}
```


Response


```
{
    "results": [
        {
            "href": "http://localhost:3090/api/chargebacks/14",
            "id": "14",
            "guid": "daily",
            "description": "1",
            "rate_type": "Compute",
            "created_on": "2020-05-26T12:15:51Z",
            "updated_on": "2020-05-26T12:15:51Z",
            "default": false
        }
    ]
}
```


Now we need to create rate details(**`api/rates`**) which includes chargebable_fields, currenies, measures and tiers. It is also important assign rate details to chargeback top level entity by **`chargeback_rate`** attribute.

Request

```
POST /api/rates
```


```
{
  "per_time" : "daily",
  "chargeback_rate" : {"href": "http://localhost:3090/api/chargebacks/14"},
```

```
  "chargeable_field_id" : "1",
  "description": "My CPU allocation rate",
  "group" : "cpu",
  "per_unit" : "megahertz",
  "source" : "allocated",
  "chargeback_rate_detail_measure_id": 1,
  "detail_currency" : {"id": 5},
  "detail_measure" : {"href": "http://localhost:3090/api/measures/3"},
  "chargeable_field" : {"href": "http://localhost:3090/api/chargeable_fields/3"},
  "chargeback_tiers" : [
    {
      "start": 0.0,
      "finish":1,
      "fixed_rate": 2.0,
      "variable_rate": 0.0
    },
    {
      "start": 1,
      "finish": "infinity",
      "fixed_rate": 2.0,
      "variable_rate": 0.0
    }
  ]
}
```

Response

```
{
    "results": [
        {
            "href": "http://localhost:3090/api/rates/1862",
            "id": "1862",
            "enabled": true,
            "description": "My CPU allocation rate",
            "group": "cpu",
            "source": "allocated",
            "metric": null,
            "per_time": "daily",
            "per_unit": "megahertz",
            "friendly_rate": "Daily @ 2.0 + 0.0 per Megahertz from 0.0 to 1.0\nDaily @ 2.0 + 0.0 per Megahertz from 1.0 to
Infinity",
            "chargeback_rate_id": "14",
            "created_on": "2020-05-26T12:36:32Z",
            "updated_on": "2020-05-26T12:36:32Z",
            "chargeback_rate_detail_measure_id": "3",
            "chargeback_rate_detail_currency_id": "5",
            "chargeable_field_id": "3",
            "sub_metric": null
        }
    ]
}
```

Assign Chargeback Rate

There are examples in those sections:

- Querying Assigned Chargeback Rates
- Assign action for Chargeback Rates
- Unassign action for Chargeback Rates

Run Chargeback Report

Please refer to Run Report section for details on how to run chargeback report. Running report requires report id or report href:

Report id or report href can be obtained by chargeback report creation or by listing chargeback reports:

Request

```
GET /api/reports?expand=resources&attributes=db&filter[]=db='ChargebackVm'&filter[]=or db='ChargebackContainer%'
```

Response

```
{
    "name": "reports",
    "count": 519,
    "subcount": 47,
    "subquery_count": 47,
    "pages": 1,
    "resources": [
        {
            "href": "http://localhost:3090/api/reports/157",
            "db": "ChargebackContainerImage",
            "id": "157"
        },
        {
            "href": "http://localhost:3090/api/reports/152",
            "db": "ChargebackVm",
            "id": "152"
```

```
        },
...
}
```

Results of Chargeback Report

There is request which list of report result.

Allowed parameters:

hash_attribute=value - enables hash sorting and pagination for value column

sort_by - column from **MiqReportResult#report#col_order**

sort_order - **asc** or **desc**

limit - count of records for one page

offset - result will start form **<offset>**. record

filter_column - chosen column for filtering

filter_string - this string value is used for filtering on **<filter_column>**, record of report result will be returned in API when **<filter_string>** is contained in value of **<filter_column>** .

There is possibility to filter according to more columns. We need to add other filter parameters with suffix **_X** where **X** is number (**..&filter_column_1=name&filter_string_1=TIE&filter_column_2=hostname&filter_string_2=test**)

Request

```
GET /api/results/23039?hash_attribute=result_set&sort_by=&sort_order=&limit=20&offset=0&filter_string=windemo
```

Response

```
{
    "href": "http://localhost:3090/api/results/23039",
    "id": "23039",
    "name": "Chargeback - Demo",
    ...,
    "result_set": [
        {
            "start_date": "01/01/19 00:00:00 UTC",
            "display_range": "Jan 2019",
            "vm_name": "windemo",
            "owner_name": "Cloud Operators",
            "provider_name": "vCenter",
            "total_cost": "$48,303.51",
            "cpu_cost": "$935.65",
            "cpu_used_cost": "$191.65",
            "cpu_used_metric": "12.88 MHz",
            "memory_cost": "$1,609.69",
            "memory_used_cost": "$1,609.69",
            "memory_used_metric": "108.18 MB",
            "net_io_used_cost": "$372.00",
            "net_io_used_metric": "0.00 KBps",
            "chargeback_rates": "Default"
        },
        {
            "start_date": "02/01/19 00:00:00 UTC",
            "display_range": "Feb 2019",
            "vm_name": "windemo",
            "owner_name": "Cloud Operators",
            "provider_name": "vCenter",
            "total_cost": "$43,851.09",
            "cpu_cost": "$861.44",
            "cpu_used_cost": "$189.44",
            "cpu_used_metric": "14.09 MHz",
            "memory_cost": "$1,659.73",
            "memory_used_cost": "$1,659.73",
            "memory_used_metric": "123.49 MB",
            "net_io_used_cost": "$336.00",
            "net_io_used_metric": "0.00 KBps",
            "chargeback_rates": "Default"
        },
        {
            "start_date": "03/01/19 00:00:00 UTC",
            "display_range": "Mar 2019",
            "vm_name": "windemo",
            "owner_name": "Cloud Operators",
            "provider_name": "vCenter",
            "total_cost": "$28,906.99",
            "cpu_cost": "$553.43",
            "cpu_used_cost": "$113.43",
            "cpu_used_metric": "12.89 MHz",
            "memory_cost": "$1,231.94",
            "memory_used_cost": "$1,231.94",
            "memory_used_metric": "139.99 MB",
            "net_io_used_cost": "$220.00",
            "net_io_used_metric": "0.00 KBps",
            "chargeback_rates": "Default"
```

```
        }
    ],
    "count": 3,
    "subcount": 3,
    "pages": 1
}
```


- Cloud Volumes
  - Querying Cloud Volumes
  - Deleting Cloud Volumes

Cloud Volumes

Management of Cloud Volumes is provided via the following collection:

```
/api/cloud_volumes
```

The following actions are available on cloud_volumes:

- Querying Cloud Volumes
- Deleting Cloud Volumes

Querying Cloud Volumes

Querying all cloud volumes in the system is simply:

```
GET /api/cloud_volumes
```

And getting details of a specific Cloud Volume is as follows:

```
GET /api/cloud_volumes/:id
```

Deleting Cloud Volumes

Cloud Volumes can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/cloud_volumes/101
```

```
{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/cloud_volumes/101
```

Deleting multiple Cloud Volumes can be done as follows:

```
POST /api/cloud_volumes
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/cloud_volumes/101" },
    { "href" : "http://localhost:3000/api/cloud_volumes/102" },
    ...
  ]
}
```


- Conditions
  - Querying Conditions
  - Creating Conditions
  - Editing Conditions
  - Deleting Conditions

Conditions

Management of conditions is provided via the following collection:

```
/api/conditions
```

Full CRUD actions on conditions is available:

- Querying Conditions
- Creating Conditions
- Editing Conditions
- Deleting Conditions

Querying Conditions

Querying all conditions in the system is simply:

```
GET /api/conditions
```

Getting details on a specific Condition:

```
GET /api/conditions/:id
```

Creating Conditions

Conditions can be created via a POST to the conditions collection or via the **create** action signature which also allows creation of multiple conditions in a single request.

```
POST /api/conditions
```

```
{
  "name" : "sample_condition",
  "description" : "Sample Condition",
  "expression" : { "=" : { "field" : "ExtManagementSystem.container_images-architecture", "value" : "dsa" } },
  "towhat" : "ExtManagementSystem",
  "modifier" : "allow"
}
```

or creating multiple conditions:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "condition_1", "description" : "Condition 1", ... },
    { "name" : "condition_2", "description" : "Condition 2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating conditions.

Editing Conditions

```
POST /api/conditions/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Condition Description"
  }
}
```

or editing multiple conditions:

```
POST /api/conditions
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/conditions/101",
      "description" : "Updated Condition Description 1"
    },
    {
      "href" : "http://localhost:3000/api/conditions/102",
      "description" : "Updated Condition Description 2"
    },
    ...
  ]
}
```

Deleting Conditions

Conditions can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/conditions/101
```

```
{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/conditions/101
```

Deleting multiple conditions can be done as follows:

```
POST /api/conditions
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/conditions/101" },
    { "href" : "http://localhost:3000/api/conditions/102" },
    ...
```

```
  ]
}
```


- Configuration Script Payloads
  - Querying Configuration Script Payloads
  - Editing configuration script payloads
  - Updating configuration script payloads
  - Patch method:
  - Querying Configuration Script Payload Authentication
  - Creating Configuration Script Payload Authentication

Configuration Script Payloads

Management of configuration script payloads is provided through the following collection:

`/api/configuration_script_payloads`

The following actions on configuration script payloads are available:

- Querying Configuration Script Payloads
- Editing Configuration Script Payloads
- Updating Configuration Script Payloads
- Querying Configuration Script Payload Authentication
- Creating Configuration Script Payload Authentication

Querying Configuration Script Payloads

Query all configuration script payloads in the system:

`GET /api/configuration_script_payloads`

Example response:

```
{
  "name": "configuration_script_payloads",
  "count": 2,
  "subcount": 2,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    },
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    }
  ],
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/configuration_script_payloads"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/configuration_script_payloads?offset=0",
    "first": "http://localhost:3000/api/configuration_script_payloads?offset=0",
    "last": "http://localhost:3000/api/configuration_script_payloads?offset=0"
  }
}
```

Query the details on a specific configuration script payload:

`GET /api/configuration_script_payloads/:id`

Example response:

```
{
  "href": "http://localhost:3000/api/configuration_script_payloads/1",
  "id": "1",
  "manager_id": "2",
  "manager_ref": null,
  "name": "provision-vm-service/list-providers.asl",
  "description": null,
  "variables": null,
  "created_at": "2023-11-08T17:43:30Z",
  "updated_at": "2023-11-08T17:46:03Z",
  "survey_spec": null,
  "inventory_root_group_id": null,
  "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
  "parent_id": null,
  "configuration_script_source_id": "11",
  "run_by_userid": null,
  "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\": \"ListProviders\",\n  \"States\": {\n
```

```
\"ListProviders\": {\n      \"Type\": \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-
providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n
\"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\":
\"$.api_bearer_token\"\n      },\n      \"Parameters\": {\n        \"API_URL.$\": \"$$.Execution._manageiq_api_url\",\n
\"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n\n        \"PROVIDER_TYPE\":
\"ManageIQ::Providers::Vmware::InfraManager\"\n      }\n    }\n  }\n}\n",
  "payload_type": "json",
  "credentials": null,
  "context": null,
  "output": null,
  "status": null,
  "miq_task_id": null,
  "payload_valid": true,
  "payload_error": null,
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    },
    {
      "name": "edit",
      "method": "patch",
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    },
    {
      "name": "edit",
      "method": "put",
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    }
  ]
}
```


Editing configuration script payloads

Configuration script payloads can be edited by using a POST to the configuration script payloads collection:

```
POST /api/configuration_script_payloads
```

POST /api/configuration_script_payloads

```
{
  "action": "edit",
  "resources": [
    {"id": 1, "description": "A Helpful Description"}
  ]
}
```

Example response:


```
{
  "results": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1",
      "description": "A Helpful Description",
      "credentials": {
        "api_user.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "userid"
        },
        "api_password.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "password"
        }
      },
      "id": "1",
      "manager_id": "2",
      "manager_ref": null,
      "name": "provision-vm-service/list-providers.asl",
      "variables": null,
      "created_at": "2023-11-08T17:43:30Z",
      "updated_at": "2023-11-15T20:14:28Z",
      "survey_spec": null,
      "inventory_root_group_id": null,
      "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
      "parent_id": null,
      "configuration_script_source_id": "2",
      "run_by_userid": null,
      "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\": \"ListProviders\",\n  \"States\": {\n
\"ListProviders\": {\n      \"Type\": \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-
providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n
\"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\":
\"$.api_bearer_token\"\n      },\n      \"Parameters\": {\n        \"API_URL.$\": \"$$.Execution._manageiq_api_url\",\n
\"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n\n        \"PROVIDER_TYPE\":
\"ManageIQ::Providers::Vmware::InfraManager\"\n      }\n    }\n  }\n}\n",
      "payload_type": "json",
      "context": null,
      "output": null,
      "status": null,
      "miq_task_id": null,
      "payload_valid": true,
```

```
      "payload_error": null
    }
  ]
}
```

For instance, configuration script payloads can be edited to map credentials:

```
POST /api/configuration_script_payloads
```

```
{
  "action": "edit",
  "resources": [
    {
      "id": 1,
      "credentials": [
        {"my-cred-user": {"credential_ref": "my-credential", "credential_field": "userid"}},
        {"my-cred-password": {"credential_ref": "my-credential", "credential_field": "password"}}
      ]
    }
  ]
}
```

Updating configuration script payloads

Configuration script payloads can be updated by using a PUT method or PATCH method.

PUT method:

```
PUT /api/configuration_script_payloads/:id
```

Example request:

```
PUT /api/configuration_script_payloads/1 {"description": "A Helpful Description"}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1",
      "description": "A Helpful Description",
      "credentials": {
        "api_user.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "userid"
        },
        "api_password.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "password"
        }
      },
      "id": "1",
      "manager_id": "2",
      "manager_ref": null,
      "name": "provision-vm-service/list-providers.asl",
      "variables": null,
      "created_at": "2023-11-08T17:43:30Z",
      "updated_at": "2023-11-15T20:14:28Z",
      "survey_spec": null,
      "inventory_root_group_id": null,
      "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
      "parent_id": null,
      "configuration_script_source_id": "2",
      "run_by_userid": null,
      "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\": \"ListProviders\",\n  \"States\": {\n
\"ListProviders\": {\n      \"Type\": \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-
providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n
\"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\":
\"$.api_bearer_token\"\n      },\n      \"Parameters\": {\n        \"API_URL.$\": \"$$.Execution._manageiq_api_url\",\n
\"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n\n        \"PROVIDER_TYPE\":
\"ManageIQ::Providers::Vmware::InfraManager\"\n      }\n    }\n  }\n}\n",
      "payload_type": "json",
      "context": null,
      "output": null,
      "status": null,
      "miq_task_id": null,
      "payload_valid": true,
      "payload_error": null
    }
  ]
}
```

Patch method:

```
PATCH /api/configuration_script_payloads/:id
```

Example request:

```
PATCH /api/configuration_script_payloads/1 {"description": "A Helpful Description"}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1",
      "description": "A Helpful Description",
      "credentials": {
        "api_user.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "userid"
        },
        "api_password.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "password"
        }
      },
      "id": "1",
      "manager_id": "2",
      "manager_ref": null,
      "name": "provision-vm-service/list-providers.asl",
      "variables": null,
      "created_at": "2023-11-08T17:43:30Z",
      "updated_at": "2023-11-15T20:14:28Z",
      "survey_spec": null,
      "inventory_root_group_id": null,
      "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
      "parent_id": null,
      "configuration_script_source_id": "2",
      "run_by_userid": null,
      "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\": \"ListProviders\",\n  \"States\": {\n
\"ListProviders\": {\n      \"Type\": \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-
providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n
\"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\":
\"$.api_bearer_token\"\n      },\n      \"Parameters\": {\n        \"API_URL.$\": \"$$.Execution._manageiq_api_url\",\n
\"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n\n        \"PROVIDER_TYPE\":
\"ManageIQ::Providers::Vmware::InfraManager\"\n      }\n    }\n  }\n}\n",
      "payload_type": "json",
      "context": null,
      "output": null,
      "status": null,
      "miq_task_id": null,
      "payload_valid": true,
      "payload_error": null
    }
  ]
}
```


Querying Configuration Script Payload Authentication

Query the authentication credentials for a specific configuration script payload:

```
GET /api/configuration_script_payloads/:id/authentications
```

Example response:


```
{
  "name": "authentications",
  "count": 7,
  "subcount": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/2/authentications/3"
    }
  ],
  "actions": [
    {
      "name": "create",
      "method": "post",
      "href": "http://localhost:3000/api/configuration_script_payloads/2/authentications"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/configuration_script_payloads/2/authentications?offset=0",
    "first": "http://localhost:3000/api/configuration_script_payloads/2/authentications?offset=0",
    "last": "http://localhost:3000/api/configuration_script_payloads/2/authentications?offset=0"
  }
}
```


Get the details on a specific authentication credential for a configuration script payload:

```
GET /api/configuration_script_payloads/:id/authentications/:id
```

Example response:


```
{
  "href": "http://localhost:3000/api/configuration_script_payloads/2/authentications/3",
  "id": "3",
  "name": "manageiq_api",
  "authtype": null,
  "userid": "admin",
  "password": null,
```

```
  "resource_id": "25",
  "resource_type": "ExtManagementSystem",
  "created_on": "2023-11-15T19:59:23Z",
  "updated_on": "2023-11-15T19:59:23Z",
  "last_valid_on": null,
  "last_invalid_on": null,
  "credentials_changed_on": null,
  "status": null,
  "status_details": null,
  "type": "ManageIQ::Providers::Workflows::AutomationManager::WorkflowCredential",
  "auth_key": null,
  "fingerprint": null,
  "service_account": null,
  "public_key": null,
  "manager_ref": "3",
  "options": null,
  "evm_owner_id": null,
  "miq_group_id": "1",
  "tenant_id": "1",
  "become_username": null,
  "become_password": null,
  "auth_key_password": null,
  "ems_ref": "manageiq_api",
  "become_method": null
}
```


Creating Configuration Script Payload Authentication

Authentication credentials for a configuration script payload can be created by using a POST to the configuration script payload collection:

```
POST /api/configuration_script_payloads/:id/authentications
```


```
{
  "action": "create",
  "description": "Description",
  "name": "my credential",
  "type": "ManageIQ::Providers::AnsibleTower::AutomationManager::Credential"
}
```


- Configuration Script Sources
  - Querying Configuration Script Sources
  - Creating Configuration Script Sources
  - Editing Configuration Script Sources
  - Refreshing Configuration Script Sources
  - Request:
  - Response:
  - Deleting Configuration Script Sources

Configuration Script Sources

Management of configuration script sources is provided via the following collection:

```
/api/configuration_script_sources
```

Full CRUD actions on configuration script sources is available:

- Querying Configuration Script Sources
- Creating Configuration Script Sources
- Editing Configuration Script Sources
- Refreshing Configuration Script Sources
- Deleting Configuration Script Sources

Querying Configuration Script Sources

Querying all configuration script sources in the system is simply:

```
GET /api/configuration_script_sources
```

Getting details on a specific configuration script source:

```
GET /api/configuration_script_sources/:id
```

Querying configuration script payloads of a particular configuration script source:

```
GET /api/configuration_script_sources/:id/configuration_script_payloads?expand=resources
```

Querying the configuration script payloads of a particular configuration script source that belong to a specific region:

```
GET /api/configuration_script_sources/:id/configuration_script_payloads?expand=resources&filter[]=region_number=10
```

Creating Configuration Script Sources

Configuration script sources can be created via a POST to the configuration script sources collection or via the **create** action signature which also allows creation of multiple configuration script sources in a single request.

```
POST /api/configuration_script_sources

{
  "name" : "My Project",
  "description" : "My Project's Description",
  "manager_resource" : { "href" : "http://localhost:3000/api/providers/7" },
  "related" : {}
}
```

or creating multiple configuration script sources:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "My First Project", "description" : "My First Project Description", ... },
    { "name" : "My Second Project", "description" : "My Second Project Description", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating configuration script sources.

Editing Configuration Script Sources

```
POST /api/configuration_script_sources/:id

{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Project Description"
  }
}
```

or editing multiple configuration script sources:

```
POST /api/configuration_script_sources

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/configuration_script_sources/101",
      "description" : "Updated Project Description 1"
    },
    {
      "href" : "http://localhost:3000/api/configuration_script_sources/102",
      "description" : "Updated Project Description 2"
    },
    ...
  ]
}
```

Refreshing Configuration Script Sources

Configuration script sources can be refreshed by posting a **refresh** action to a single resource or to multiple resources in bulk by targeting the collection.

```
POST /api/configuration_script_sources/:id

{
  "action" : "refresh"
}
```

or refreshing multiple configuration script sources:

```
POST /api/configuration_script_sources

{
  "action" : "refresh",
  "resources" : [
    { "id" : "101" },
    { "id" : "102" }
  ]
}
```

Example:

Request:

```
POST /api/configuration_script_sources/101

{
  "action" : "refresh"
}
```

Response:

```
{
  "success" : true,
  "message" : "Refreshing ConfigurationScriptSource id:101 name:'Test Configuration Script Source'",
  "task_id" : "7",
  "task_href" : "http://localhost:3000/api/tasks/7",
  "tasks" : [
    {
      "id" : "7",
      "href" : "http://localhost:3000/api/tasks/7"
    }
  ]
}
```

Deleting Configuration Script Sources

Configuration script sources can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/configuration_script_sources/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/configuration_script_sources/101
```

Deleting multiple configuration script sources can be done as follows:

```
POST /api/configuration_script_sources

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/configuration_script_sources/101" },
    { "href" : "http://localhost:3000/api/configuration_script_sources/102" },
    ...
  ]
}
```


- Custom Actions
  - Request:
  - Response:
    - Request:
    - Response:

Custom Actions

With button groups and custom buttons defined on records, the REST API provides visibility to custom actions that represent the custom buttons for the related resources. The custom actions are not returned when querying collections, they are only available when fetching an individual resource, e.g. /api/:collection/:id

This is made available with the *custom_actions* and *custom_action_buttons* virtual attributes on resources for the following collections:

/api/cloud_tenants
/api/cloud_volumes
/api/clusters
/api/container_nodes
/api/data_stores
/api/generic_objects
/api/hosts
/api/providers
/api/templates
/api/vms

- *custom_actions* return the custom buttons as laid out in the UI on the resource. The information returned includes any Resource Actions and related Dialogs for the custom button.
- *custom_action_buttons* return the array of custom buttons without the custom button groups hierarchy or details on related dialogs.

Fetching custom actions and custom action buttons for a resource:

```
GET /api/:collection/:id?attributes=custom_actions,custom_action_buttons
```

Example VM with custom buttons:

Request:

```
GET /api/vms/173?attributes=custom_actions,custom_action_buttons
```

Response:

```
{
  "href": "https://172.16.30.129/api/vms/173",
  "id": "173",
  "vendor": "vmware",
  "name": "aab-ldap",
  "location": "aab-ldap/aab-ldap.vmx",
  "host_id": "4",
  "created_on": "2018-01-09T20:23:37Z",
  "updated_on": "2018-01-09T20:23:37Z",
  "storage_id": "8",
  "guid": "dee6d969-7c69-44b6-bcd3-3b14f21702c5",
  "ems_id": "1",
  "uid_ems": "420c8f0d-4a8f-0b6d-fc3a-0c3cfab7b33f",
  "tools_status": "toolsNotRunning",
  "standby_action": "checkpoint",
  "power_state": "off",
  ...
  "custom_actions": {
    "buttons": [

    ],
    "button_groups": [
      {
        "id": "23",
        "name": "VM Actions|Vm|",
        "description": "VM Custom Actions",
        "created_on": "2018-01-09T20:36:50Z",
        "updated_on": "2018-01-09T20:41:40Z",
        "guid": "559b37d1-3232-4c51-8668-f994a5b9149e",
        "set_data": {
          "button_order": [
            1,
            2
          ],
          "button_icon": "pficon pficon-virtual-machine",
          "display": true,
          "applies_to_class": "Vm",
          "group_index": 1
        },
        "buttons": [
          {
            "id": "1",
            "guid": "22b5bf4f-d5d6-43af-bc7b-77460d120065",
            "description": "Analyze Virtual Machine",
            "applies_to_class": "Vm",
            "options": {
              "button_icon": "ff ff-timeline",
              "button_type": "default",
              "display": true,
              "open_url": false,
              "display_for": "single",
              "submit_how": "one"
            },
            "userid": "admin",
            "created_on": "2018-01-09T20:40:11Z",
            "updated_on": "2018-01-09T20:40:11Z",
            "name": "Analyze",
            "visibility": {
              "roles": [
                "_ALL_"
              ]
            },
            "resource_action": {
              "id": "21",
              "resource_id": "1",
              "resource_type": "CustomButton",
              "created_at": "2018-01-09T20:40:11Z",
              "updated_at": "2018-01-09T20:40:11Z",
              "ae_namespace": "SYSTEM",
              "ae_class": "PROCESS",
              "ae_instance": "Automation",
              "ae_attributes": {
                "request": "general/process/analyze"
              }
            },
            "enabled": true,
            "href": "https://172.16.30.129/api/custom_buttons/1"
          },
          {
            "id": "2",
            "guid": "e9e77bb3-f52a-4d08-8d58-13923031c6de",
            "description": "Ping Virtual Machine",
            "applies_to_class": "Vm",
            "options": {
              "button_icon": "ff ff-scsi",
              "button_color": "#b4c10f",
              "button_type": "default",
              "display": true,
              "open_url": false,
              "display_for": "single",
              "submit_how": "one"
```

```
            },
            "userid": "admin",
            "created_on": "2018-01-09T20:41:40Z",
            "updated_on": "2018-01-09T20:41:40Z",
            "name": "Ping",
            "visibility": {
              "roles": [
                "_ALL_"
              ]
            },
            "resource_action": {
              "id": "22",
              "resource_id": "2",
              "resource_type": "CustomButton",
              "created_at": "2018-01-09T20:41:40Z",
              "updated_at": "2018-01-09T20:41:40Z",
              "ae_namespace": "SYSTEM",
              "ae_class": "PROCESS",
              "ae_instance": "Automation",
              "ae_attributes": {
                "request": "general/virtual_machines/ping"
              }
            },
            "enabled": true,
            "href": "https://172.16.30.129/api/custom_buttons/2"
          }
        ],
        "href": "https://172.16.30.129/api/custom_button_sets/23"
      }
    ]
  },
  "custom_action_buttons": [
    {
      "href": "https://172.16.30.129/api/custom_buttons/1",
      "id": "1",
      "guid": "22b5bf4f-d5d6-43af-bc7b-77460d120065",
      "description": "Analyze Virtual Machine",
      "applies_to_class": "Vm",
      "options": {
        "button_icon": "ff ff-timeline",
        "button_type": "default",
        "display": true,
        "open_url": false,
        "display_for": "single",
        "submit_how": "one"
      },
      "userid": "admin",
      "created_on": "2018-01-09T20:40:11Z",
      "updated_on": "2018-01-09T20:40:11Z",
      "name": "Analyze",
      "visibility": {
        "roles": [
          "_ALL_"
        ]
      }
    },
    {
      "href": "https://172.16.30.129/api/custom_buttons/2",
      "id": "2",
      "guid": "e9e77bb3-f52a-4d08-8d58-13923031c6de",
      "description": "Ping Virtual Machine",
      "applies_to_class": "Vm",
      "options": {
        "button_icon": "ff ff-scsi",
        "button_color": "#b4c10f",
        "button_type": "default",
        "display": true,
        "open_url": false,
        "display_for": "single",
        "submit_how": "one"
      },
      "userid": "admin",
      "created_on": "2018-01-09T20:41:40Z",
      "updated_on": "2018-01-09T20:41:40Z",
      "name": "Ping",
      "visibility": {
        "roles": [
          "_ALL_"
        ]
      }
    }
  ],
  "actions": [
    ...
  ]
}
```


When custom action buttons are defined on a VM, the returned VM resource is extended with custom actions which can be triggered via a POST action.

The action name to trigger the action is the lower-cased name of the custom button.

As an example, here's a VM without custom buttons so all actions returned are the common ones defined by the system:

```
{
  "href": "https://172.16.30.129/api/vms/172",
  "id": "172",
  "vendor": "vmware",
  "name": "aab-idp",
  "location": "aab-idp/aab-idp.vmx",
  "host_id": "4",
  "created_on": "2018-01-09T19:23:37Z",
  "updated_on": "2018-01-09T19:23:37Z",
  "storage_id": "8",
  ...
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "add_event",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    ...
    {
      "name": "set_miq_server",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "scan",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "https://172.16.30.129/api/vms/173"
    }
  ]
}
```

Here is an example of a VM with the additional custom buttons *analyze* and *ping* defined:

```
{
  "href": "https://172.16.30.129/api/vms/173",
  "id": "173",
  "vendor": "vmware",
  "name": "aab-ldap",
  "location": "aab-ldap/aab-ldap.vmx",
  "host_id": "4",
  "created_on": "2018-01-09T20:23:37Z",
  "updated_on": "2018-01-09T20:23:37Z",
  "storage_id": "8",
  "guid": "dee6d969-7c69-44b6-bcd3-3b14f21702c5",
  "ems_id": "1",
  "uid_ems": "420c8f0d-4a8f-0b6d-fc3a-0c3cfab7b33f",
  "tools_status": "toolsNotRunning",
  "standby_action": "checkpoint",
  "power_state": "off",
  ...
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "add_event",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    ...
    {
      "name": "set_miq_server",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "scan",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "analyze",
```

```
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "ping",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    }
  ]
}
```

Here, the *analyze* and *ping* are custom buttons for the VM and are returned after the common actions.

For example, to request the *analyze* custom action shown above, this is simply a POST action as follows:

Request:

```
POST /api/vms/173

{
  "action" : "analyze"
}
```

Response:

Responses from custom actions are the normal action responses from the REST API:

```
{
  "success": true,
  "message": "Invoked custom action analyze for VM id: 173",
  "href": "http://localhost:3000/api/vms/173"
}
```


- Custom Attributes
  - Querying Custom Attributes
  - Specifying Attribute Actions
  - Custom Attribute Actions
  - Adding Custom Attributes
  - Editing Custom Attributes
  - Deleting Custom Attributes

Custom Attributes

Custom Attributes can be managed for several types of resources. This is done via the custom_attributes sub-collection as follows:

Resource Type	Custom Attribute Subcollection
Virtual Machines	/api/vms/:id/custom_attributes
Cloud Instances	/api/instances/:id/custom_attributes
Providers	/api/providers/:id/custom_attributes
Services	/api/services/:id/custom_attributes

- Querying Custom Attributes
- Specifying Custom Attributes
- Custom Attribute Actions
- Adding Custom Attributes
- Editing Custom Attributes
- Deleting Custom Attributes

Querying Custom Attributes

Custom attributes of a resource can be asked for by expanding the subcollection as follows:

```
GET /api/vms/:id?expand=custom_attributes
GET /api/instances/:id?expand=custom_attributes
GET /api/providers/:id?expand=custom_attributes
GET /api/services/:id?expand=custom_attributes
```

Optionally, just the custom attributes of a resource can be queried:

```
GET /api/vms/:id/custom_attributes
GET /api/instances/:id/custom_attributes
GET /api/providers/:id/custom_attributes
GET /api/services/:id/custom_attributes
```

Specifying Attribute Actions

Custom attributes are specified as follows:

```
{
  "name" : "...",
  "value" : "...",
  "field_type" : "...",
  "source" : "...",
  "section" : "...",
}
```

For adding custom attributes, **name** field is required.

field_type, **source** and **section** are optional.

There is no default value for **field_type**. Allowed values for **field_type** are **DateTime**, **Date** and **Time**.

Default value for **source** is **EVM**. If **source** is set to **VC**, the custom attribute field will be updated on the provider.

Default value for **section** is **metadata**. Allowed values for **section** include **metadata** and **cluster_settings**.

For editing a custom attribute, specify the custom attribute either by **name**:

```
{
  "name" : "...",
  "value" : "..."
}
```

Or by **href**:

```
{
  "href" : "http://localhost:3000/api/vms/:id/custom_attributes/:id",
  "value" : "..."
}
```

Custom Attribute Actions

Action	Description
add	Adding Custom Attributes to a Resource
edit	Editing Custom Attributes of a Resource
delete	Deleting Custom Attributes from a Resource

Adding Custom Attributes

Adding custom attributes to a Resource:

```
POST /api/vms/:id/custom_attributes
POST /api/instances/:id/custom_attributes
POST /api/providers/:id/custom_attributes
POST /api/services/:id/custom_attributes
```

```
{
  "action" : "add",
  "resources" : [
    { ... },
    { ... },
    ...
  ]
}
```

For example, adding two custom attributes:

```
{
  "action" : "add",
  "resources" : [
    { "name" : "ca1", "value" : "value 1" },
    { "name" : "ca2", "value" : "value 2" }
  ]
}
```

Editing Custom Attributes

Editing custom attributes of a Resource:

```
POST /api/vms/:id/custom_attributes
POST /api/instances/:id/custom_attributes
POST /api/providers/:id/custom_attributes
POST /api/services/:id/custom_attributes
```

```
{
  "action" : "edit",
  "resources" : [
    { ... },
    { ... },
    ...
  ]
}
```

For example:

```
{
  "action" : "edit",
```

```
  "resources" : [
    { "name" : "ca_name1", "value" : "updated custom attribute value 1" },
    { "name" : "ca_name2", "value" : "updated custom attribute value 2" }
  ]
}
```

Deleting Custom Attributes

Deleting custom attributes of a Resource:

```
POST /api/vms/:id/custom_attributes
POST /api/instances/:id/custom_attributes
POST /api/providers/:id/custom_attributes
POST /api/services/:id/custom_attributes

{
  "action" : "delete",
  "resources" : [
    { ... },
    { ... },
    ...
  ]
}
```

In the following example, a request to delete custom attributes of a VM by name and by href:

```
{
  "action" : "delete",
  "resources" : [
    { "name" : "ca_name1" },
    { "href" : "http://localhost:3000/api/vms/:id/custom_attributes/:id" }
  ]
}
```

Custom attributes can be deleted via a *delete* action as follows:

```
POST /api/vms/:id/custom_attributes/:ca_id
POST /api/instances/:id/custom_attributes/:ca_id
POST /api/providers/:id/custom_attributes/:ca_id
POST /api/services/:id/custom_attributes/:ca_id

{
  "action" : "delete"
}
```

They can also be deleted via the DELETE method:

```
DELETE /api/vms/:id/custom_attributes/:ca_id
DELETE /api/instances/:id/custom_attributes/:ca_id
DELETE /api/providers/:id/custom_attributes/:ca_id
DELETE /api/services/:id/custom_attributes/:ca_id
```


- Custom Button Management
  - Managing Custom Button Sets
    - Querying Custom Button Sets
    - Creating Custom Button Sets
    - Editing Custom Button Sets
    - Deleting Custom Button Sets
  - Managing Custom Buttons
    - Querying Custom Buttons
    - Creating Custom Buttons
    - Editing Custom Buttons
    - Deleting Custom Buttons

Custom Button Management

Management of Custom Buttons is provided with the following collections:

```
/api/custom_button_sets
/api/custom_buttons
```


- Managing Custom Button Sets
  - Querying Custom Button Sets
  - Creating Custom Button Sets
  - Editing Custom Button Sets
  - Deleting Custom Button Sets
- Managing Custom Buttons
  - Querying Custom Buttons
  - Creating Custom Buttons

- Editing Custom Buttons
- Deleting Custom Buttons

Managing Custom Button Sets

Querying Custom Button Sets

Querying all available Custom Button Sets:

```
GET /api/custom_button_sets
```

Getting details on a specific Custom Button Set:

```
GET /api/custom_button_sets/:id
```

Creating Custom Button Sets

Custom Button Sets can be created as follows:

```
POST /api/custom_button_sets
```

```
{
  "name"        : "Generic Object Custom Button Group",
  "description" : "Generic Object Custom Button Group Description",
  "set_data"    : {
    "button_icon"      : "ff ff-view-expanded",
    "button_color"     : "#4727ff",
    "display"          : true,
    "applies_to_class" : "GenericObjectDefinition",
    "applies_to_id"    : "50"
  }
}
```

Editing Custom Button Sets

Updating Custom Button Sets can be done via the *edit* action as follows:

```
POST /api/custom_button_sets/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "Updated Custom Button Group",
    "description" : "Updated Custom Button Group Description",
    "set_data" : {
      "button_icon" : "ff ff-closed"
    }
  }
}
```

Custom Button Sets can also be updated in bulk:

```
POST /api/custom_button_sets
```

```
{
  "action" : "edit",
  "resources" : [
    { "id" : "11", "set_data" : { "button_icon" : "ff ff-closed" } },
    { "id" : "12", "set_data" : { "button_icon" : "ff ff-closed" } },
    { "id" : "13", "set_data" : { "button_icon" : "ff ff-closed" } }
  ]
}
```

Custom button sets can also be updated with the PUT method as follows:

```
PUT /api/custom_button_sets/:id
```

```
{
  "name" : "Updated Custom Button Group",
  "description" : "Updated Custom Button Group Description",
  "set_data" : {
    "button_icon" : "ff ff-closed"
  }
}
```

Deleting Custom Button Sets

Custom Button Sets can be deleted via the *delete* action as follows:

```
POST /api/custom_button_sets/:id
```

```
{
  "action" : "delete"
}
```

and in bulk as follows:

```
POST /api/custom_button_sets
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/custom_button_sets/21" },
    { "href" : "http://localhost:3000/api/custom_button_sets/22" },
    { "id" : "23" }
  ]
}
```

A single Custom Button Set can also be deleted with the DELETE method as follows:

```
DELETE /api/custom_button_sets/:id
```

Managing Custom Buttons

Querying Custom Buttons

Querying all available Custom Buttons:

```
GET /api/custom_buttons
```

Getting details on a specific Custom Button:

```
GET /api/custom_buttons/:id
```

Creating Custom Buttons

Custom Buttons can be created as follows:

```
POST /api/custom_buttons
```

```
{
  "name"              : "Generic Object Custom Button",
  "description"       : "Generic Object Custom Button Description",
  "applies_to_class" : "GenericObjectDefinition",
  "options"           : {
    "button_icon"     : "ff ff-view-expanded",
    "button_color"    : "#4727ff",
    "display"         : true
  },
  "resource_action"   : {
    "ae_namespace"    : "SYSTEM",
    "ae_class"        : "PROCESS",
    "dialog_id"       : "1"
  },
  "visibility"        : {
    "roles"           : ["_ALL_"]
  }
}
```

Editing Custom Buttons

Updating Custom Buttons can be done via the *edit* action as follows:

```
POST /api/custom_buttons/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "Updated Custom Button",
    "description" : "Updated Custom Button Description",
    "resource_action" : {
      "ae_namespace"  : "SYSTEM2"
    }
  }
}
```

Custom Buttons can also be updated in Bulk as follows:

```
POST /api/custom_buttons
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/custom_buttons/31",
      "options" :  { "display" : false }
    },
    {
      "href" : "http://localhost:3000/api/custom_buttons/32",
      "options" :  { "display" : false }
    },
    {
      "id"   : "33",
      "options" :  { "display" : false }
    }
  ]
}
```

Deleting Custom Buttons

Custom Buttons can be deleted via the *delete* action as follows:

```
POST /api/custom_buttons/:id
```

```
{
  "action" : "delete"
}
```

Multiple Custom Buttons can also be deleted in a single request as follows:

```
POST /api/custom_buttons
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/custom_buttons/21" },
    { "id" : "22" },
    { "id" : "23" }
  ]
}
```

Single Custom Buttons can also be deleted with the DELETE method as follows:

```
DELETE /api/custom_buttons/:id
```


- Event Streams
  - Filtering by event type
  - Filtering by timestamp
  - Filtering by target type
  - Filtering by target type and id

Event Streams

Querying of Event Streams is made available via the following collection:

```
/api/event_streams
```

Event Streams can be queried as follows:

```
GET /api/event_streams
```

An individual Event Stream can also be fetched as follows:

```
GET /api/event_streams/:id
```

Relevant filter[] parameters when Event Streams are queried:

Parameter	Examples
event_type	event_type=vm_start, event_type=vm_stop
timestamp	timestamp>2017-01-03, timestamp<2017-12-31
target_type	target_type=VmOrTemplate
target_id	target_id=11

Filtering by event type

```
GET /api/event_streams?filter[]=event_type=vm_start
```

Filtering by timestamp

```
GET /api/event_streams?filter[]=timestamp>2017-01-03
                      &filter[]=timestamp<2017-01-10
```

Filtering by target type

```
GET /api/event_streams?filter[]=target_type=VmOrTemplate
```

Filtering by target type and id

```
GET /api/event_streams?filter[]=target_type=VmOrTemplate
                      &filter[]=target_id=21
```


- Flavor Management
  - Querying Flavors
  - Managing Flavors
  - Create Flavors
  - Delete Flavors

Flavor Management

Management of Flavors is available via the following endpoint:

`/api/flavors`

Querying Flavors

Querying all available Flavors:

```
GET /api/flavors
```

Getting details on a specific Flavor:

```
GET /api/flavors/:id
```

Querying the Flavors for a specific Provider:

```
GET /api/providers/:id/flavors
```

Getting details on a specific Flavor of a Provider:

```
GET /api/providers/:id/flavors/:flavor_id
```

Managing Flavors

The following actions are available on Flavors of Providers:

Action	Description
create	Create Flavors
delete	Delete Flavors

Create Flavors

Flavors can be created directly for a Provider as follows:

```
POST /api/providers/:id/flavors
```

```
{
  "name" : "base-system",
  "cpu_cores" : 2,
  "memory" :  4096000000,
  "root_disk_size" : 16384000000
}
```

Delete Flavors

Flavors of a Provider can be deleted via the *delete* action:

```
POST /api/providers/:id/flavors/:flavor_id
```

```
{
  "action" : "delete"
}
```

Multiple Flavors can also be deleted in a single request as follows:

```
POST /api/providers/:id/flavors
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/flavors/6" },
    { "id" : "7" }
  ]
}
```

Single Flavors can also be deleted with the DELETE HTTP method as follows:

```
DELETE /api/providers/:id/flavors/:flavor_id
```


- Generic Object Management
  - Generic Object Definitions
    - Querying Generic Object Definitions
    - Creating Generic Object Definitions
    - Editing Generic Object Definitions
      - Add Attributes
      - Remove Attributes
      - Add Methods
      - Remove Methods
      - Add Associations
      - Remove Associations
    - Deleting Generic Object Definitions
  - Generic Objects
    - Querying Generic Objects
      - Request:
      - Response:
    - Creating Generic Objects
    - Editing Generic Objects

- Calling Generic Object Methods
- Deleting Generic Objects

Generic Object Management

Management of Generic Objects is provided with the following collections:

```
/api/generic_object_definitions
/api/generic_objects
```


- Managing Generic Object Definitions
  - Querying Generic Object Definitions
  - Creating Generic Object Definitions
  - Editing Generic Object Definitions
  - Deleting Generic Object Definitions
- Managing Generic Objects
  - Querying Generic Objects
  - Creating Generic Objects
  - Editing Generic Objects
  - Calling Generic Object Methods
  - Deleting Generic Objects

Generic Object Definitions

Querying Generic Object Definitions

Querying all available Generic Object Definitions:

```
GET /api/generic_object_definitions
```

Getting details on a specific Generic Object Definition:

```
GET /api/generic_object_definitions/:id
```

Getting details on a specific Generic Object Definition by name:

```
GET /api/generic_object_definitions/:name
```

Querying all the Generic Objects defined for a specific Generic Object Definition:

```
GET /api/generic_object_definitions/:id/generic_objects
```

Getting a specific Generic Object defined for a Generic Object Definition:

```
GET /api/generic_object_definitions/:id/generic_objects/:generic_object_id
```

Fetching the vms associations for a specific Generic Object:

```
GET /api/generic_object_definitions/:id/generic_objects/:generic_object_id?associations=vms
```

Creating Generic Object Definitions

Generic Object Definitions can be created as follows:

```
POST /api/generic_object_definitions
```

```
{
  "name"        : "LoadBalancer",
  "description" : "LoadBalancer description",
  "properties"  : {
    "attributes"  : {
      "address"      : "string",
      "last_restart" : "datetime"
    },
    "associations" : {
      "vms"      : "Vm",
      "services" : "Service"
    },
    "methods" : [
      "add_vm",
      "remove_vm"
    ]
  },
  "picture"     : {
    "extension" : "png",
    "content"   : "iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAfFcSJAAAABGdBTUEAALGP"\
      "C/xhBQAAACBjSFJNAAB6JgAAgIQAAPoAAACA6AAAdTAAAOpgAAA6mAAAF3Cc"\
```

```
      "ulE8AAAACXBIWXMAAAsTAAALEwEAmpwYAAABWWlUWHRYTUw6Y29tLmFkb2Jl"\
      "LnhtcAAAAAAAPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyIg"\
      "eDp4bXB0az0iWE1QIENvcmUgNS40LjAiPgogICA8cmRmOlJERiB4bWxuczpy"\
      "ZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1u"\
      "cyMiPgogICAgICA8cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0iIgogICAg"\
      "ICAgICAgICB4bWxuczp0aWZmPSJodHRwOi8vbnMuYWRvYmUuY29tL3RpZmYv"\
      "MS4wLyI+CiAgICAgICAgICAgIDx0aWZmOk9yaWVudGF0aW9uPjE8L3RpZmY6T3Jp"\
      "ZW50YXRpb24+CiAgICAgICAgICAgIDwvcmRmOkRlc2NyaXB0aW9uPgogICA8L3JkZjpS"\
      "REY+CjwveDp4bXBtZXRhPgpMwidZAAAADUlEQVQIHWNgYGCwBQAAQgA+3N0+"\
      "xQAAAABJRU5ErkJggg=="
  }
}
```

Example of creating a Generic Object Definition with a picture reference:

```
{
  "name"        : "LoadBalancer",
  "description" : "LoadBalancer description",
  "properties"  : {
    "attributes"  : {
      "address"      : "string",
      "last_restart" : "datetime"
    },
    "associations" : {
      "vms"      : "Vm",
      "services" : "Service"
    },
    "methods" : [
      "add_vm",
      "remove_vm"
    ]
  },
  "picture"     : {
    "href" : "http://localhost:3000/api/pictures/11"
  }
}
```

Note: picture references can be by *href* or *id*.

Editing Generic Object Definitions

Updating Generic Object Definitions can be done via the *edit* action as follows:

```
POST /api/generic_object_definitions/:id
```

or by name:

```
POST /api/generic_object_definitions/:name
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Latest LoadBalancer Description"
  }
}
```

Generic object definitions can also be updated with the PUT method as follows:

```
PUT /api/generic_object_definitions/:id
```

```
{
  "name" : "New LoadBalancer Name",
  "description" : "New LoadBalancer Description"
}
```

Pictures can be removed from Generic Object Definitions by sending in an empty picture reference via any of the edit methods, for example:

```
POST /api/generic_object_definitions/:name
```

```
{
  "action" : "edit",
  "resource" : {
    "picture" : {}
  }
}
```

Generic Object Definitions can be edited by href, id or name in bulk as in the following example:

```
POST /api/generic_object_definitions
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "name" : "Load Balancer",
      "resource" : {
        "name" : "Enhanced Load Balancer",
        "description" : "Description for Enhanced Load Balancer",
        "picture" : { "id" : "16" }
      }
    },
```

```
    {
      "href" : "http://localhost:3000/api/generic_object_definitions/12",
      "resource" : {
        "name" : "NetworkScanner",
        "description" : "Description for Network Scanner"
      }
    },
    {
      "id" : "13",
      "resource" : {
        "name" : "NetworkFirewall",
        "description" : "Description for Network Firewall"
      }
    }
  ]
}
```

Editing Generic Object Definitions can also be done by using one of the following actions for updating *attributes*, *methods* and *associations*:

- add_attributes
- remove_attributes
- add_methods
- remove_methods
- add_associations
- remove_associations

Add Attributes

```
POST /api/generic_object_definitions/:id

{
  "action" : "add_attributes",
  "resource" : {
    "attributes" : {
      "optional_attribute1" : "string",
      "optional_attribute2" : "boolean"
    }
  }
}
```

Remove Attributes

```
POST /api/generic_object_definitions/:id

{
  "action" : "remove_attributes",
  "resource" : {
    "attributes" : {
      "unused_attribute9" : "string"
    }
  }
}
```

Add Methods

```
POST /api/generic_object_definitions/:id

{
  "action" : "add_methods",
  "resource" : {
    "methods" : [ "go_method1", "go_method2" ]
  }
}
```

Remove Methods

```
POST /api/generic_object_definitions/:id

{
  "action" : "remove_methods",
  "resource" : {
    "methods" : [ "go_method3" ]
  }
}
```

Add Associations

```
POST /api/generic_object_definitions/:id

{
  "action" : "add_associations",
  "resource" : {
    "associations" : {
      "az" : "AvailabilityZone",
```

```
      "chargeback" : "ChargebackVm"
    }
  }
}
```

Remove Associations

```
POST /api/generic_object_definitions/:id
```

```
{
  "action" : "remove_associations",
  "resource" : {
    "associations" : {
      "chargeback" : "ChargebackVm"
    }
  }
}
```

Deleting Generic Object Definitions

Generic Object Definitions can be deleted via the *delete* action as follows:

```
POST /api/generic_object_definitions/:id
```

or by name:

```
POST /api/generic_object_definitions/:name
```

```
{
  "action" : "delete"
}
```

Multiple Generic Object Definitions can also be deleted in a single request as follows:

```
POST /api/generic_object_definitions
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/generic_object_definitions/11" },
    { "id" : "12" }
  ]
}
```

Single Generic Object Definitions can also be deleted with the DELETE method as follows:

```
DELETE /api/generic_object_definitions/:id
```

or by name:

```
DELETE /api/generic_object_definitions/:name
```

Generic Objects

Querying Generic Objects

Querying all available Generic Objects:

```
GET /api/generic_objects
```

Getting details on a specific Generic Object:

```
GET /api/generic_objects/:id
```

Querying all the Generic Objects defined for a specific Service:

```
GET /api/services/:id/generic_objects
```

Querying Generic Objects and related picture and image_href for a specific Service:

Request:

```
GET /api/services/:id?expand=generic_objects&attributes=generic_objects.picture
```

Response:

```
{
  "href" : "http://localhost:3000/api/services/645",
  "id" : "645",
  "name" : "Test Azure Service",
  "description" : "Azure",
  "guid" : "6ce8f6c2-7f93-4a7e-b618-da9acab470ac",
  "options": {  },
  "display": false,
  "created_at" : "2017-08-07T16:52:46Z",
  "updated_at" : "2017-08-07T16:52:46Z",
  "miq_group_id" : "1",
  "retired": false,
  "tenant_id" : "1",
```

```
  "initiator" : "user",
  "generic_objects": [
    {
      "href" : "http://localhost:3000/api/services/645/generic_objects/1",
      "id" : "1",
      "name" : "generic_object_1",
      "generic_object_definition_id" : "1",
      "created_at" : "2017-08-15T17:35:08Z",
      "updated_at" : "2017-09-07T12:05:14Z",
      "properties": {
         "vms" : [ 615 ],
         "start" : true
      },
      "picture" : {
        "href" : "http://localhost:3000/api/picture/1",
        "id" : "1",
        "resource_id" : "1",
        "resource_type" : "GenericObjectDefinition",
        "image_href" : "http://localhost:3000/pictures/1.png",
        "extension" : "png"
      }
    }
  ],
  "actions" : {
    ...
  }
}
```

Creating Generic Objects

Generic Objects can be created as follows:

```
POST /api/generic_objects

{
  "generic_object_definition" : { "href" : "http://localhost:3000/api/generic_object_definitions/11" },
  "name" : "go1_name",
  "uid" : "optional_uid",
  "property_attributes" : {
    "widget" : "widget value",
    "supports_feature_a" : false
  },
  "associations" : {
    "vms" : [
      { "href" : "http://localhost:3000/api/vms/11" },
      { "id" : "12 }
    ]
  }
}
```

Generic Objects can also be directly created for a specific Generic Object Definition type as follows:

```
POST /api/generic_object_definitions/:id/generic_objects

{
  "name" : "go1_name",
  "uid" : "optional_uid",
  "property_attributes" : {
    "widget" : "widget value",
    "supports_feature_a" : false
  },
  "associations" : {
    "vms" : [
      { "href" : "http://localhost:3000/api/vms/11" },
      { "id" : "12 }
    ]
  }
}
```

In which case, the generic_object_definition reference is not needed.

Editing Generic Objects

Updating Generic Objects can be done via the *edit* action as follows:

```
POST /api/generic_objects/:id

{
  "action" : "edit",
  "resource" : {
    "name" : "updated_name",
    "property_attributes" : {
      "widget" : "updated_widget"
    }
  }
}
```

Calling Generic Object Methods

When generic objects have methods defined in their definitions, those methods are exposed with the generic objects as actions.

The following example shows a generic object with two methods *method_a* and *method_b*:

```
GET /api/generic_objects/101
```

```
{
  "href" : "http://localhost:3000/api/generic_objects/101",
  "id"   : "101",
  "name" : "sample_object",
  "generic_object_definition_id" : "5",
  "property_attributes" : {
    "widget" : "a sample widget",
    "object_flag" : true
  },
  "actions" : [
    {
      "name" : "edit",
      "method" : "post",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "delete",
      "method" : "post",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "delete",
      "method" : "delete",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "method_a",
      "method" : "post",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "method_b",
      "method" : "post",
      "href" : "http://localhost:3000/api/generic_objects/101"
    }
  ]
}
```

With the above sample generic object, the custom methods can be invoked like any other action as follows:

```
POST /api/generic_objects/101
```

```
{
  "action" : "method_a",
  "resource" : {
    "attr1" : "value 1",
    "attr2" : "value 2"
  }
}
```

Deleting Generic Objects

Generic Objects can be deleted via the *delete* action as follows:

```
POST /api/generic_objects/:id
```

```
{
  "action" : "delete"
}
```

Multiple Generic Objects can also be deleted in a single request as follows:

```
POST /api/generic_objects
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/generic_objects/21" },
    { "id" : "22" },
    { "id" : "23" }
  ]
}
```

Single Generic Objects can also be deleted with the DELETE method as follows:

```
DELETE /api/generic_objects/:id
```


- Group Management
  - Querying Groups
  - Creating Groups
  - Editing Groups
  - Deleting Groups

Group Management

Management of User Groups is provided via the following collection:

```
/api/groups
```

Full CRUD actions on groups is available:

- Querying Groups
- Creating Groups
- Editing Groups
- Deleting Groups

Querying Groups

Querying all groups in the system:

```
GET /api/groups
```

Getting details on a specific group:

```
GET /api/groups/:id
```

Creating Groups

Groups can be created via a POST to the groups collection or via the **create** action signature which also allows creation of multiple groups in a single request.

```
POST /api/groups
```

```
{
  "description" : "sample_group",
  "role" : { "id" : "2" },
  "tenant" : { "href" : "http://localhost:3000/api/tenants/1" },
  "filters" : {
    "belongsto" : [ "/managed/area/1", "/managed/area/2" ],
    "managed" : [[ "/managed/infra/1", "/managed/infra/2" ]]
  }
}
```

Note:

role can be specified via **name**, **href**, or **id**

tenant can be specified via **href** or **id**

or creating multiple groups:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "sample_group1", ... },
    { "description" : "sample_group2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Groups.

Editing Groups

```
POST /api/groups/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "updated_sample_group",
    "filters" : {
      "belongsto" : [ "/managed/area/1", "/managed/area/2", "/managed/area/3" ],
      "managed" : [[ "/managed/infra/1", "/managed/infra/2"], ["/managed/other/3"]]
    }
  }
}
```

or editing multiple groups:

```
POST /api/groups
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/groups/101",
      "description" : "updated_sample_group1"
    },
    {
      "href" : "http://localhost:3000/api/groups/102",
      "description" : "updated_sample_group2"
```

```
    },
    ...
  ]
}
```

Deleting Groups

groups can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/groups/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/groups/101
```

Deleting multiple groups can be done as follows:

```
POST /api/groups

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/groups/101" },
    { "href" : "http://localhost:3000/api/groups/102" },
    ...
  ]
}
```


- Host Management
  - Querying Hosts
  - Editing a Host

Host Management

Management of Hosts adds support of the following actions:

Action	Description
edit	Edit a Host

- Querying Hosts
- Editing a Host

Querying Hosts

Hosts are queried via the primary collection URL:

```
/api/hosts
```

Editing a Host

Currently, only the host password can be edited via the API. All other edit requests will be rejected.

The host password is provided by a *credentials* hash similar to the signature supported for providers.

```
POST /api/hosts/:id

{
  "action" : "edit",
  "resource" : {
    "credentials" : {
      "password" : "updated_password"
    }
  }
}
```

When not specified, the authentication type used for the password is *default*:

```
POST /api/hosts/:id

{
  "action" : "edit",
  "resource" : {
    "credentials" : {
      "auth_type" : "default",
      "password" : "updated_password"
    }
  }
}
```

Password for multiple hosts can also be updated with a single request as follows:

```
POST /api/hosts
```

```
{
  "action" : "edit",
  "resources" : [
    { "id" : "101", "credentials" : { "password" : "updated_password" } },
    { "id" : "102", "credentials" : { "password" : "updated_password" } },
    { "id" : "103", "credentials" : { "password" : "updated_password" } },
    ...
  ]
}
```

Note:

host can be referenced by **href** or **id**

- Instance Management
  - Targeting Instances
  - Querying Instances
    - Load Balancers
  - Starting an Instance
  - Stopping an Instance
  - Suspending an Instance
  - Pausing an Instance
  - Terminating an Instance
  - Shelving an Instance
  - Rebooting Guest in an Instance
  - Resetting an Instance

Instance Management

Management of instances adds support for the following actions:

Action	Description
start	Starts an Instance
stop	Stops an Instance
suspend	Suspending an Instance
pause	Pausing an Instance
reboot_guest	Rebooting Guest in an Instance
reset	Resetting an Instance
shelve	Shelving an Instance
terminate	Terminating an Instance

- Targeting Instances
- Querying Instances
- Starting an Instance
- Stopping an Instance
- Suspending an Instance
- Pausing an Instance
- Rebooting Guest in an Instance
- Resetting an Instance
- Shelving an Instance
- Terminating an Instance

Targeting Instances

These actions can be triggered on individual instance resources:

```
/api/instances/:id
```

As simply as POSTing the following action to an instance.

```
{
  "action" : "start"
}
```

Requests can also be made on multiple instances by targeting the primary collection:

```
/api/instances
```

```
{
  "action" : "start",
  "resources" : [
    { "href" : "http://localhost:3000/api/instances/101" },
```

```
    { "href" : "http://localhost:3000/api/instances/102" },
    ...
  ]
}
```

Querying Instances

Instances are queried via the primary collection URL:

```
/api/instances
```

Filtering, sorting and paging as mentioned on the Querying page.

Load Balancers

Load Balancers of an instance can be queried via the load_balancers subcollection as follows:

Query all Load Balancers of an instance:

```
GET /api/instances/:id/load_balancers
```

Query a specific Load Balancer of an instance:

```
GET /api/instances/:id/load_balancers/:load_balancer_id
```

Or expanding the Load Balancers for the specific instance as follows:

```
GET /api/instances/:id?expand=load_balancers
```

Starting an Instance

```
{
  "action" : "start"
}
```

Stopping an Instance

```
{
  "action" : "stop"
}
```

Suspending an Instance

```
{
  "action" : "suspend"
}
```

Pausing an Instance

```
{
  "action" : "pause"
}
```

Terminating an Instance

```
{
  "action" : "terminate"
}
```

Shelving an Instance

```
{
  "action" : "shelve"
}
```

Rebooting Guest in an Instance

```
{
  "action" : "reboot-guest"
}
```

Resetting an Instance

```
{
  "action" : "reset"
}
```


- Metric Rollups
  - Getting 60 Daily Metric Rollups
  - Getting Hourly Metric Rollups for a range of Dates
  - Getting Hourly Metric Rollups for a Service
  - Getting 30 Daily Metric Rollups for a Vm

Metric Rollups

Querying of Metric Rollups is made available via the following collection:

```
/api/metric_rollups
```

Supported query parameters:

Parameter	Examples
resource_type	VmOrTemplate, Service or any resource type that supports Metric Rollups
resource_ids[]	id1
resource_ids[]	id2
resource_ids[]	...
capture_interval	hourly, daily
start_date	2018-01-10, 2018-01-10T11:20:00Z
end_date	2018-01-17, 2017-12-31T23:59:00Z

Getting 60 Daily Metric Rollups

```
GET /api/metric_rollups?resource_type=VmOrTemplate
                       &resource_ids[]=101
                       &resource_ids[]=102
                       &resource_ids[]=103
                       &capture_interval=daily
                       &start_date='2017-10-01'
                       &limit=60
```

Getting Hourly Metric Rollups for a range of Dates

```
GET /api/metric_rollups?resource_type=Service
                       &resource_ids[]=55
                       &resource_ids[]=56
                       &capture_interval=hourly
                       &start_date='2017-11-15'
                       &end_date='2017-11-18'
```

Metric rollups can also be obtained for a specific resource as a subcollection:

```
/api/vms/:id/metric_rollups
```

```
/api/services/:id/metric_rollups
```

Getting Hourly Metric Rollups for a Service

```
GET /api/services/57/metric_rollups?capture_interval=hourly&start_date='2017-06-15'
```

Getting 30 Daily Metric Rollups for a Vm

```
GET /api/vms/104/metric_rollups?capture_interval=daily&start_date='2017-01-01'&limit=30
```


- Network Routers
  - Querying Network Routers
  - Deleting Network Routers
    - Request:
    - Response:

Network Routers

Management of Network Routers is provided via the following collection:

```
/api/network_routers
```

The following actions are available on Network Routers:

- Querying Network Routers
- Deleting Network Routers

Querying Network Routers

Querying all Network Routers in the system is simply:

```
GET /api/network_routers
```

Getting details on a specific Network Router:

```
GET /api/network_routers/:id
```

Deleting Network Routers

Network Routers can be deleted via either the **delete** POST action or via the DELETE HTTP method.

Request:

```
POST /api/network_routers/101
```

```
{
  "action" : "delete"
}
```

Response:

```
{
  "success" : true,
  "message" : "Deleting Networking Router id:101 name: 'test_network_router'",
  "task_id" : "33",
  "task_href" : "http://localhost:3000/api/tasks/33"
}
```

or simply:

```
DELETE /api/network_routers/101
```

Deleting multiple Network Routers can be done as follows:

```
POST /api/network_routers
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/network_routers/101" },
    { "href" : "http://localhost:3000/api/network_routers/102" },
    ...
  ]
}
```


- Notifications Support
- Querying Notifications
- Marking Notifications as Seen
  - Deleting Notifications

Notifications Support

Support for Notifications to the authenticated user is provided via the following collection:

```
/api/notifications
```


- Querying Notifications
- Marking Notifications as Seen
- Deleting Notifications

Querying Notifications

The following fetches the current notifications for the authenticated user:

```
GET /api/notifications
```

```
{
  "name": "notifications",
  "count": 3,
  "subcount": 3,
  "resources": [
    {
      "href": "http://localhost:3000/api/notifications/44"
    },
    {
      "href": "http://localhost:3000/api/notifications/45"
    },
    {
      "href": "http://localhost:3000/api/notifications/46"
    }
  ],
  "actions": [
    {
      "name": "mark_as_seen",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    }
  ]
}
```

Or fetching additional details on the notifications as follows:

```
GET /api/notifications?expand=resources&attributes=details
```

```
{
  "name": "notifications",
  "count": 3,
  "subcount": 3,
  "resources": [
    {
      "href": "http://localhost:3000/api/notifications/44",
      "id": "44",
      "user_id": "1",
      "seen": false,
      "details": {
        "level": "info",
        "created_at": "2016-11-03T17:23:29Z",
        "text": "Vm [aab-test] was powered on",
        "bindings": {
          "initiator": {
            "link": {
              "id": "1",
              "model": "User"
            },
            "text": "Administrator"
          }
        },
        "seen": false
      },
      "actions": [
        {
          "name": "mark_as_seen",
          "method": "post",
          "href": "http://localhost:3000/api/notifications/44"
        },
        {
          "name": "delete",
          "method": "post",
          "href": "http://localhost:3000/api/notifications/44"
        },
        {
          "name": "delete",
          "method": "delete",
          "href": "http://localhost:3000/api/notifications/44"
        }
      ]
    },
    {
      "href": "http://localhost:3000/api/notifications/45",
      "id": "45",
      "notification_id": "22",
      "user_id": "1",
      "seen": true,
      "details": {
        "level": "info",
        "created_at": "2016-11-03T17:23:29Z",
        "text": "Vm [aab-test2] was powered on",
        "bindings": {
          "initiator": {
            "link": {
              "id": "1",
              "model": "User"
            },
            "text": "Administrator"
          }
        },
        "seen": true
      },
      "actions": [
        {
          "name": "mark_as_seen",
          "method": "post",
          "href": "http://localhost:3000/api/notifications/45"
        },
        {
          "name": "delete",
          "method": "post",
          "href": "http://localhost:3000/api/notifications/45"
        },
        {
          "name": "delete",
          "method": "delete",
          "href": "http://localhost:3000/api/notifications/45"
        }
      ]
    },
    {
      "href": "http://localhost:3000/api/notifications/46",
      "id": "46",
      "notification_id": "23",
      "user_id": "1",
      "seen": false,
      "details": {
        "level": "info",
        "created_at": "2016-11-03T17:23:54Z",
```

```
        "text": "Vm [aab-dev] was powered off",
        "bindings": {
          "initiator": {
            "link": {
              "id": "1",
              "model": "User"
            },
            "text": "Administrator"
          }
        },
        "seen": false
      },
      "actions": [
        {
          "name": "mark_as_seen",
          "method": "post",
          "href": "http://localhost:3000/api/notifications/46"
        },
        {
          "name": "delete",
          "method": "post",
          "href": "http://localhost:3000/api/notifications/46"
        },
        {
          "name": "delete",
          "method": "delete",
          "href": "http://localhost:3000/api/notifications/46"
        }
      ]
    }
  ],
  "actions": [
    {
      "name": "mark_as_seen",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    }
  ]
}
```

Marking Notifications as Seen

Marking Notifications as seen for the authenticated user is done via the **mark_as_seen** action as follows:

```
POST /api/notifications/:id

{
  "action" : "mark_as_seen"
}
```

Or, marking several notifications as seen in a single call as follows:

```
POST /api/notifications

{
  "action" : "mark_as_seen",
  "resources" : [
    { "href" : "http://localhost:3000/api/notifications/44" },
    { "href" : "http://localhost:3000/api/notifications/45" },
    ...
  ]
}
```

Deleting Notifications

Deleting a notification is available via the **delete** action or DELETE on the individual notification as follows:

```
POST /api/notifications/:id

{
  "action" : "delete"
}
```

Or simply:

```
DELETE /api/notifications/:id
```

Also, deleting multiple notifications is available via the **delete** action on the collection passing in references to the different notifications:

```
POST /api/notifications

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/notifications/44" },
```

```
    { "href" : "http://localhost:3000/api/notifications/45" },
    ...
  ]
}
```


- Orchestration Support
  - Orchestration Templates ===
  - Querying Orchestration Templates
  - Creating Orchestration Templates
  - Editing Orchestration Templates
  - Copying Orchestration Templates
  - Deleting Orchestration Templates

Orchestration Support

Support for orchestration is provided by the following:

- Orchestration Templates

Orchestration Templates ===

Management of orchestration templates is provided with the following collection:

```
/api/orchestration_templates
```

Full CRUD actions on orchestration templates is available:

- Querying Orchestration Templates
- Creating Orchestration Templates
- Editing Orchestration Templates
- Copying Orchestration Templates
- Deleting Orchestration Templates

Querying Orchestration Templates

Querying all orchestration templates in the system:

```
GET /api/orchestration_templates
```

Getting details on a specific orchestration template:

```
GET /api/orchestration_templates/:id
```

Creating Orchestration Templates

Orchestration templates can be created via a POST to the orchestration templates collection or via the **create** action signature which also allows creation of multiple orchestration templates in a single request.

```
POST /api/orchestration_templates
```

```
{
  "name" : "azure-single-vm-from-user-image",
  "type" : "OrchestrationTemplateAzure",
  "description" : "Create a single VM from a user image",
  "content" : " ... ",
  "orderable" : true,
  ...
}
```

or creating multiple orchestration templates:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "sample_orchestration_template1", ... },
    { "description" : "sample_orchestration_template2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating orchestration templates.

Editing Orchestration Templates

```
POST /api/orchestration_templates/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "updated_sample_orchestration_template",
```

```
  }
}
```

or editing multiple orchestration templates:

```
POST /api/orchestration_templates

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/orchestration_templates/101",
      "description" : "updated_sample_orchestration_template1"
    },
    {
      "href" : "http://localhost:3000/api/orchestration_templates/102",
      "description" : "updated_sample_orchestration_template2"
    },
    ...
  ]
}
```

Copying Orchestration Templates

This capability allows orchestration templates to be copied onto new orchestration templates with updated content via the *copy* action as follows:

```
POST /api/orchestration_templates/:id

{
  "action" : "copy",
  "resource" : {
    "content" : "{ 'Description' : 'Copied Test Content' }"
  }
}
```

or copying multiple orchestration templates:

```
POST /api/orchestration_templates

{
  "action" : "copy",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/orchestration_templates/101",
      "content" : "{ 'Description' : 'Copied Test Content 1' }"
    },
    {
      "href" : "http://localhost:3000/api/orchestration_templates/102",
      "content" : "{ 'Description' : 'Copied Test Content 2' }"
    },
    ...
  ]
}
```

Deleting Orchestration Templates

Orchestration templates can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/orchestration_templates/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/orchestration_templates/101
```

Deleting multiple orchestration templates can be done as follows:

```
POST /api/orchestration_templates

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/orchestration_templates/101" },
    { "href" : "http://localhost:3000/api/orchestration_templates/102" },
    ...
  ]
}
```


- Physical Server Management
  - Querying Physical Servers
  - Managing Physical Servers
  - Powering On a Physical Server
  - Powering Off a Physical Server
  - Powering Off Now a Physical Server
  - Refresh a Physical Server
  - Restart a Physical Server

- Restart Now a Physical Server
- Restart a Physical Server to System Setup
- Restart Management Controller
- Blink the Identify LED
- Turn On the Identify LED
- Turn Off the Identify LED

Physical Server Management

Management of Physical Servers is available via the following endpoint:

```
/api/physical_servers
```

Querying Physical Servers

Querying all physical servers in the system is simply:

```
GET /api/physical_servers
```

Getting details on a specific Physical Server:

```
GET /api/physical_servers/:id
```

Managing Physical Servers

The following actions are available on Physical Servers:

Action	Description
power_on	Power On
power_off	Power Off
power_off_now	Power Off Now
refresh_now	Refresh
restart	Restart
restart_now	Restart Now
restart_to_sys_setup	Restart to System Setup
restart_mgmt_controller	Restart Management Controller
blink_loc_led	Blink the Identify LED
turn_on_loc_led	Turn On the Identify LED
turn_off_loc_led	Turn Off the Identify LED

Powering On a Physical Server

```
{
  "action" : "power_on"
}
```

Powering Off a Physical Server

```
{
  "action" : "power_off"
}
```

Powering Off Now a Physical Server

```
{
  "action" : "power_off_now"
}
```

Refresh a Physical Server

```
{
  "action" : "refresh"
}
```

Restart a Physical Server

```
{
  "action" : "restart"
}
```

Restart Now a Physical Server

```
{
  "action" : "restart_now"
}
```

Restart a Physical Server to System Setup

```
{
  "action" : "restart_to_sys_setup"
}
```

Restart Management Controller

```
{
  "action" : "restart_mgmt_controller"
}
```

Blink the Identify LED

```
{
  "action" : "blink_loc_led"
}
```

Turn On the Identify LED

```
{
  "action" : "turn_on_loc_led"
}
```

Turn Off the Identify LED

```
{
  "action" : "turn_off_loc_led"
}
```


- Picture Management
  - Creating a Picture
  - Querying Pictures

Picture Management

Management of Pictures adds support of the following actions:

- Creating a Picture
- Querying Pictures

Creating a Picture

```
POST /api/pictures
```

```
{
  "extension": "<extension of image>",
  "content": "<base 64 encoded image here>"
}
```

Supported extensions include:

Extension
jpg
png
svg

Querying Pictures

Querying pictures like any other collection:

```
GET /api/pictures?expand=resources
```

```
{
  "name": "pictures",
  "count": 5,
  "subcount": 5,
  "resources": [
    {
      "href": "http://localhost:3000/api/pictures/1",
      "id": "1",
      "image_href" : "http://localhost:3000/pictures/1.jpg",
      "extension" : "jpg"
    },
    ...
    {
      "href": "http://localhost:3000/api/pictures/4",
      "id": "4",
      "image_href" : "http://localhost:3000/pictures/4.png",
      "extension" : "png",
      "resource_id": "1",
      "resource_type": "ServiceTemplate"
    },
    {
      "href": "http://localhost:3000/api/pictures/5",
      "id": "5",
      "image_href" : "http://localhost:3000/pictures/5.jpg",
      "extension" : "jpg"
    }
  ],
  "actions": [
```

```
    {
      "name": "create",
      "method": "post",
      "href": "http://localhost:3000/api/pictures"
    }
  ]
}
```

This query does not return the pictures themselves, simply metadata about the pictures.

Actually fetching the particular picture is done by getting the image by its image_href.

```
GET /api/pictures/4
```

```
{
  "href": "http://localhost:3000/api/pictures/4",
  "id": "4",
  "resource_id": "1",
  "resource_type": "ServiceTemplate",
  "image_href": "http://localhost:3000/pictures/4.jpg"
}
```

In which case the image can simply be downloaded with a GET:

```
GET http://localhost:3000/pictures/4.jpg
```

For fetching information about pictures pertaining to services, see the section Querying Pictures for details.

- Policy Management
  - Querying Policies
  - Querying Policies of a Resource
  - Creating Policies
  - Editing Policies
  - Deleting Policies
  - Policy and Policy Profiles Actions
  - Resource Policy Management
    - Policy specification
    - Assigning policies to a resource
      - Response:
    - Unassigning policies from a resource
      - Response:
  - Resolving Policies of a Resource
    - Response:
  - Resolving Policy Profiles of a Resource
    - Response:

Policy Management

Management of policies and policy profiles is provided in the following forms:

- Querying of policies via the primary collection:

```
/api/policies
```

- Querying of policy profiles via the primary collection:

```
/api/policy_profiles
```

- Assigning and Unassigning policies to resources via the policies subcollection of a resource

```
/api/:collection/:id/policies
```

and posting **assign** or **unassign** actions to the policies subcollection.

- Assigning and Unassigning policy profiles to resources via the policy_profiles subcollection of a resource

```
/api/:collection/:id/policy_profiles
```

and posting **assign** or **unassign** actions to the policy_profiles subcollection.

- Querying Policies
- Querying Policies of a Resource
- Creating Policies
- Editing Policies
- Deleting Policies
- Policy and Policy Profiles Actions
- Resource Policy Management

- Policy specification
- Assigning policies to a resource
- Unassigning policies from a resource
- Resolving Policies of a Resource
- Resolving Policy Profiles of a Resource

Querying Policies

Querying all policies in the system is simply:

```
GET /api/policies
```

```
{
  "name": "policies",
  "count": 20,
  "subcount": 20,
  "resources": [
    {
      "href": "http://localhost:3000/api/policies/7"
    },
    {
      "href": "http://localhost:3000/api/policies/8"
    },
    ...
  ]
}
```

And expanding the policies, providing additional details:

```
GET /api/policies?expand=resources
```

```
{
  "name": "policies",
  "count": 3,
  "subcount": 3,
  "resources": [
    {
      "href": "http://localhost:3000/api/policies/7",
      "id": "7",
      "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "description": "test_control_policy",
      "created_on": "2015-02-04T23:22:32Z",
      "updated_on": "2015-04-09T13:20:58Z",
      "towhat": "Vm",
      "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "created_by": "admin",
      "updated_by": "admin",
      "active": true,
      "mode": "control"
    },
    {
      "href": "http://localhost:3000/api/policies/8",
      "id": "8",
      "name": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
      "description": "server_type_control_policy",
      "created_on": "2015-02-05T13:46:39Z",
      "updated_on": "2015-04-09T13:20:58Z",
      "towhat": "Vm",
      "guid": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
      "created_by": "admin",
      "updated_by": "admin",
      "active": true,
      "mode": "control"
    },
    ...
  ]
}
```

Querying Policies of a Resource

To find out which policies are assigned to a resource, the policies subcollection can be expanded for that particular resource as follows:

```
GET /api/vms/320?expand=policies
```

This will provide the resources as well as the related policies

```
{
  "href": "http://localhost:3000/api/vms/320",
  "id": "320",
  "vendor": "vmware",
  "name": "aab-vm1",
  "description": "this is a test",
  "raw_power_state": "poweredOn",
  ...
  "policies": [
    {
```

```
      "href": "http://localhost:3000/api/vms/320/policies/7",
      "id": "7",
      "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "description": "test_control_policy",
      "created_on": "2015-02-04T23:22:32Z",
      "updated_on": "2015-04-09T13:20:58Z",
      "towhat": "Vm",
      "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "created_by": "admin",
      "updated_by": "admin",
      "active": true,
      "mode": "control"
    }
  ]
}
```

One can also simply query the policies of a resource by the subcollection as follows:

```
GET /api/vms/320/policies
```

```
{
  "name": "policies",
  "count": 3,
  "subcount": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/320/policies/7"
    }
  ],
  "actions": [
    ...
  ]
}
```

Finding the policies that are part of a policy profile is queried the same as regular resources

```
GET /api/policy_profiles/:id?expand=policies
```

or just the policies themselves as follows:

```
GET /api/policy_profiles/:id/policies
```

Creating Policies

Policies can be created via a POST to the policies collection or via the **create** action signature which also allows creation of multiple policies in a single request.

```
POST /api/policies
```

```
{
  "name" : "sample_policy",
  "description" : "Sample Policy",
  "mode" : "compliance",
  "towhat" : "ManageIQ::Providers::Redhat::InfraManager",
  "condition_ids" : [11, 12],
  "policy_contents" : [
    {
      "event_id" : 201,
      "actions" : [
        {
          "action_id" : 3001,
          "opts" : { "qualifier" : "failure" }
        }
      ]
    }
  ]
}
```

or creating multiple policies:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "sample_policy_1", "description" : "Sample Policy 1", ... },
    { "name" : "sample_policy_2", "description" : "Sample Policy 2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating policies.

Editing Policies

```
POST /api/policies/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Policy Description"
```

```
  }
}
```

or editing multiple policies:

```
POST /api/policies

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/policies/101",
      "description" : "Updated Policy Description 1"
    },
    {
      "href" : "http://localhost:3000/api/policies/102",
      "description" : "Updated Policy Description 2"
    },
    ...
  ]
}
```

Deleting Policies

Policies can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/policies/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/policies/101
```

Deleting multiple policies can be done as follows:

```
POST /api/policies

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/policies/101" },
    { "href" : "http://localhost:3000/api/policies/102" },
    ...
  ]
}
```

Policy and Policy Profiles Actions

For managing policies and policy profiles on resource three available actions are available. These are:

Action	Description
assign	Assign a policy or policy profile to the resource
unassign	Unassign a policy or policy profile from the resource
resolve	Resolves a resource policy or policy profile

Resource Policy Management

Policy management on resources can be done by POSTing **assign** and **unassign** actions to the policies or policy_profiles subcollection of resources. Policy management is available on the following primary collections:

Collection
/api/clusters
/api/hosts
/api/policy_profiles
/api/providers
/api/resource_pools
/api/templates
/api/vms

Policy specification

Policies and policy profiles can be specified using one of the following forms:

By href:

```
{
  "href" : "http://localhost:3000/api/policies/:id"
}

{
  "href" : "http://localhost:3000/api/policy_profiles/:id"
}
```

Or by policy Guid:

```
{
  "guid" : "b1fd23bc-acc4-11e4-abc7-b8e85646e742"
}
```

Assigning policies to a resource

Assigning policies or policy_profiles to resources is done by posting an **assign** action against the policies or policy_profiles subcollection of a resource.

```
POST /api/vms/320/policies
```

```
{
  "action" : "assign",
  "resource" : { "href" : "http://localhost:3000/api/policies/7" }
}
```

or multiple policies:

```
{
  "action" : "assign",
  "resource" : [
    { "href" : "http://localhost:3000/api/policies/7" },
    { "href" : "http://localhost:3000/api/policies/10" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Policy: id:'7' description:'test_control_policy' guid:'b1fd23bc-acc4-11e4-abc7-b8e85646e742'",
      "href": "http://localhost:3000/api/vms/320",
      "policy_id": "7",
      "policy_href": "http://localhost:3000/api/policies/7"
    }
  ]
}
```

Unassigning policies from a resource

Unassign policies or policy_profiles to resources is done by posting an **unassign** action against the policies or policy_profiles subcollection of a resource.

```
POST /api/vms/320/policies
```

```
{
  "action" : "unassign",
  "resource" : { "href" : "http://localhost:3000/api/policies/7" }
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Policy: id:'7' description:'test_control_policy' guid:'b1fd23bc-acc4-11e4-abc7-
b8e85646e742'",
      "href": "http://localhost:3000/api/vms/320",
      "policy_id": "7",
      "policy_href": "http://localhost:3000/api/policies/7"
    }
  ]
}
```

Resolving Policies of a Resource

Resolving a resource policy by targeting the subcollection resource directly as follows:

```
POST /api/vms/320/policies/7
```

```
{
  "action" : "resolve"
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Resolving Policy: id:'7' description:'test_control_policy' guid:'b1fd23bc-acc4-11e4-abc7-b8e85646e742'",
      "result": [
        {
          "id": "7",
          "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "description": "test_control_policy",
```

```
          "created_on": "2015-02-04T23:22:32Z",
          "updated_on": "2015-04-09T13:20:58Z",
          "towhat": "Vm",
          "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "created_by": "admin",
          "updated_by": "admin",
          "active": true,
          "mode": "control",
          "result": "allow",
          "conditions": [

          ],
          "actions": [

          ]
        }
      ],
      "href": "http://localhost:3000/api/vms/320",
      "policy_id": "7",
      "policy_href": "http://localhost:3000/api/policies/7"
    }
  ]
}
```

Resolving Policy Profiles of a Resource

Resolving a resource policy profile by targeting the subcollection resource directly as follows:

```
POST /api/vms/320/policy_profiles/34

{
  "action" : "resolve"
}
```

Response:

```
{
  "success": true,
  "message": "Resolving Policy Profile: id:'34' description:'Control Policies' guid:'f39b25e2-ad3e-11e4-8d4e-
b8e85646e742'",
  "result": [
    {
      "id": "34",
      "name": "f39b25e2-ad3e-11e4-8d4e-b8e85646e742",
      "description": "Control Policies",
      "set_type": "MiqPolicySet",
      "created_on": "2015-02-05T13:57:41Z",
      "updated_on": "2015-02-26T13:42:43Z",
      "guid": "f39b25e2-ad3e-11e4-8d4e-b8e85646e742",
      "mode": "control",
      "result": "allow",
      "policies": [
        {
          "id": "7",
          "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "description": "test_control_policy",
          "created_on": "2015-02-04T23:22:32Z",
          "updated_on": "2015-04-09T13:20:58Z",
          "towhat": "Vm",
          "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "created_by": "admin",
          "updated_by": "admin",
          "active": true,
          "mode": "control",
          "result": "allow",
          "conditions": [

          ],
          "actions": [

          ]
        },
        {
          "id": "8",
          "name": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
          "description": "second_test_control_policy",
          "created_on": "2015-02-05T13:46:39Z",
          "updated_on": "2015-04-09T13:20:58Z",
          "towhat": "Vm",
          "guid": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
          "created_by": "admin",
          "updated_by": "admin",
          "active": true,
          "mode": "control",
          "result": "allow",
          "conditions": [

          ],
          "actions": [
```

```
          ]
        }
      ]
    }
  ],
  "href": "http://localhost:3000/api/vms/320",
  "policy_profile_id": "34",
  "policy_profile_href": "http://localhost:3000/api/policy_profiles/34"
}
```


- Policy Collections
- Policy Actions
  - Response:
- Policy Conditions
  - Response:
- Policy Events
  - Response:

Policy Collections

Primary Policy and Policy Profile collections:

`/api/policies`

`/api/policy_profiles`

With Policy Profiles being sets of Policies with the ability to query policies of a specific policy profile via:

`GET /api/policy_profiles/:id?expand=policies`

Or just the subcollection:

`GET /api/policy_profiles/:id/policies`

- Policy Actions
- Policy Conditions
- Policy Events

Policy Actions

Policy actions can be queried as a primary collection or as a subcollection of policies as follows:

`GET /api/policy_actions`

`GET /api/policies/:id?expand=policy_actions`

or simply the subcollection:

`GET /api/policies/:id/policy_actions`

In this example, we're expanding all policy actions

`GET /api/policy_actions?expand=resources`

Response:


```
{
  "name": "policy_actions",
  "count": 33,
  "subcount": 33,
  "resources": [
    {
      "href": "http://localhost:3000/api/policy_actions/1",
      "id": "1",
      "name": "audit",
      "description": "Generate Audit Event",
      "created_on": "2014-11-20T19:10:05Z",
      "updated_on": "2014-11-20T19:10:05Z",
      "guid": "d619567a-70e8-11e4-a5b4-b8e85646e742",
      "action_type": "default",
      "options": {
      }
    },
    {
      "href": "http://localhost:3000/api/policy_actions/2",
      "id": "2",
      "name": "log",
      "description": "Generate log message",
      "created_on": "2014-11-20T19:10:05Z",
      "updated_on": "2014-11-20T19:10:05Z",
      "guid": "d61c3be2-70e8-11e4-a5b4-b8e85646e742",
      "action_type": "default",
      "options": {
      }
```

```
    },
    {
      "href": "http://localhost:3000/api/policy_actions/3",
      "id": "3",
      "name": "evm_event",
      "description": "Show EVM Event on Timeline",
      "created_on": "2014-11-20T19:10:05Z",
      "updated_on": "2014-11-20T19:10:05Z",
      "guid": "d61cb05e-70e8-11e4-a5b4-b8e85646e742",
      "action_type": "default",
      "options": {
      }
    },
    {
      "href": "http://localhost:3000/api/policy_actions/4",
      "id": "4",
      "name": "script",
      "description": "Execute an external script",
      "created_on": "2014-11-20T19:10:05Z",
      "updated_on": "2014-11-20T19:10:05Z",
      "guid": "d61d225a-70e8-11e4-a5b4-b8e85646e742",
      "action_type": "default",
      "options": {
      }
    },
    ...
  ]
}
```


Policy Conditions

Policy conditions can be queried as a primary collection or as a subcollection of policies as follows:

```
GET /api/conditions
```

```
GET /api/policies/:id?expand=conditions
```

or simply the subcollection:

```
GET /api/policies/:id/conditions
```

In this example, we're expanding the conditions of a policy:

```
GET /api/policies/10?expand=conditions
```

Response:


```
{
  "href": "http://localhost:3000/api/policies/10",
  "id": "10",
  "name": "85dd1f68-ae53-11e4-9c45-b8e85646e742",
  "description": "web_server_policy",
  "created_on": "2015-02-06T22:57:27Z",
  "updated_on": "2015-04-09T13:20:58Z",
  "towhat": "Vm",
  "guid": "85dd1f68-ae53-11e4-9c45-b8e85646e742",
  "created_by": "admin",
  "updated_by": "admin",
  "active": true,
  "mode": "control",
  "conditions": [
    {
      "href": "http://localhost:3000/api/policies/10/conditions/1",
      "id": "1",
      "name": "fe75d0bc-860f-11e4-b12d-b8e85646e742",
      "description": "Memory For Web Server",
      "modifier": "allow",
      "expression": {
        "exp": {
          ">=": {
            "field": "Vm-num_cpu",
            "value": "2"
          }
        },
        "context_type": null
      },
      "towhat": "Vm",
      "created_on": "2014-12-17T17:13:17Z",
      "updated_on": "2015-02-08T15:16:16Z",
      "guid": "fe75d0bc-860f-11e4-b12d-b8e85646e742"
    }
  ]
}
```


Policy Events

Policy events can be queried as a primary collection or as a subcollection of policies as follows:

`GET /api/events`

`GET /api/policies/:id?expand=events`

or simply the subcollection:

`GET /api/policies/:id/events`

In this example, we're expanding all policy events

`GET /api/events?expand=resources`

Response:


```
{
  "name": "events",
  "count": 121,
  "subcount": 121,
  "resources": [
    {
      "href": "http://localhost:3000/api/events/1",
      "id": "1",
      "name": "evm_server_start",
      "description": "EVM Server Start",
      "created_on": "2014-11-20T19:10:11Z",
      "updated_on": "2014-11-20T19:10:11Z",
      "guid": "d9edd528-70e8-11e4-a5b4-b8e85646e742",
      "event_type": "Default"
    },
    {
      "href": "http://localhost:3000/api/events/2",
      "id": "2",
      "name": "evm_server_stop",
      "description": "EVM Server Stop",
      "created_on": "2014-11-20T19:10:11Z",
      "updated_on": "2014-11-20T19:10:11Z",
      "guid": "d9f44c00-70e8-11e4-a5b4-b8e85646e742",
      "event_type": "Default"
    },
    {
      "href": "http://localhost:3000/api/events/3",
      "id": "3",
      "name": "evm_server_not_responding",
      "description": "EVM Server Not Responding",
      "created_on": "2014-11-20T19:10:11Z",
      "updated_on": "2014-11-20T19:10:11Z",
      "guid": "d9f5dcaa-70e8-11e4-a5b4-b8e85646e742",
      "event_type": "Default"
    },
    ...
  ]
}
```


- Provider Support
  - Querying Providers
    - Cloud Networks
    - Cloud Subnets
    - Load Balancers
    - Virtual Machines
  - Creating Providers
  - Specifying Credentials
  - Specifying Connection Configurations
  - Editing Providers
  - Deleting Providers
  - Refresh Providers
  - Import Vm to Provider
  - Foreman Support
  - Provider Examples

Provider Support

Provider management including CRUD operations as well as the **refresh** action is available via the REST API. Management of providers is through:

`/api/providers`

- Querying Providers
- Creating Providers
- Specifying Credentials
- Specifying Connection Configurations
- Editing Providers

- Deleting Providers
- Refresh Providers
- Import Vm to Provider
- Foreman Support
- Provider Examples

Querying Providers

Providers can be queried as follows:

```
GET /api/providers
```

A specific provider can also be queried by its id:

```
GET /api/providers/:id
```

Foreman type providers can be queried by specifying the provider_class as follows:

```
GET /api/providers?provider_class=provider
```

Cloud Networks

The Cloud Networks of a provider can be queried by asking for the *cloud_networks* subcollection as follows:

```
GET /api/providers/:id/cloud_networks
```

```
GET /api/providers/:id/cloud_networks/:cloud_network_id
```

Or expanding the Cloud Networks in addition to the provider as follows:

```
GET /api/providers/:id?expand=cloud_networks
```

Cloud Subnets

The Cloud Subnets of a provider can be queried by asking for the *cloud_subnets* subcollection as follows:

```
GET /api/providers/:id/cloud_subnets
```

```
GET /api/providers/:id/cloud_subnets/:cloud_subnet_id
```

Or expanding the Cloud Subnets in addition to the provider as follows:

```
GET /api/providers/:id?expand=cloud_subnets
```

Load Balancers

Load Balancers of providers can be queried via the load_balancers subcollection as follows:

Query all Load Balancers of a provider:

```
GET /api/providers/:id/load_balancers
```

Query a specific Load Balancer of a provider:

```
GET /api/providers/:id/load_balancers/:load_balancer_id
```

Or expanding the Load Balancers for the specific provider as follows:

```
GET /api/providers/:id?expand=load_balancers
```

Virtual Machines

Virtual Machines of providers can be queried via the vms subcollection as follows:

Query all VMs of a provider:

```
GET /api/providers/:id/vms
```

Query a specific VM of a provider:

```
GET /api/providers/:id/vms/:vm_id
```

Or expanding the VMs for the specific provider as follows:

```
GET /api/providers/:id?expand=vms
```

Query specific VMs of a provider and get their name and vendor attributes:

```
GET /api/providers/:id/vms?filter[]=name='test*'&attributes=name,vendor&expand=resources
```

Creating Providers

Creating a provider is done via the **create** action as follows:

```
POST /api/providers
```

```
{
  "action" : "create",
  "resource" : {
    # New Provider JSON
  }
}
```

And like other **create** actions, the new provider data can simply be posted to the /api/providers URL.

```
POST /api/providers

{
  # New Provider JSON
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating providers.

For example, creating a VMware type provider:

```
POST /api/providers

{
  "name" : "vCenter 5",
  "type" : "ManageIQ::Providers::Vmware::InfraManager",
  "hostname" : "my_vcenter_50",
  "ipaddress" : "192.168.150.1",
  "zone" :  { "href" : "http://localhost:3000/api/zone/1" },
  "credentials" :  {
    "userid" : "vcenter_admin",
    "password" : "vcenter_password"
  }
}
```

If zone is not specified, the Default zone will be used.

The **type** attribute specifies the supported provider class names which include:

ManageIQ::Providers::Amazon::CloudManager
ManageIQ::Providers::Azure::CloudManager
ManageIQ::Providers::Kubernetes::ContainerManager
ManageIQ::Providers::Microsoft::InfraManager
ManageIQ::Providers::Openshift::ContainerManager
ManageIQ::Providers::Openstack::CloudManager
ManageIQ::Providers::Openstack::InfraManager
ManageIQ::Providers::Lenovo::PhysicalInfraManager
ManageIQ::Providers::Redhat::InfraManager
ManageIQ::Providers::Vmware::CloudManager
ManageIQ::Providers::Vmware::InfraManager

Note:

Please note that the provider **type** classes have changed in the appliance from earlier releases (*v2.0.0* of the API). Please refer to the Provider Types page for a mapping from the old to the new types.

Specifying Credentials

When creating or updating providers, the credentials can be specified as a a single default set, or a compound set where additional credentials are necessary, for example AMPQ for OpenStack or Metrics for RHEVM.

Single default credentials set:

```
{
  "name" : "vCenter 50",
  "type" : "ManageIQ::Providers::Vmware::InfraManager",
  ...
  "credentials" :  {
    "userid" : "vc_admin",
    "password" : "vc_password"
  }
}
```

Compound credentials set:

```
{
  "name" : "RHEVM",
  "type" : " ManageIQ::Providers::Redhat::InfraManager",
  ...
  "credentials" : [
     {
       "userid"    : "default_userid",
       "password"  : "default_password"
     },
     {
```

```
      "userid"    : "metrics_userid",
      "password"  : "metrics_password",
      "auth_type" : "metrics"
    }
  ]
}
```

Specifying Connection Configurations

When creating or updating providers, connection configurations can be set. Connection configurations can be used to specify resources such as an **amqp** event provider for OpenStack, adding Prometheus metrics, or an authentication token.

Prometheus metrics:

```
{
  "name": "Openshift Provider",
  "type": "ManageIQ::Providers::Openshift::ContainerManager",
  ...
  "connection_configurations": [
    {
      "endpoint": {
        "role"                  : "prometheus",
        "hostname"              : "prometheus_host",
        "port"                  : "443",
        "security_protocol"     : "ssl-without-validation",
        "certificate_authority" : "-----BEGIN CERTIFICATE-----",
        "verify_ssl": 0
      },
      "authentication": {
        "role"     :  "prometheus",
        "auth_key" :  "prometheus_auth_key"
      }
    }
  ]
}
```

Amqp event provider:

```
{
  "name": "Openstack Provider",
  "type": "ManageIQ::Providers::Openstack::CloudManager",
  ...
  "connection_configurations": [
    {
      "endpoint": {
        "role"              : "amqp",
        "hostname"          : "amqphost.com",
        "port"              : "5672",
        "security_protocol" : "non-ssl"
      },
      "authentication": {
        "userid"   : "amqp_userid",
        "password" : "amqp_password",
        "role"     : "amqp"
      }
    }
  ]
}
```

Ceilometer event provider:

```
{
  "name"              : "Openstack Provider",
  "type"              : "ManageIQ::Providers::Openstack::CloudManager",
  "hostname"          : "ceilometerhost.com",
  "ipaddress"         : "192.168.150.1",
  "security_protocol" : "ssl",
  ...
  "connection_configurations": {
      "userid"   :  "userid",
      "password" :  "password"
  },
  "connection_configurations": [
      "endpoint" : {
        "role"   : "ceilometer"
      }
    }
  ]
}
```

Editing Providers

Editing provider is available via the **edit** action with the **resource** specifying the attributes to update for that provider.

```
POST /api/providers/:id
```

```
{
  "action" : "edit" ,
  "resource" : {
    "hostname" : "new_vcenter_50",
    "ipaddress" : "192.168.150.2"
```

```
  }
}
```

Note:

type is a restricted attribute and cannot be edited.

Deleting Providers

Deleting a provider is done via the delete method:

```
DELETE /api/providers/:id
```

or via the **delete** action as follows:

```
POST /api/providers/:id

{
  "action" : "delete"
}
```

or by href:

```
POST /api/providers

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/providers/:id" }
  ]
}
```

Refresh Providers

Performing a refresh of providers is done via the **refresh** action.

```
POST /api/providers/:id

{
  "action" : "refresh"
}
```

You can also refresh multiple providers:

```
POST /api/providers

{
  "action" : "refresh",
  "resources" : [
    { "href" : "http://localhost:3000/api/providers/:id" },
    { "href" : "http://localhost:3000/api/providers/:id" }
  ]
}
```

Import Vm to Provider

Importing a VM into a provider is supported via the *import_vm* action.

This is currently supported for importing a Vm from a VMware vCenter provider into a Red Hat Virtualization provider.

```
POST /api/providers/:id

{
  "action" : "import_vm"
  "resource" : {
    "source" : { "href" : "http://localhost:3000/api/vms/11" },
    "target" : {
      "name"       : "new_vm_name",
      "cluster"    : { "href" : "http://localhost:3000/api/clusters/201" },
      "data_store" : { "href" : "http://localhost:3000/api/data_stores/301" },
      "sparse"     : true
    }
  }
}
```

Foreman Support

Management of Foreman providers is provided as a seperate class of providers. This is still accessed via the **/api/providers** entry point, but enabled by a new parameter:

```
  GET|POST|DELETE /api/providers?provider_class=provider
```

The **provider_class** parameter is supported with **/api/provider** for all CRUD operations including the **refresh** action.

An example of creating a Foreman provider:

```
POST /api/providers?provider_class=provider

{
  "type"      : "ManageIQ::Providers::Foreman::Provider",
  "name"      : "new_foreman_111",
```

```
  "url"       : "100.200.300.111",
  "credentials" : {
    "userid"   : "foreman_admin",
    "password" : "foreman_password"
  }
}
```

When creating providers with provider_class as *provider*, the **type** attribute specifies the supported provider class names which include:

ManageIQ::Providers::AnsibleTower::Provider
ManageIQ::Providers::Foreman::Provider
ManageIQ::Providers::Openstack::Provider

Note:

When managing providers of *provider* provider_class, Tagging and Policy management does not currently apply. So requests including the tags, policies and policy_profiles subcollections will be rejected.

Provider Examples

Below you will find example requests for creating various providers.

Ansible Tower Automation Manager:

```
{
  "type": "ManageIQ::Providers::AnsibleTower::AutomationManager",
  "name": "API test",
  "api_version": "3.1.2",
  "provider_id": 10000000000012,
  "credentials": {
    "userid": "userid",
    "password": "password"
  }
}
```

AWS:

```
{
  "type": "ManageIQ::Providers::Amazon::CloudManager",
  "provider_region": "us-east-1",
  "name": "Amazon Example",
  "credentials": {
    "userid": "<access key id>",
    "password": "<secret access key>"
  }
}
```

Azure:

```
{
  "type": "ManageIQ::Providers::Azure::CloudManager",
  "name": "Azure Example",
  "provider_region": "eastus",
  "uid_ems": "<tenant_id>",
  "subscription": "<subscription_id>",
  "credentials": {
    "userid": "<client id>",
    "password": "<client key>"
  }
}
```

Kubernetes Container Manager:

```
{
  "type": "ManageIQ::Providers::Kubernetes::ContainerManager",
  "name": "Kubernetes Test",
  "connection_configurations": [
    {
      "endpoint": {
        "hostname": "00.0.00.00",
        "port": "6443",
        "security_protocol": "ssl"
      },
      "authentication": {
        "authtype": "bearer",
        "auth_key": "<token_here>"
      }
    }
  ]
}
```

Microsoft System Center VMM (SCVMM):

```
{
  "type": "ManageIQ::Providers::Microsoft::InfraManager",
  "name": "SCVMM API example",
  "api_version": "2.1.0",
  "connection_configurations": [
    {
```

```
      "endpoint": {
        "hostname": "00.0.00.00",
        "port": "443",
        "security_protocol": "ssl"
      },
      "authentication": {
        "username": "username",
        "password": "password"
      }
    }
  ]
}
```

OpenShift Container Manager:

```
{
   "name":"ocp-example",
   "port":8443,
   "type":"ManageIQ::Providers::Openshift::ContainerManager",
   "hostname":"ocp.example.com",
   "connection_configurations":[
      {
         "endpoint":{
            "role":"default",
            "security_protocol":"ssl-without-validation",
            "verify_ssl":0
         },
         "authentication":{
            "authtype":"bearer",
            "type":"AuthToken",
            "auth_key":"<auth key>"
         }
      }
   ]
}
```

OpenStack Cloud Manager:

```
{
  "type": "ManageIQ::Providers::Openstack::CloudManager",
  "name": "API Example",
  "api_version": "v2",
  "connection_configurations": [
    {
      "endpoint": {
        "role": "default",
        "hostname": "00.0.00.001",
        "port": "5000",
        "security_protocol": "non-ssl"
      },
      "authentication": {
        "type": "ManageIQ::Providers::Openstack::CloudManager::AuthKeyPair",
        "fingerprint": "fingerprint"
      }
    }
  ]
}
```

OpenStack Infra Manager:

```
{
  "type": "ManageIQ::Providers::Openstack::InfraManager",
  "name": "OSP API example",
  "api_version": "v2",
  "connection_configurations": [
    {
      "endpoint": {
        "hostname": "00.0.00.00",
        "port": "5000",
        "security_protocol": "non-ssl"
      },
      "authentications": {
        "username": "username",
        "password": "password"
      }
    }
  ]
}
```

Red Hat Virtualization Manager:

```
{
  "type": "ManageIQ::Providers::Redhat::InfraManager",
  "name": "Sample Rhevm",
  "port": 5000,
  "hostname": "sample_rhevm.provider.com",
  "ipaddress": "100.200.300.4",
  "security_protocol": "kerberos",
  "certificate_authority": "----BEGIN CERTIFICATE----...",
  "credentials": {
    "userid": "uname",
```

```
    "password": "password"
  }
}
```

VMware Infra Manager:

```
{
  "type": "ManageIQ::Providers::Vmware::InfraManager",
  "name": "Sample VMWare provider",
  "hostname": "sample_vmware.provider.com",
  "ipaddress": "100.200.300.1",
  "credentials": {
    "userid": "uname",
    "password": "pword"
  }
}
```

VMware Cloud Manager:

```
{
  "type": "ManageIQ::Providers::Vmware::CloudManager",
  "api_version": "9.0",
  "name": "Api Test",
  "connection_configurations": [
    {
      "endpoint": {
        "port": "443",
        "hostname": "00.0.00.000"
      },
      "authentication": {
        "username": "username",
        "password": "pword"
      }
    }
  ]
}
```


- Provision Requests
  - Creating Requests
  - Provision Requests Attribute Groups
  - Editing Requests
    - Response:
  - Approving or Denying Requests
    - Approving a Request
    - Denying a Request
  - Monitoring Requests
    - Response:
  - Editing Request Tasks

Provision Requests

Provisioning requests are made available via the following entrypoint.

`/api/provision_requests`

- Creating Requests
- Editing Requests
- Approving or Denying Requests
- Monitoring Requests
- Editing Request Tasks

Creating Requests

Either by specifying a *create* action or by posting the request directly to **/api/provision_requests**.

As an example:

```
{
  "version" : "1.1",
  "template_fields" : {
    "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742"
  },
  "vm_fields" : {
    "number_of_cpus" : 1,
    "vm_name" : "aab_test_vm1",
    "vm_memory" : "1024",
    "vlan" : "nic1"
  },
  "requester" : {
    "user_name" : "jdoe",
    "owner_first_name" : "John",
    "owner_last_name" : "Doe",
```

```
    "owner_email" : "jdoe@sample.com",
    "auto_approve" : true
  },
  "tags" : {
    "network_location" : "Internal",
    "cc" : "001"
  },
  "additional_values" : {
    "request_id" : "1001"
  },
  "ems_custom_attributes" : { },
  "miq_custom_attributes" : { }
}
```

Provision Requests Attribute Groups

Attribute Group	Type	Description
version	string	Interface version. Defaults to 1.1
template_fields	hash	Fields used to find template virtual machine. Provide any or all fields. Supply a guid or ems_guid to protect against matching same-named templates on different Providers within the appliance. Supported fields are: name=[VM Template Name] Example: template_1 guid=[guid value from vms resource] ems_guid=[uid_ems value from vms resource]
vm_fields	hash	Allows for the setting of properties on the Catalog, Hardware, Network, Customize, and Schedule tabs in the Provisioning dialog
requester	hash	Allows for the setting of properties on the Requester tab in the Provisioning dialog
tags	hash	Tags to apply to newly created VM Example: network_location=Internal cc=001
additional_values	hash	Additional values are name-value pairs stored with a provision request, but not used by the core provisioning code. These values are usually referenced from automate methods for custom processing Example: Store a request_id from an external system so the system can be notified during the provisioning process
ems_custom_attributes	hash	Custom attributes applied to the virtual machine through the Provider as part of provisioning
miq_custom_attributes	hash	Custom attributes applied to the virtual machine and stored in the database as part of provisioning

Editing Requests

Updating provision requests can be performed by posting an *edit* action as follows:

```
POST /api/provision_requests/11

{
  "action" : "edit",
  "resource" : {
    "options" : {
      "custom_var1" : "custom value1"
    }
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/provision_requests/11",
  "id": "11",
  "description": "Provision from [template1] to [###]",
  "approval_state": "pending_approval",
  "type": "MiqProvisionRequest",
  "created_on": "2015-04-14T17:36:30Z",
  "updated_on": "2015-04-14T18:32:30Z",
  "requester_id": "88913",
  "requester_name": "API User",
  "request_type": "template",
  "request_state": "pending",
  "message": "VM Provisioning - Request Created",
  "status": "Ok"
  "options": {
    "use_pre_dialog": false,
    "request_type": "template",
    "miq_request_dialog_name": "miq_provision_dialogs",
    "src_vm_id": [
      109996,
      "template1"
    ],
    "src_vm_nics": [],
    "src_vm_lans": [],
    "src_ems_id": [
      59136,
      "ems_0000000000002"
    ],
    "placement_auto": [
      true,
```

```
      1
    ],
    "vm_tags": [],
    "ws_values": {
    },
    "ws_ems_custom_attributes": {
    },
    "ws_miq_custom_attributes": {
    },
    "tags": {
    }
    "userid": "admin",
    "source_id": "109996",
    "source_type": "VmOrTemplate",
    "custom_var1" : "custom value1"
  }
}
```

Approving or Denying Requests

If a provision is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the provision request resource.

Approving a Request

```
POST /api/provision_requests/:id

{
  "action" : "approve",
  "reason" : "reason for approval"
}
```

Denying a Request

```
POST /api/provision_requests/:id

{
  "action" : "deny",
  "reason" : "reason for denial"
}
```

Monitoring Requests

Once a provision request is created, the response result will include the queryable provision request itself, i.e. **/api/provision_requests/:id**

Response:

```
{
  "results": [
    {
      "id": "3068",
      "description": "Provision from [template1] to [###]",
      "approval_state": "pending_approval",
      "type": "MiqProvisionRequest",
      "created_on": "2015-04-14T17:36:30Z",
      "updated_on": "2015-04-14T17:36:30Z",
      "requester_id": "88913",
      "requester_name": "API User",
      "request_type": "template",
      "request_state": "pending",
      "message": "VM Provisioning - Request Created",
      "status": "Ok"
      "options": {
        "use_pre_dialog": false,
        "request_type": "template",
        "miq_request_dialog_name": "miq_provision_dialogs",
        "src_vm_id": [
          109996,
          "template1"
        ],
        "src_vm_nics": [],
        "src_vm_lans": [],
        "src_ems_id": [
          59136,
          "ems_0000000000002"
        ],
        "placement_auto": [
          true,
          1
        ],
        "vm_tags": [],
        "ws_values": {
        },
        "ws_ems_custom_attributes": {
        },
        "ws_miq_custom_attributes": {
        },
        "tags": {
        }
      },
```

```
      "userid": "admin",
      "source_id": "109996",
      "source_type": "VmOrTemplate"
    }
  ]
}
```

In the above example, the request could be queried periodically until the **request_state** reaches the **finished** state. Specifically,

```
GET /api/provision_requests/3068
```

Note:

The requests tasks of a provisioning request can also be queried by expanding the request_tasks subcollection as follows:

```
GET /api/provision_requests/:id?expand=request_tasks
```

or by fetching them directly as follows:

```
GET /api/provision_requests/:id/request_tasks?expand=resources
```

Editing Request Tasks

Request tasks for Provision Requests can be updated as follows:

```
POST /api/provision_requests/:id/request_tasks/:request_task_id
```

```
{
  "action" : "edit",
  "resource" : {
    "options" : {
      "request_param_a" : "value_a",
      "request_param_b" : "value_b"
    }
  }
}
```

For a list of attributes available for the provision request attribute groups please refer to the Provision Request Supported Attributes in the appendices section.

- Request Support
  - Creating Requests
  - Querying Requests
  - Editing Requests
  - Approving or Denying Requests
    - Approving a Request
    - Denying a Request
  - Request Types

Request Support

Support for Requests in the API is provided and implemented by specific request types as listed here:

API Collection	Request Type
/api/automation_requests	Automation Requests
/api/provision_requests	Provision Requests
/api/service_requests	Service Requests

A lower level support for creating and updating requests is also provided by the following:

```
/api/requests
```

- Creating Requests
- Querying Requests
- Editing Requests
- Approving or Denying Requests
- Request Types

Creating Requests

Either by specifying a *create* action or by posting the request directly to **/api/requests**.

As an example:

```
POST /api/requests
```

```
{
  "options" : {
    "request_type" : "service_reconfigure",
    "src_id" : "4",
    "service_param1" : "updated_value1",
    "service_param2" : "updated_value2"
  },
```

```
  "requester" : {
    "user_name" : "admin"
  },
  "auto_approve" : true | false
}
```

Response:

```
{
  "results": [
    {
      "id": "97",
      "description": "Service Reconfigure for: svc4",
      "approval_state": "pending_approval",
      "type": "ServiceReconfigureRequest",
      "created_on": "2016-11-10T21:39:08Z",
      "updated_on": "2016-11-10T21:39:08Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "service_reconfigure",
      "request_state": "pending",
      "message": "Service Reconfigure - Request Created",
      "status": "Ok",
      "options": {
        "request_type": "service_reconfigure",
        "src_id": "4",
        "service_param1": "updated_value1",
        "service_param2": "updated_value2"
      },
      "userid": "admin",
      "source_id": "4",
      "source_type": "Service",
      "tenant_id": "1",
      "process": true
    }
  ]
}
```

Querying Requests

Once a request is created, the response result as shown above will include the id of request itself.

In the above example, the request could be queried periodically until the **request_state** reaches the **finished** state.

```
GET /api/requests/97
```

NOTE: The requests tasks of a request can also be queried by expanding the request_tasks subcollection:

```
GET /api/requests/:id?expand=request_tasks
```

Editing Requests

Updating requests is provided by specifying an *edit* action as follows:

```
POST /api/requests/97

{
  "action" : "edit",
  "resource" : {
    "options" : {
      "service_param1" : "new_updated_value1",
      "service_param2" : "new_updated_value2"
    },
    "requester" : {
      "user_name" : "admin"
    }
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/requests/97",
  "id": "97",
  "options": {
    "request_type": "service_reconfigure",
    "src_id": "4",
    "service_param1": "new_updated_value1",
    "service_param2": "new_updated_value2"
  },
  "description": "Service Reconfigure for: svc3",
  "approval_state": "pending_approval",
  "type": "ServiceReconfigureRequest",
  "created_on": "2016-11-10T21:39:08Z",
  "updated_on": "2016-11-10T22:11:03Z",
  "requester_id": "1",
  "requester_name": "Administrator",
  "request_type": "service_reconfigure",
  "request_state": "pending",
  "message": "Service Reconfigure - Request Created",
  "status": "Ok",
```

```
  "userid": "admin",
  "source_id": "4",
  "source_type": "Service",
  "tenant_id": "1",
  "process": true
}
```

Approving or Denying Requests

If a request is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the request resource.

Approving a Request

```
POST /api/requests/:id

{
  "action" : "approve",
  "reason" : "reason for approval"
}
```

Denying a Request

```
POST /api/requests/:id

{
  "action" : "deny",
  "reason" : "reason for denial"
}
```

Request Types

The details for creating each type of request using the lower level API, including validation of *request_type* and necessary options is driven by the model.

The basic structure for creating the request is as follows:

```
{
  "options" : {
    "request_type" : "...",
    ...
  },
  "requester" : {
    "user_name" : "..."
  },
  "auto_approve" : true | false
}
```

The list of supported *request_type* values as of this writing is shown here below:

request_type	Description
automation	Automation
clone_to_service	Service Provision
clone_to_template	VM Publish
clone_to_vm	VM Clone
host_pxe_install	Host Provision
provision_via_foreman	Foreman Provision
service_reconfigure	Service Reconfigure
template	VM Provision
vm_migrate	VM Migrate
vm_reconfigure	VM Reconfigure

- Reports Support
  - Querying Reports
  - Running Reports
    - Request:
    - Response:
  - Scheduling Reports
    - Example: “One off” report
      - Request
      - Response
  - Requesting Report Result Downloads
    - Request:
    - Response:
  - Importing Reports
    - Request:
    - Response:

Reports Support

Support for querying, importing and running Reports is now available. Access to reports and report results is through the following collections:

`/api/reports`

`/api/results`

- Querying Reports
- Running Reports
- Scheduling Reports
- Requesting Report Result Downloads
- Importing Reports

Querying Reports

Querying Available Reports

```
GET /api/reports
```

Querying a Single Report

```
GET /api/reports/:id
```

Querying Results of a Particular Reports

```
GET /api/reports/:id/results
```

Optionally, expanding all results

```
GET /api/reports/:id/results?expand=resources
```

Querying a single result of a report

```
GET /api/reports/:id/results/:result_id
```

Querying report results

```
GET /api/results
```

Running Reports

Running reports is done via the **run** action on a report resource as follows:

```
POST /api/reports/:id

{
  "action" : "run"
}
```

Example for running a report on the following:

```
{
  "href": "http://localhost:3000/api/reports/80",
  "id": "80",
  "name": "VM Relationships"
}
```

Request:

```
POST /api/reports/80

{
  "action" : "run"
}
```

Response:

```
{
  "success": true,
  "message": "running report 80",
  "href": "http://localhost:3000/api/reports/80",
  "result_id": "27",
  "result_href": "http://localhost:3000/api/results/27",
  "task_id": "726",
  "task_href": "http://localhost:3000/api/tasks/726"
}
```

The report run's task can be queried periodically via the task_href specified above as follows:

```
GET /api/tasks/726

{
  "href": "http://localhost:3000/api/tasks/726",
  "id": "726",
  "name": "Generate Report: 'VM Relationships'",
  "state": "Queued",
  "status": "Ok",
  "message": "Task has been queued",
  "userid": "system",
```

```
  "created_on": "2015-11-09T18:44:41Z",
  "updated_on": "2015-11-09T18:44:41Z"
}

{
  "href": "http://localhost:3000/api/tasks/726",
  "id": "726",
  "name": "Generate Report: 'VM Relationships'",
  "state": "Finished",
  "status": "Ok",
  "message": "Generating report complete",
  "userid": "system",
  "created_on": "2015-11-09T18:44:41Z",
  "updated_on": "2015-11-09T18:47:13Z"
}
```


With the results queried via the result_href provided above:


```
GET /api/results/27

{
  "href": "http://localhost:3000/api/results/27",
  "id": "27",
  "name": "VM Relationships",
  "miq_report_id": "80",
  "miq_task_id": "726",
  "userid": "system",
  "report_source": "Requested by user",
  "db": "Vm",
  "report": {
    "name": "VM Relationships",
    "title": "VM Relationships",
    "rpt_group": "Relationships - Virtual Machines, Folders, Clusters",
    "rpt_type": "Default",
    "priority": 10,
    "db": "Vm",
    "cols": [
      "name",
      "v_owning_datacenter",
      "v_owning_folder",
      "v_owning_resource_pool"
    ],
    "include": {
      "ext_management_system": {
        "columns": [
          "name"
        ]
      },
      "host": {
        "columns": [
          "v_owning_cluster",
          "hostname"
        ]
      },
      "storage": {
        "columns": [
          "name"
        ]
      }
    },
    "col_order": [
      "name",
      "ext_management_system.name",
      "v_owning_datacenter",
      "host.v_owning_cluster",
      "host.hostname",
      "v_owning_folder",
      "v_owning_resource_pool",
      "storage.name"
    ],
    "headers": [
      "VM Name",
      "Provider Name",
      "Datacenter",
      "Cluster",
      "Host",
      "Folder",
      "Resource Pool",
      "Datastore"
    ],
    "conditions": {
      "exp": {
        "IS NOT EMPTY": {
          "field": "Vm.host-name"
        }
      },
      "col_details": {
        "Vm.host-name": {
          "data_type": "string",
          "virtual_reflection": false,
          "virtual_column": false,
```

```
          "excluded_by_preprocess_options": false,
          "tag": false,
          "include": {
            "host": {
            }
          },
          "format_sub_type": "string"
        }
      },
      "pexp": {
        "IS NOT EMPTY": {
          "field": "Vm.host-name"
        }
      },
      "ruby": "<value ref=vm, type=string>/virtual/host/name</value> != \"\""
    },
    "order": "Ascending",
    "sortby": [
      "name"
    ],
    "filename": "450_Relationships - Virtual Machines, Folders, Clusters/010_VMs Relationships.yaml",
    "file_mtime": "2015-08-19T13:23:50Z",
    "template_type": "report",
    "col_formats": [

    ],
    "rpt_options": {
    },
    "extras": {
      "total_html_rows": 1709
    },
    "user_categories": [

    ],
    "trend_data": {
    }
  },
  "created_on": "2015-11-09T18:52:48Z",
  "last_run_on": "2015-11-09T18:52:56Z",
  "last_accessed_on": "2015-11-09T18:52:56Z",
  "result_set": [
    {
      "name": "aab-vm1",
      "v_owning_datacenter": "",
      "v_owning_folder": "",
      "v_owning_resource_pool": "",
      "host.v_owning_cluster": "",
      "host.hostname": "test-host.testdomain.com",
      "storage.name": "StarStorage1"
      "id": "320"
    },
    {
      "name": "aab_demo_vm",
      "v_owning_datacenter": "",
      "v_owning_folder": "",
      "v_owning_resource_pool": "",
      "host.v_owning_cluster": "",
      "host.hostname": "test-host2.testdomain.com",
      "storage.name": "StarStorage2",
      "id": "1"
    },
    ...
  ]
}
```


Note that one can query all the results run for a report as follows:

```
GET /api/reports/80/results?expand=resources&attributes=created_on
```


```
{
  "name": "results",
  "count": 27,
  "subcount": 2,
  "resources": [
    {
      "href": "http://localhost:3000/api/reports/80/results/27",
      "id": "27",
      "created_on": "2015-11-09T18:52:48Z"
    },
    {
      "href": "http://localhost:3000/api/reports/80/results/13",
      "id": "13",
      "created_on": "2015-07-29T18:24:35Z"
    }
  ]
}
```


Scheduling Reports

Scheduling a report is done via the **schedule** action on a report resource as follows

```
POST /api/reports/:id

{
  "action" : "schedule"
  "resource": {
    "send_email": true|false,
    "start_date": DateTime,
    "time_zone":  TZ,
    "interval": {
      "unit":  "minutely"|"hourly"|"daily"|"weekly"|"monthly"|"once",
      "value": "once"
    },
    ...
  }
}
```

Example: “One off” report

Request

```
POST /api/reports/5

{
  "action": "schedule",
  "resource": {
    "name":        "Chargeback Report Schedule - 2000/01/01",
    "description": "Chargeback Report 2000 New Year Report",
    "resource_type": "MiqReport",
    "start_date": "2000-01-01T00:00:00Z",
    "time_zone": "UTC",
    "interval": { "unit": "once" },
    "enabled": true
  }
}
```

Response

```
{
  "success":       true,
  "message":       "scheduling of report 5",
  "schedule_id":   "10",
  "schedule_href": "http://localhost:3000/api/reports/5/schedules/10",
  "href":          "http://localhost:3000/api/reports/5"
}
```

Requesting Report Result Downloads

Requesting a **txt**, **csv** or **pdf** download of the report results can be done via the **request_download** action requiring the desired *result_type* as follows:

Request:

```
POST /api/results/22

{
  "action" : "request_download",
  "result_type" : "csv"
}
```

This triggers an async request of the report and returns an action response as follows:

Response:

```
{
  "success": true,
  "message": "Requesting a download of a csv report for Result id:22 name:VMware VM Summary",
  "task_id": "46",
  "task_href": "http://localhost:3000/api/tasks/46",
  "task_results_href": "http://localhost:3000/api/tasks/46/task_results"
}
```

The task can simply be queried for a *Finished* state and the results then fetched via the *task_results_href* provided above.

Querying for the task completion:

```
GET /api/tasks/46

{
  "href": "http://localhost:3000/api/tasks/46",
  "id": "46",
  "name": "Generate Report result [csv]: 'VMware VM Summary'",
  "state": "Finished",
  "status": "Ok",
  "message": "Generate Report result [csv]",
  "userid": "admin",
  "created_on": "2020-05-06T17:15:29Z",
  "updated_on": "2020-05-06T17:15:30Z",
  "pct_complete": null,
  "context_data": {
```

```
    "result_id": "22",
    "result_type": "csv",
    "session_id": "1cae5e9c-9b3e-4ee1-a286-bc2a0131233d-7f4bc905"
  },
  "results": null,
  "miq_server_id": "3",
  "identifier": null,
  "started_on": "2020-05-06T17:15:30Z",
  "zone": null,
  "actions": [
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/tasks/46"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/tasks/46"
    }
  ]
}
```

And the results can be fetched as follows:

```
GET /api/tasks/46/task_results
```

This will return the report with the appropriately typed **Content-Type**, *application/txt*, *application/csv*, or *application/pdf* as per the requested download.

With a **request_download** action targeted at the collection level, one can also request the download for multiple report results as follows:

```
POST /api/results
```

```
{
  "action" : "request_download",
  "resources" : [
      { "id" : "15", "result_type" : "txt" },
      { "id" : "17", "result_type" : "txt" },
      { "id" : "22", "result_type" : "csv" }
   ]
}
```

```
{
  "results" : [
    {
      "success": true,
      "message": "Requesting a download of a txt report for Result id:15 name:VMware VM Summary",
      "task_id": "54",
      "task_href": "http://localhost:3000/api/tasks/54",
      "task_results_href": "http://localhost:3000/api/tasks/54/task_results"
    },
    {
      "success": true,
      "message": "Requesting a download of a txt report for Result id:17 name:VMware VM Summary",
      "task_id": "55",
      "task_href": "http://localhost:3000/api/tasks/55",
      "task_results_href": "http://localhost:3000/api/tasks/55/task_results"
    },
    {
      "success": true,
      "message": "Requesting a download of a txt report for Result id:22 name:VMware VM Summary",
      "task_id": "56",
      "task_href": "http://localhost:3000/api/tasks/56",
      "task_results_href": "http://localhost:3000/api/tasks/56/task_results"
    }
  ]
}
```

Each can then be individually queried and downloaded upon completion.

Importing Reports

Importing reports to the system can be done by posting an **import** action to the reports collection as follows:

```
POST /api/reports
```

```
{
  "action" : "import",
  "resource" : {
    "report" : { JSON serialized report },
    "options" : { "save" : true }
  }
}
```

Here's an example:

Request:

```
POST /api/reports
```

```
{
  "action" : "import",
  "resource" : {
    "report" : {
      "menu_name" : "Test Report",
      "col_order" : ["col1", "col2", "col3"],
      "cols"      : ["col1", "col2", "col3"],
      "rpt_type"  : "Custom",
      "title"     : "Test Report",
      "db"        : "My::Db",
      "rpt_group" : "Custom"
    },
    "options" : { "save" : true }
  }
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Imported Report: [Test Report]",
      "result": {
        "id": "199",
        "name": "Test Report",
        "title": "Test Report",
        "rpt_group": "Custom",
        "rpt_type": "Custom",
        "db": "My::Db",
        "cols": [
          "col1",
          "col2",
          "co3l"
        ],
        "col_order": [
          "col1",
          "col2",
          "col3"
        ],
        "created_on": "2015-11-09T19:44:37Z",
        "updated_on": "2015-11-09T19:44:37Z",
        "miq_group_id": "1",
        "user_id": "1"
      }
    }
  ]
}
```


- Role Management
  - Querying Roles
  - Creating Roles
  - Editing Roles
    - Assigning features to a role
    - Unassigning features from a role
  - Deleting Roles

Role Management

Management of User Roles is provided via the following collection:

`/api/roles`

as well as the associated product features via:

`/api/features`

Full CRUD actions on roles is available:

- Querying Roles
- Creating Roles
- Editing Roles
- Deleting Roles

Querying Roles

Querying all roles in the system is simply:

`GET /api/roles`

Getting details on a specific role:

`GET /api/roles/:id`

Including the features it entitles:

```
GET /api/roles/:id?expand=features
```

Or simply querying the features entitled by the role:

```
GET /api/roles/:id/features?expand=resources
```

Querying all product features of the system that can be assigned to a role.

```
GET /api/features
```

Creating Roles

Roles can be created via a POST to the roles collection or via the **create** action signature which also allows creation of multiple roles in a single request.

```
POST /api/roles
```

```
{
  "action" : "create",
  "resource" : {
    "name" : "sample_role",
    "settings" : { "restrictions" : { "vms" : "user" } },
    "features" : [
      { "identifier" : "vm_explorer" },
      { "identifier" : "ems_infra_tag" },
      { "identifier" : "miq_report_run" }
    ]
  }
}
```

Note:

restrictions for **vms** can be either *user* or *user_or_group*

features can be specified via **identifier**, **href**, or **id**

or creating multiple roles:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "sample_role1", ... },
    { "name" : "sample_role2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Roles.

Editing Roles

```
POST /api/roles/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "updated_sample_role",
    "settings" : { "restrictions" : { "vms" : "user_or_group" } }
  }
}
```

or editing multiple roles:

```
POST /api/roles
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/roles/101",
      "name" : "updated_sample_role1"
    },
    {
      "href" : "http://localhost:3000/api/roles/102",
      "name" : "updated_sample_role2"
    },
    ...
  ]
}
```

Assigning features to a role

```
POST /api/roles/:id/features
```

```
{
  "action" : "assign",
  "resource" : {
```

```
    "identifier" : "miq_request_view"
  }
}
```

or assigning multiple features:

```
{
  "action" : "assign",
  "resources" : [
    { "identifier" : "miq_request_view" },
    { "identifier" : "storage_manager_show_list" },
    ...
  ]
}
```

Unassigning features from a role

```
POST /api/roles/:id/features
```

```
{
  "action" : "unassign",
  "resource" : {
    { "identifier" : "miq_request_view" }
  }
}
```

or unassigning multiple features:

```
{
  "action" : "unassign",
  "resources" : [
    { "identifier" : "miq_request_view" },
    { "identifier" : "storage_manager_show_list" },
    ...
  ]
}
```

Deleting Roles

Non system roles (i.e. read_only false) can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/roles/101
```

```
{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/roles/101
```

Deleting multiple roles can be done as follows:

```
POST /api/roles
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/roles/101" },
    { "href" : "http://localhost:3000/api/roles/102" },
    ...
  ]
}
```


- Security Groups
  - Querying Security Groups
  - Managing Security Groups
  - Add Security Group
  - Remove Security Group

Security Groups

Management of Security Groups is available via the following collection:

```
/api/security_groups
```

Querying Security Groups

Querying all available Security Groups:

```
GET /api/security_groups
```

Getting details on a specific Security Group:

```
GET /api/security_groups/:id
```

Querying the Security Groups of a specific resource is also possible for the following collections:

Collection
/api/cloud_tenants
/api/instances
/api/providers
/api/vms

```
GET /api/cloud_tenants/:id/security_groups
GET /api/cloud_tenants/:id/security_groups/:security_group_id

GET /api/instances/:id/security_groups
GET /api/instances/:id/security_groups/:security_group_id

GET /api/providers/:id/security_groups
GET /api/providers/:id/security_groups/:security_group_id

GET /api/vms/:id/security_groups
GET /api/vms/:id/security_groups/:security_group_id
```

Managing Security Groups

Management of Security Groups is available for Instances with the following actions:

Action	Description
add	Add security group(s) to the Instance
remove	Remove security group(s) from the Instance

Add Security Group

Security Groups can be added to Instances individually:

```
POST /api/instances/:id/security_groups
```

```
{
  "action" : "add",
  "resource" : {
    "name" : "security_group_a"
  }
}
```

Or in bulk as follows:

```
{
  "action" : "add",
  "resources" : [
    { "name" : "security_group_a" },
    { "name" : "security_group_b" }
  ]
}
```

Remove Security Group

Security groups can be removed from Instances individually:

```
POST /api/instances/:id/security_groups
```

```
{
  "action" : "remove",
  "resource" : {
    "name" : "security_group_c"
  }
}
```

Or in bulk as follows:

```
{
  "action" : "remove",
  "resources" : [
    { "name" : "security_group_d" },
    { "name" : "security_group_e" },
    { "name" : "security_group_f" }
  ]
}
```


- Service Management
  - Service Catalogs
    - Creating Service Catalogs
    - Editing Service Catalogs
    - Assigning Service Templates
    - Ordering Services Service Templates
    - Refresh Service Dialog Fields
      - Refresh Dialog Fields of Service Templates Service Dialogs
      - Refresh Dialog Fields of Service Dialogs
    - Deleting Service Catalogs
  - Service Templates

        - Creating Service Templates
        - Editing Service Templates
        - Deleting Service Templates
    - Services ===
        - Creating Services ====
        - Service Actions ====
        - Service Editing
        - Adding and Removing Resources from Services
        - Adding Provider VMs to Services
        - Requesting a Chargeback Report for a Service
        - Service Retiring (now)
        - Service Retiring (future)
        - Service Reconfiguring
        - Starting Services
        - Stopping Services
        - Suspending Services
        - Deleting Services
    - Service Requests ===
        - Querying Service Requests
        - Approving or Denying Service Requests
            - Approving a Service Request
        - Denying a Service Request
        - Adding Approvers to Service Requests
        - Removing Approvers from Service Requests
    - Editing Request Tasks

Service Management

Service management is provided via the following collections:

Path	Description
/api/service_catalogs	Service Catalogs
/api/service_templates	Service Templates
/api/services	Services
/api/service_requests	Service Requests

Service Catalogs

Full CRUD is available on service catalogs.

- Creating Service Catalogs
- Editing Service Catalogs
- Assigning/Unassigning Service Templates from Service Catalogs
- Ordering Services
- Refresh Service Dialog Fields
- Deleting Service Catalogs

Creating Service Catalogs

`/api/service_catalogs`

Creating service catalog is provided via the following actions:

Method	Versions	Comment
create	2.0.0	Preferred
add	since v1.0	

Example of creating service catalog with *create* method:

`POST /api/service_catalogs`

```
{
   "action" : "create",
   "resource" : {
     "name" : "Sample Service Catalog",
     "description" : "Description of Sample Service Catalog",
     "service_templates" : [
       { "href" : "http://localhost:3000/api/service_templates/3" },
       { "href" : "http://localhost:3000/api/service_templates/4" }
     ]
   }
}
```

Optionally without the create action signature:

```
{
  "name" : "Sample Service Catalog",
  "description" : "Description of Sample Service Catalog",
  "service_templates" : [
    { "href" : "http://localhost:3000/api/service_templates/3" },
    { "href" : "http://localhost:3000/api/service_templates/4" }
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Service Catalogs.

Editing Service Catalogs

Editing a service catalog via the *edit* action as follows:

```
POST /api/service_catalogs/:id

{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Description of Third Service Catalog"
   }
}
```

Assigning Service Templates

Assign and Unassigning service templates to service catalogs is done via the *assign* and *unassign* actions as follows:

```
POST /api/service_catalogs/:id

{
  "action" : "assign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/5" },
    { "href" : "http://localhost:3000/api/service_templates/6" }
  ]
}
```

```
POST /api/service_catalogs/:id

{
  "action" : "unassign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/1" },
    { "href" : "http://localhost:3000/api/service_templates/2" }
  ]
}
```

Ordering Services Service Templates

Ordering services is done by triggering an *order* action on service templates in a service catalog as follows:

```
POST /api/service_catalogs/:id/service_templates

{
  "action" : "order",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "option_0_vm_target_name" : "test-vm-0001",
    "option_0_vm_target_hostname" : "test-vm-0001"
  }
}
```

Note: If you need to include a password in the call to the Ordering Services, you can use the **`password::dialog_password`** field to supply the password as a parameter. This password is passed as a **`string`** datatype.

```
{
  "action" : "order",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "option_0_vm_target_name" : "test-vm-0001",
    "option_0_vm_target_hostname" : "test-vm-0001",
    "password::dialog_password" : "password"
  }
}
```

Refresh Service Dialog Fields

Refreshing dialog fields is done via a *refresh_dialog_fields* action as shown in the following cases:

Refresh Dialog Fields of Service Templates Service Dialogs

Refreshing dialog fields for a service template service dialogs is done as follows:

```
POST /api/service_catalogs/:id/service_templates
```

```
{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "dialog_fields" : {
      "my_field1" : "my field1 value",
      "my_field2" : "my field2 value",
      "my_field3" : "my field3 value",
      ...
    },
    "fields" : [ "my_field1", "my_field3" ]
  }
}
```

Refresh Dialog Fields of Service Dialogs

Refreshing dialog fields for a service dialog is done as follows:

```
POST /api/service_dialogs/:id
```

```
{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "dialog_fields" : {
      "my_field1" : "my field1 value",
      "my_field2" : "my field2 value",
      "my_field3" : "my field3 value",
      ...
    },
    "fields" : [ "my_field1", "my_field3" ]
  }
}
```

Deleting Service Catalogs

Single service catalogs can simply be deleted via the DELETE method as follows:

```
DELETE /api/service_catalogs/:id
```

Multiple service catalogs can also be deleted with a single request as follows:

```
POST /api/service_catalogs
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_catalogs/11" },
    { "href" : "http://localhost:3000/api/service_catalogs/12" },
    { "href" : "http://localhost:3000/api/service_catalogs/13" }
  ]
}
```

Service Templates

Management of Service Templates is available via the following collection:

```
/api/service_templates
```

Full CRUD is available on service templates.

- Creating Service Templates
- Editing Service Templates
- Deleting Service Templates

Creating Service Templates

Service Templates can be created via a POST to the /api/service_templates collection:

```
/api/service_templates
```

```
{
  "name" : "Atomic Service Template",
  "service_type" : "atomic",
  "prov_type"    : "amazon",
  "display"      : "false",
  "config_info"  : {
    "miq_request_dialog_name" : "Dialog_Name",
    "placement_auto"          : [true, 1],
    "number_of_vms"           : [1, "1"],
    "src_vm_id"               : [11, "test_vm"],
    "vm_name"                 : "AtomicVMName",
    "schedule_type"           : ["immediately", "Immediately on Approval"],
    "instance_type"           : [21, "instance_flavor"],
    "src_ems_id"              : [5, "test_provider"],
    "provision"               : {
      "fqname"    : "/Sample/System/ProvisionEndpoint",
      "dialog_id" : "21"
```

```
    },
    "retirement"              : {
      "fqname"    : "/Sample/System/RetirementEndpoint",
      "dialog_id" : "22"
    }
  }
}
```

Editing Service Templates

Editing a single service template as follows:

```
POST /api/service_templates/:id

{
  "action" : "edit",
  "resource" : {
    "name" : "svc_template_01",
    "description" : "This is a sample service template"
  }
}
```

Deleting Service Templates

Deleting single service template via the DELETE method:

```
DELETE /api/service_templates/:id
```

or via the *delete* action as follows:

```
POST /api/service_templates/:id

{
  "action" : "delete"
}
```

Multiple service templates can also be deleted in bulk as follows:

```
POST /api/service_templates

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/11" },
    { "href" : "http://localhost:3000/api/service_templates/12" }
  ]
}
```

Services ===

Accessing services is via the **services** collection as follows:

```
/api/services
```

or addressing an individual Service via the **services** resource:

```
/api/services/:id
```

Creating Services ====

In general, services are created once ordered from service catalogs.

Service resources can also be created as follows:

```
POST /api/services

{
  "action" : "create",
  "resource" : {
    "name" : "test_service_name",
    "description" : "Test Service",
    "parent_service" : { "href" : "http://localhost:3000/api/services/15" }
  }
}
```

In the above example *parent_service* is an *href* or *id* reference to an existing service.

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Services.

Service Actions ====

Once services are ordered from a catalog, they can be:

- Edited (Basic)
- Retired now

- Retired in future
- Reconfigured
- Started
- Stopped
- Suspended
- Deleted

Additional Service actions include:

- Adding and Removing Resources from Services
- Adding Provider VMs to Services
- Requesting a Chargeback Report for a Service

Other than service creates and edits, service actions return the action response as in the following example:

Request:

```
POST /api/services/12

{
  "action" : "start"
}
```

Response:

```
{
  "success": true,
  "message": "Service id:12 name:'SampleService' starting",
  "task_id": "62",
  "task_href": "http://localhost:3000/api/tasks/62",
  "href": "http://localhost:3000/api/services/12"
}
```

Service Editing

```
POST /api/services/:id

{
   "action" : "edit",
   "resource" : {
     "name" : "Updated Service Name",
     "description" : "Updated Service Description"
   }
}
```

Adding and Removing Resources from Services

Adding and removing resources from services is made available with the following actions:

Action
add_resource
remove_resource
remove_all_resources

The following example adds a single resource to a service:

```
POST /api/services/:id

{
  "action" : "add_resource",
  "resource" : { "resource" : { "href" : "http://localhost:3000/api/vms/11" } }
}
```

The following example bulk adds resources to multiple services:

```
POST /api/services

{
   "action" : "add_resource",
   "resources" : [
      {
        "href" : "http://localhost:3000/api/services/3",
        "resource" : { "href" : "http://localhost:3000/api/vms/11" }
      },
      {
        "href" : "http://localhost:3000/api/services/3",
        "resource" : { "href" : "http://localhot:3000/api/vms/12" }
      },
      {
        "href" : "http://localhost:3000/api/services/4",
        "resource" : { "href" : "http://localhost:3000/api/vms/21" }
      },
```

```
      {
        "href" : "http://localhost:3000/api/services/5",
        "resource" : { "href" : "http://localhost:3000/api/vms/22" }
      }
   ]
}
```

The following example removes a single resource from a service:

```
POST /api/services/:id

{
  "action" : "remove_resource",
  "resource" : { "resource" : { "href" : "http://localhost:3000/api/vms/11" } }
}
```

The following example removes resources in bulk from multiple services:

```
POST /api/services

{
  "action" : "remove_resource",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/3",
      "resource" : { "href" : "http://localhost:3000/api/vms/11" }
    },
    {
      "href" : "http://localhost:3000/api/services/3",
      "resource" : { "href" : "http://localhot:3000/api/vms/12" }
    },
    {
      "href" : "http://localhost:3000/api/services/4",
      "resource" : { "href" : "http://localhost:3000/api/vms/21" }
    },
    {
      "href" : "http://localhost:3000/api/services/5",
      "resource" : { "href" : "http://localhost:3000/api/vms/22" }
    }
  ]
}
```

The following example removes all resources from a single service:

```
POST /api/services/:id

{
  "action" : "remove_all_resources"
}
```

The following example removes all resources in bulk from multiple services:

```
POST /api/services

{
  "action" : "remove_all_resources",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/11" },
    { "href" : "http://localhost:3000/api/services/12" },
    { "href" : "http://localhost:3000/api/services/13" }
  ]
}
```

Adding Provider VMs to Services

For cases where VMs are created externally, i.e. via Ansible Playbooks, this action allows the system to identify the VMs created and link those VMs back to the service.

```
POST /api/services/:id

{
  "action" : "add_provider_vms",
  "resource" : {
    "provider" : { "href" : "http://localhot:3000/api/providers/11" }
    "uid_ems" : [
      "provider_vm_id1",
      "provider_vm_id2",
      ...
    ]
  }
}
```

Requesting a Chargeback Report for a Service

This action will request a chargeback report for the service.

```
POST /api/services/:id

{
  "action" : "queue_chargeback_report"
}
```

The queue_chargeback_report action will return an action response which includes the task reference as shown here:

```
{
  "success": true,
  "message": "Queued chargeback report generation for Service id:1 name:'test_service1'",
  "task_id": "39",
  "task_href": "http://localhost:3000/api/tasks/39"
}
```

The task_href can be queried and upon completion of the task, the chargeback report for the service can be queried as follows:

```
GET /api/services/:id?attributes=chargeback_report
```

Service Retiring (now)

```
POST /api/services/:id

{
  "action" : "request_retire"
}
```

Service Retiring (future)

```
POST /api/services/:id

{
  "action" : "request_retire",
  "resource" : { "date" : "10/30/2015", "warn" : "5" }
}
```

Service Reconfiguring

When a service is defined with a Reconfigure Automation Entrypoint, the *reconfigure* action is shown as a valid action for that service.

This action takes any of the dialog fields the service supports during reconfiguration.

```
POST /api/services/:id

{
  "action" : "reconfigure",
  "resource" : {
    "field1" : "updated field1 value",
    "field2" : "updated field2 value",
    ...
  }
}
```

Starting Services

```
POST /api/services/:id

{
  "action" : "start"
}
```

Stopping Services

```
POST /api/services/:id

{
  "action" : "stop"
}
```

Suspending Services

```
POST /api/services/:id

{
  "action" : "suspend"
}
```

Deleting Services

```
DELETE /api/services/:id
```

Or via the delete action for deleting multiple services:

```
POST /api/services

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/101",
    { "href" : "http://localhost:3000/api/services/102",
    { "href" : "http://localhost:3000/api/services/103"
    ...
```

```
  ]
}
```

Service Requests ===

- Querying Service Requests
- Approving or Denying Service Requests
- Adding Approvers to Service Requests
- Removing Approvers to Service Requests
- Editing Request Tasks

Querying Service Requests

Once services are ordered from a catalog, service requests can be monitored as follows:

```
GET /api/service_requests/:id
```

Note:

The request tasks of service requests can be queried by expanding the request_tasks as follows:

```
GET /api/service_requests/:id?expand=request_tasks
```

or by fetching them directly as follows:

```
GET /api/service_requests/:id/request_tasks?expand=resources
```

Approving or Denying Service Requests

If a service request is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the service request resource.

Approving a Service Request

```
POST /api/service_requests/:id

{
  "action" : "approve",
  "reason" : "reason for approval"
}
```

Denying a Service Request

```
POST /api/service_requests/:id

{
  "action" : "deny",
  "reason" : "reason for denial"
}
```

Adding Approvers to Service Requests

Approvers can be added to service requests with the *add_approver* action. Approvers can be specified by *user_id* or *user* reference including an *href* or *id* of the approver.

```
POST /api/service_requests/:id

{
  "action" : "add_approver",
  "resource" : {
    "user_id" : 11
  }
}
```

Or one of the following signatures:

```
{
  "action" : "add_approver",
  "resource" : {
    "user" : { "id" : 11  }
  }
}

{
  "action" : "add_approver",
  "resource" : {
    "user" : { "href" : "http://localhost:3000/api/users/11" }
  }
}
```

Approvers can also be added in bulk as follows:

```
POST /api/service_requests
```

```
{
  "action" : "add_approver",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_requests/101",
      "user_id" : "11"
    },
    {
      "href" : "http://localhost:3000/api/service_requests/102",
      "user" : { "id" : 11 }
    },
    {
      "href" : "http://localhost:3000/api/service_requests/103",
      "user" : { "href" : "http://localhost:3000/api/users/11" }
    }
  ]
}
```

Removing Approvers from Service Requests

Approvers can be removed from service requests with the *remove_approver* action. Approvers can be specified by *user_id* or *user* reference including an *href* or *id* of the approver.

```
POST /api/service_requests/:id

{
  "action" : "remove_approver",
  "resource" : {
    "user_id" : "11"
  }
}
```

Approvers can also be removed in bulk as follows:

```
POST /api/service_requests

{
  "action" : "remove_approver",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_requests/101",
      "user_id" : "11"
    },
    {
      "href" : "http://localhost:3000/api/service_requests/102",
      "user" : { "id" : "11" }
    },
    {
      "href" : "http://localhost:3000/api/service_requests/103",
      "user" : { "href" : "http://localhost:3000/api/users/11" }
    }
  ]
}
```

Editing Request Tasks

Request tasks of Service Requests can be updated as follows:

```
POST /api/service_requests/:id/request_tasks/:request_task_id

{
  "action" : "edit",
  "resource" : {
    "options" : {
      "request_param_a" : "value_a",
      "request_param_b" : "value_b"
    }
  }
}
```


- Service Queries
  - Querying Pictures pertaining to services
  - Querying Provision Dialogs on services and service requests
  - Querying Resource Actions
  - Querying Service and Provision Dialogs
  - Querying Service Dialogs
  - Querying Users that ordered services
  - Querying VMs of Services
  - Querying Orchestration Stacks of Services

Service Queries

Queries to services and related service dialogs are provided via the following collections:

Path	Description
/api/services	Services

Path	Description
/api/service_requests	Service Requests
/api/service_templates	Service Templates
/api/service_dialogs	Service Dialogs

- Querying Pictures
- Querying Provision Dialogs
- Querying Resource Actions
- Querying Service and Provision Dialogs
- Querying Service Dialogs
- Querying Users
- Querying VMs
- Querying Orchestration Stacks

Querying Pictures pertaining to services

Access to pictures for service related collections is made available with the *picture* virtual attribute and related *image_href* attribute.

Querying all resources:

```
GET /api/services?expand=resources&attributes=picture,picture.image_href
```

```
GET /api/service_requests?expand=resources&attributes=picture,picture.image_href
```

```
GET /api/service_templates?expand=resources&attributes=picture,picture.image_href
```

Querying specific resources:

```
GET /api/services/:id?attributes=picture,picture.image_href
```

```
GET /api/service_requests/:id?attributes=picture,picture.image_href
```

```
GET /api/service_templates/:id?attributes=picture,picture.image_href
```

Querying Provision Dialogs on services and service requests

Querying provision dialogs on ordered services and related service_requests is provided by the *provision_dialog* virtual attribute:

```
GET /api/services/:id?attributes=provision_dialog
```

```
GET /api/service_requests/:id?attributes=provision_dialog
```

Querying Resource Actions

Querying resource_actions is provided as a subcollection of service_templates as follows:

```
GET /api/service_templates/:id/resource_actions
```

This allows queries like:

```
GET /api/service_templates/:id/resource_actions
GET /api/service_templates/:id/resource_actions/:resource_action_id
GET /api/service_templates/:id/resource_actions?expand=resources
GET /api/service_templates/:id/resource_actions?expand=resources&filter[]=action='Provision'
```

Querying Service and Provision Dialogs

Access to service and provision dialogs is provided by the primary collections:

/api/service_dialogs
/api/provision_dialogs

Querying Service Dialogs

Service dialogs can be queried directly from the primary collection:

```
GET /api/service_dialogs
GET /api/service_dialogs/:id
GET /api/service_dialogs/:id?resource_action_id=:ra_id
                            &target_id=:target_id
                            &target_type=:target_type
```

Note:

Above :target_type would be a valid resource type containing the service dialog like *service_template* or *service*.

Or from the Service Template where they are defined and from the related ordered services as follows:

```
GET /api/service_templates/:id/service_dialogs
GET /api/service_templates/:id/service_dialogs/:service_dialog_id

GET /api/services/:id/service_dialogs
GET /api/services/:id/service_dialogs/:service_dialog_id
```

Querying Users that ordered services

This is made available via the *user* virtual attribute which reflect the user object.

```
GET /api/services/:id?attributes=user

GET /api/service_requests/:id?attributes=user
```

Optionally, specific user attribute can also be queried as follows:

```
GET /api/services/:id?attributes=user.name,user.email
GET /api/services/:id?attributes=user.name,user.email
```

Querying VMs of Services

This is made available via the *vms* subcollection of services as follows:

```
GET /api/services/:id/vms

GET /api/services/:id?expand=vms
```

It is also possible to fetch additional attributes for the vms of a service all within the same request by using the *vms.* prefix in the attributes specification as follows:

```
GET /api/services/:id?expand=vms&attributes=vms.num_cpu
```

Accessing decorators for VM resources is also made available here as follows:

```
GET /api/services/:id?expand=vms
                     &attributes=vms.num_cpu
                     &decorators=vms.supports_console?,vms.supports_cockpit?,vms.supports_vnc_console?
```

Querying Orchestration Stacks of Services

This is made available via the *orchestration_stacks* subcollection of services as follows:

```
GET /api/services/:id/orchestration_stacks
```

Fetching the orchestration stacks in addition to the service as follows:

```
GET /api/services/:id?expand=orchestration_stacks
```

Accessing a specific orchestration stack of a service as follows:

```
GET /api/services/:id/orchestration_stacks/:orchestration_stack_id
```


- Service Custom Actions
  - Request:
  - Response:
    - Request:
    - Response:

Service Custom Actions

With button groups, custom buttons and related dialogs defined on Service Templates, the REST API provides visibility to custom buttons on service templates and related ordered services.

This is made available with the *custom_actions* and *custom_action_buttons* virtual attributes on service_templates and services.

- *custom_actions* returns the custom buttons as layed out in the UI on the service. The information returned includes any Resource Actions and related Dialogs for the custom button.
- *custom_action_buttons* returns the array of custom buttons without the custom button groups hierarchy or details on related dialogs.

Fetching custom actions and custom action buttons for a service_template or a deployed service

```
GET /api/service_templates/:id?attributes=custom_actions,custom_action_buttons

GET /api/services/:id?attributes=custom_actions,custom_action_buttons
```

Example Service Template with custom buttons:

Request:

```
GET /api/service_templates/7?attributes=custom_actions,custom_action_buttons
```

Response:

```
{
  "href": "http://localhost:3000/api/service_templates/7",
  "id": "7",
  "name": "SampleServiceCatalogItem2",
```

```
  "description": "Sample Service Catalog Item 2",
  "guid": "47edc73a-4cef-11e5-8525-000a27020067",
  "options": {
    "button_order": [
      "cb-1",
      "cb-2",
      "cbg-43",
      "cb-3"
    ]
  },
  "created_at": "2015-08-27T19:10:29Z",
  "updated_at": "2015-09-16T18:24:13Z",
  "display": true,
  "evm_owner_id": "1",
  "service_type": "atomic",
  "prov_type": "generic",
  "service_template_catalog_id": "7",
  "long_description": "",
  "tenant_id": "1",
  "custom_actions": {
    "buttons": [
      {
        "id": "1",
        "guid": "3d5975fe-57bb-11e5-8d22-000a27020067",
        "description": "Rescan Service Offering",
        "applies_to_class": "ServiceTemplate",
        "options": {
          "button_image": 2,
          "display": true
        },
        "userid": "admin",
        "created_on": "2015-09-10T12:55:40Z",
        "updated_on": "2015-09-16T14:28:17Z",
        "name": "Rescan",
        "visibility": {
          "roles": [
            "_ALL_"
          ]
        },
        "applies_to_id": "7",
        "resource_action": {
          "id": "7",
          "resource_id": "1",
          "resource_type": "CustomButton",
          "created_at": "2015-09-10T12:55:40Z",
          "updated_at": "2015-09-16T14:28:17Z",
          "ae_namespace": "SYSTEM",
          "ae_class": "PROCESS",
          "ae_instance": "Request",
          "ae_attributes": {
            "key1": "val1",
            "key2": "val2",
            "request": "InspectMe"
          }
        }
      },
      {
        "id": "2",
        "guid": "7c900d18-57c1-11e5-8d22-000a27020067",
        "description": "Ping Service Status",
        "applies_to_class": "ServiceTemplate",
        "options": {
          "button_image": 3,
          "display": true
        },
        "userid": "admin",
        "created_on": "2015-09-10T13:40:23Z",
        "updated_on": "2015-09-16T14:16:21Z",
        "name": "Ping",
        "visibility": {
          "roles": [
            "_ALL_"
          ]
        },
        "applies_to_id": "7",
        "resource_action": {
          "id": "8",
          "resource_id": "2",
          "resource_type": "CustomButton",
          "created_at": "2015-09-10T13:40:23Z",
          "updated_at": "2015-09-16T14:16:21Z",
          "ae_namespace": "SYSTEM",
          "ae_class": "PROCESS",
          "ae_instance": "Request",
          "ae_attributes": {
            "Class": "System",
            "rel2": "//ManageIQ/System/Request/InspectMe",
            "request": "InspectMe"
          }
        }
      }
```

```
  ],
  "button_groups": [
    {
      "id": "43",
      "name": "MyButtonGroup|ServiceTemplate|7",
      "description": "My Custom Buttons",
      "created_on": "2015-09-10T13:41:20Z",
      "updated_on": "2015-09-15T14:53:49Z",
      "guid": "9e4a9900-57c1-11e5-8d22-000a27020067",
      "set_data": {
        "button_order": [
          3
        ],
        "button_image": 1,
        "display": true,
        "applies_to_class": "ServiceTemplate",
        "applies_to_id": "7",
        "group_index": 1
      },
      "owner_type": "ServiceTemplate",
      "owner_id": "7",
      "buttons": [
        {
          "id": "3",
          "guid": "92ac7d88-5bb9-11e5-b2f7-000a27020067",
          "description": "Simulate Button Dialog",
          "applies_to_class": "ServiceTemplate",
          "options": {
            "button_image": 12,
            "display": true
          },
          "userid": "admin",
          "created_on": "2015-09-15T14:53:49Z",
          "updated_on": "2015-09-16T14:28:50Z",
          "name": "simulate",
          "visibility": {
            "roles": [
              "_ALL_"
            ]
          },
          "applies_to_id": "7",
          "resource_action": {
            "id": "12",
            "dialog_id": "2",
            "resource_id": "3",
            "resource_type": "CustomButton",
            "created_at": "2015-09-15T14:53:49Z",
            "updated_at": "2015-09-16T14:28:50Z",
            "ae_namespace": "SYSTEM",
            "ae_class": "PROCESS",
            "ae_instance": "Request",
            "ae_attributes": {
              "request": "InspectMe"
            },
            "dialog": {
              "description": "Test Dialog 2",
              "buttons": "submit,cancel",
              "label": "TestDialog2",
              "dialog_tabs": [
                {
                  "description": "VM Sizing",
                  "display": "edit",
                  "label": "VmSizing",
                  "position": 0,
                  "dialog_groups": [
                    {
                      "description": "Memory",
                      "display": "edit",
                      "label": "Memory",
                      "position": 0,
                      "dialog_fields": [
                        {
                          "name": "DynamicMemory",
                          "description": "Amount of Memory for VM is Dynamic",
                          "type": "DialogFieldCheckBox",
                          "display": "edit",
                          "display_method_options": {
                          },
                          "required_method_options": {
                          },
                          "default_value": "f",
                          "values_method_options": {
                          },
                          "options": {
                          },
                          "label": "DynamicMemory",
                          "position": 0,
                          "dynamic": false,
                          "read_only": false,
                          "resource_action": {
                            "resource_type": "DialogField",
```

```
                            "ae_attributes": {
                            }
                          }
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  ],
  "custom_action_buttons": [
    {
      "id": "1",
      "guid": "3d5975fe-57bb-11e5-8d22-000a27020067",
      "description": "Rescan Service Offering",
      "applies_to_class": "ServiceTemplate",
      "options": {
        "button_image": 2,
        "display": true
      },
      "userid": "admin",
      "created_on": "2015-09-10T12:55:40Z",
      "updated_on": "2015-09-16T14:28:17Z",
      "name": "Rescan",
      "visibility": {
        "roles": [
          "_ALL_"
        ]
      },
      "applies_to_id": "7"
    },
    {
      "id": "2",
      "guid": "7c900d18-57c1-11e5-8d22-000a27020067",
      "description": "Ping Service Status",
      "applies_to_class": "ServiceTemplate",
      "options": {
        "button_image": 3,
        "display": true
      },
      "userid": "admin",
      "created_on": "2015-09-10T13:40:23Z",
      "updated_on": "2015-09-16T14:16:21Z",
      "name": "Ping",
      "visibility": {
        "roles": [
          "_ALL_"
        ]
      },
      "applies_to_id": "7"
    },
    {
      "id": "3",
      "guid": "b9afcdfa-57c1-11e5-8d22-000a27020067",
      "description": "Simulate Button Dialog",
      "applies_to_class": "ServiceTemplate",
      "options": {
        "button_image": 3,
        "display": true
      },
      "userid": "admin",
      "created_on": "2015-09-10T13:42:06Z",
      "updated_on": "2015-09-16T14:27:30Z",
      "name": "Simulate",
      "visibility": {
        "roles": [
          "_ALL_"
        ]
      },
      "applies_to_id": "7"
    }
  ],
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/service_templates/7"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/service_templates/7"
    }
  ]
}
```

When service templates define custom buttons, the ordered services are extended with custom actions which can be triggered via a POST action.

Note that the custom actions are exposed as actions on the deployed services, not the service_templates. The action name to trigger the action is the lower-cased name of the custom button.

As an example, here's a service without custom buttons:

```
{
  "href": "http://localhost:3000/api/services/91",
  "id": "91",
  "name": "new_service_91",
  "description": "This is a description for the new service 91",
  ...
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/services/91"
    },
    {
      "name": "request_retire",
      "method": "post",
      "href": "http://localhost:3000/api/services/91"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/services/91"
    }
  ]
}
```

Here is an example of a service with custom buttons:

```
{
  "href": "http://localhost:3000/api/services/92",
  "id": "92",
  "name": "enhanced_service_92",
  "description": "This is a description for the enhanced service 92",
  "guid": "24c193a2-57c8-11e5-9332-000a27020067",
  "service_template_id": "7",
  "options": {
    "button_order": [
      "cb-1",
      "cb-2",
      "cb-3"
    ],
    "dialog": {
      "dialog_DynamicMemory": "t"
    }
  },
  "display": true,
  "created_at": "2015-09-10T14:28:02Z",
  "updated_at": "2015-09-10T14:28:02Z",
  "evm_owner_id": "1",
  "miq_group_id": "1",
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "request_retire",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "rescan",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "ping",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "simulate",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    }
  ]
}
```

Here, the *rescan*, *ping* and *simulate* are custom buttons for the service and are returned after the common actions.

For example, to request the *rescan* custom action shown above, this is simply a POST action as follows:

Request:

```
POST /api/services/92

{
  "action" : "rescan"
}
```

Response:

Responses from custom action are the normal action responses from the REST API:

```
{
  "success": true,
  "message": "Invoked custom action rescan for services id: 92",
  "href": "http://localhost:3000/api/services/92"
}
```


- Service Dialogs
  - Creating Service Dialogs
  - Editing Service Dialogs
  - Copying Service Dialogs
  - Refreshing Dialog Fields of Service Dialogs
  - Deleting Service Dialogs

Service Dialogs

Management of Service Dialogs is provided via the collection:

```
/api/service_dialogs
```

The following actions are provided on service_dialogs resources:

- Creating Service Dialogs
- Editing Service Dialogs
- Copying Service Dialogs
- Refreshing Dialog Fields of Service Dialogs
- Deleting Service Dialogs

Creating Service Dialogs

Service Dialogs can be created via a POST to the *service_dialogs* collection or via the **create** action signature which also allows creation of multiple service_dialogs in a single request.

```
POST /api/service_dialogs

{
  "description" : "Dialog",
  "label" : "dialog_label",
  "dialog_tabs" : [
    {
      "description" : "Dialog Tab",
      "position"    : 0,
      "label"       : "dialog_tab_label",
      "dialog_groups" : [
        {
          "description" : "Dialog Group",
          "label"       : "group_label",
          "dialog_fields" : [
            {
              "name"  : "A Dialog Field",
              "label" : "dialog_field_label"
            }
          ]
        }
      ]
    }
  ]
}
```

or creating multiple Service Dialogs:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "Dialog 1", "label" : "dialog_1_label", ... },
    { "description" : "Dialog 2", "label" : "dialog_2_label", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Service Dialogs.

Editing Service Dialogs

Service Dialogs can be updated via the *edit* action as follows:

```
POST /api/service_dialogs/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "label" : "updated_label",
    "content" : {
      "dialog_tabs" : [
        "id" : "11",
        "label" : "updated_tab_label",
        "dialogs_groups" : [
          {
            "id" : "31",
            "dialog_fields" : [
               { "id" : "41" }
            ]
          }
        ]
      ]
    }
  }
}
```

or editing multiple Service Dialogs in a single request as follows:

```
POST /api/service_dialogs
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_dialogs/21",
      "label" : "updated_label_1",
      "content" : ...
    },
    {
      "href" : "http://localhost:3000/api/service_dialogs/22",
      "label" : "updated_label_2",
      "content" : ...
    },
    ...
  ]
}
```

Copying Service Dialogs

Service Dialogs can be copied to create new Service Dialogs via the *copy* action as follows:

```
POST /api/service_dialogs/:id
```

```
{
  "action" : "copy"
}
```

When a label is not specified, the new Service Dialog will be named **Copy of ...** the name of the Service Dialog being copied

Optionally, the label of the new Service Dialog can also be specified as follows:

```
{
  "action" : "copy"
  "resource" : {
    "label" : "test_service_dialog_label"
  }
}
```

Multiple Service Dialogs can also be copied in a single request as follows:

```
POST /api/service_dialogs
```

```
{
  "action" : "copy",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_dialogs/11",
      "label" : "test_service_dialog_label_1"
    },
    {
      "href" : "http://localhost:3000/api/service_dialogs/12",
      "label" : "test_service_dialog_label_2"
    },
    ...
  ]
}
```

Refreshing Dialog Fields of Service Dialogs

Refreshing dialog fields for a Service Dialog is done as follows:

```
POST /api/service_dialogs/:id

{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "resource_action_id" : "6",
    "target_id" : "12",
    "target_type" : "service_template_ansible_tower",
    "dialog_fields" : {
      "my_field1" : "my field1 value",
      "my_field2" : "my field2 value",
      "my_field3" : "my field3 value",
      ...
    },
    "fields" : [ "my_field1", "my_field3" ]
  }
}
```

Deleting Service Dialogs

Deleting a Service Dialog can be done via the *delete* action as follows:

```
POST /api/service_dialogs/:id

{
  "action" : "delete"
}
```

or via the DELETE HTTP method as follows:

```
DELETE /api/service_dialogs/:id
```

One can also delete multiple Service Dialogs via the *delete* action on the collection with references to the service dialogs to delete as follows:

```
POST /api/service_dialogs

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_dialogs/101" },
    { "href" : "http://localhost:3000/api/service_dialogs/102" },
    { "href" : "http://localhost:3000/api/service_dialogs/103" },
    ...
  ]
}
```


- Service Orders
  - Querying Service Orders
  - Creating Service Orders with multiple service requests
  - Adding Service Requests
  - Removing Service Requests
  - Clearing Service Orders
  - Copying Service Orders
  - Deleting Service Orders
  - Ordering Service Orders

Service Orders

The service orders collection and actions are provided for managing Shopping Carts and their related service requests.

```
/api/service_orders
```

Available actions on service_orders instances include:

Action	Description
create	Creating the Shopping Cart
clear	Clearing the Shopping Cart
order	Ordering all service requests in the Shopping Cart
copy	Copying the Shopping Cart
delete	Deleting the Shopping Cart

Some important points on Shopping Carts:

- There is one and only one shopping cart for the authenticated user
- A shopping cart is an /api/service_orders resource with a state of **cart**
- Shopping carts can be referenced by their resource id or preferably via the alphanumeric **cart** identifier.
- resource id's need to be used for accessing non-cart service_orders states of **wish** list and **ordered**

Items to add to a shopping cart are *service_requests* that are specified similarly to when services are ordered from a service template.

- Querying Service Orders
- Creating Service Orders
- Adding Service Requests
- Removing Service Requests
- Clearing Service Orders
- Copying Service Orders
- Deleting Service Orders
- Ordering Service Orders

Querying Service Orders

All the queries listed here below return the service orders for the authenticated user.

Getting all service_orders:

```
GET /api/service_orders
```

Getting the shopping cart:

```
GET /api/service_orders/cart
```

Getting the shopping cart with expanded service_requests:

```
GET /api/service_orders/cart?expand=service_requests,resources
```

Getting the service requests of the shopping cart:

```
GET /api/service_orders/cart/service_requests
```

Creating Service Orders with multiple service requests

```
POST /api/service_orders
```

```
{
  "service_requests" : [
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "super_vm_a",
      "Basic_Host_Ip" : "192.168.120.101"
    },
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "super_vm_b",
      "Basic_Host_Ip" : "192.168.120.102"
    },
    ...
  ]
}
```

Creating a wish list service_order:

```
POST /api/service_orders
```

```
{
  "state" : "wish",
  "service_requests" : [
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/2",
      "host_name" : "sample_service_order_host_name",
      "host_ip" : "192.168.120.101"
    }
  ]
}
```

Adding Service Requests

Adding service requests to a shopping cart can be done by either of the create methods shown above which will all add the service requests to the existing shopping cart:

```
POST /api/service_orders
```

or specifically to the shopping cart or any wish list service_orders via its *id* instead of **cart** as follows:

```
POST /api/service_orders/cart/service_requests
```

```
POST /api/service_orders/:id/service_requests
```

```
{
  "action" : "add",
  "resources" : [
    {
```

```
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "sample_service_order_host_name_2",
      "Basic_Host_Ip" : "192.168.120.102"
    },
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "sample_service_order_host_name_3",
      "Basic_Host_Bad_Ip" : "192.168.120.103"
    },
    ...
  ]
}
```

Note: * POST /api/service_orders is atomic, requests are validated for all requests before adding them. * POST /api/service_orders/cart/service_requests will add the requests one by one, the action responses have the **success** field for each requests reflecting the success or failure of each request.

Removing Service Requests

Removing requests from the shopping cart is available via the *remove* action:

```
POST /api/service_orders/cart/service_requests

{
  "action" : "remove",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_orders/6/service_requests/56" },
    { "href" : "http://localhost:3000/api/service_orders/6/service_requests/57" }
  ]
}
```

Clearing Service Orders

Clearing the shopping cart can be done via the *clear* action:

```
POST /api/service_orders/cart

{
  "action" : "clear"
}
```

Copying Service Orders

A service order can be copied into a new service order via the *copy* action:

```
POST /api/service_orders/:id

{
   "action" : "copy",
   "resource" : {
      "name" : "new_service_order_name"
   }
}
```

Service orders can also be copied in bulk as follows:

```
POST /api/service_orders/:id

{
   "action" : "copy",
   "resources" : [
      {
        "href" : "http://localhost:3000/api/service_orders/1",
        "name" : "new_service_order_name_1"
      },
      {
        "href" : "http://localhost:3000/api/service_orders/2",
        "name" : "new_service_order_name_2"
      },
      ...
   ]
}
```

Deleting Service Orders

Deleting the shopping cart can be done via the *delete* action:

```
POST /api/service_orders/cart

{
   "action" : "delete"
}
```

or simply via the DELETE method:

```
DELETE /api/service_orders/cart
```

Ordering Service Orders

Ordering the shopping cart is done via the *order* action as follows:

```
POST /api/service_orders/cart
```

```
{
   "action" : "order"
}
```


- Setting Ownership
  - Specifying Ownership
  - Setting Ownership to single resource
    - Request:
    - Response:
  - Setting Ownership to multiple resources
    - Request:
    - Response:

Setting Ownership

Setting ownership of resources is provided with the **set_ownership** action. This action is available for the following resources:

/api/auth_key_pairs
/api/cloud_templates
/api/instances
/api/service_templates
/api/services
/api/templates
/api/vms

- Specifying Ownership
- Setting Ownership to single resource
- Setting Ownership to multiple resources

Specifying Ownership

Specifying ownership in requests is done via the **owner** and **group** specification.

```
{
  "owner" : { "href" : "http://localhost:3000/api/users/:id"  },
  "group" : { "href" : "http://localhost:3000/api/groups/:id" }
}
```

Note:

Specifying an owner can be done via **href**, **id**, **name** or **userid**

Note:

Specifying a group can be done via **href**, **id** or **description**

Setting ownership can be done on a specific resource or multiple resources in a single request:

Setting Ownership to single resource

Example setting ownership to a single resource:

Request:

```
POST /api/vms/320
```

```
{
  "action" : "set_ownership",
  "resource" : {
    "owner" : { "userid" : "jdoe" },
    "group" : { "description" : "TestGroup" }
  }
}
```

Response:

```
{
  "success": true,
  "message": "setting ownership of vms id 320 to owner: jdoe, group: TestGroup",
  "href": "http://localhost:3000/api/vms/320"
}
```

Setting Ownership to multiple resources

Example setting ownership to multiple resources with a single request:

Request:

```
POST /api/services
```

```
{
  "action" : "set_ownership",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/104", "owner" : { "name" : "John Doe" } },
    { "href" : "http://localhost:3000/api/services/105", "owner" : { "name" : "John Doe" } },
    { "href" : "http://localhost:3000/api/services/106", "owner" : { "name" : "John Doe" } }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "setting ownership of services id 104 to owner: John Doe",
      "href": "http://localhost:3000/api/services/104"
    },
    {
      "success": true,
      "message": "setting ownership of services id 105 to owner: John Doe",
      "href": "http://localhost:3000/api/services/105"
    },
    {
      "success": true,
      "message": "setting ownership of services id 106 to owner: John Doe",
      "href": "http://localhost:3000/api/services/106"
    }
}
```


- Settings Management
  - Querying Settings
  - Updating Settings
    - Request:
    - Response:
  - Deleting Settings
    - Example:

Settings Management

Management of advanced settings is available on servers, zones and regions. They are provided with the following collection:

```
/api/settings
```

for the server processing the API, or for a particular server, zone or region as follows:

```
/api/servers/:id/settings
```

```
/api/zones/:id/settings
```

```
/api/regions/:id/settings
```

The following actions are available with advanced settings:

- Querying Settings
- Updating Settings
- Deleting Settings

Querying Settings

Settings can be queried as follows:

```
GET /api/servers
```

```
GET /api/servers/:id/settings
```

```
GET /api/zones/:id/settings
```

```
GET /api/regions/:id/settings
```

Note:

Authorization

- For users with an *EvmGroup-super_administrator* role, the complete advanced settings are returned.
- For users with roles that include the *Configuration Settings* feature, only a whitelisted set of settings are returned.
- Requests from all other users will be rejected with a Forbidden error.

Updating Settings

Updating settings can only be performed by users with an *EvmGroup-super_administrator* role.

Updates to settings is available with the HTTP PATCH method as follows:

```
PATCH /api/servers/:id/settings
```

```
PATCH /api/zones/:id/settings
```

```
PATCH /api/regions/:id/settings
```

The following is an example updating the policy_events's purge_window_size:

Request:

```
PATCH /api/servers/1/settings
```

```
{
  "policy_events" : {
    "purge_window_size" : 500
  }
}
```

Response:

The complete updated settings is returned:

```
{
  "ems": {
    "ems_ansible_tower": {
      "blacklisted_event_names": [

      ],
  ...
  "policy_events" : {
    "history": {
      "keep_policy_events": "6.months",
      "purge_window_size": 1000
    },
    "purge_window_size" : 500
  },
  ...
}
```

Deleting Settings

Deleting settings can only be performed by users with an *EvmGroup-super_administrator* role.

Deleting settings is available with the HTTP DELETE method as follows:

```
DELETE /api/servers/:id/settings
```

```
DELETE /api/zones/:id/settings
```

```
DELETE /api/regions/:id/settings
```

Requests must include an array representing the path of the setting to delete.

Example:

```
DELETE /api/servers/1/settings
```

```
[ "policy_events", "purge_window_size" ]
```

This request deletes the policy_event's purge_window_size setting for /api/servers/1 and its value would revert to its default.

- Snapshot Management
  - Snapshot Actions
  - Querying Snapshots
  - Creating Snapshots
  - Reverting Snapshots
  - Deleting Snapshots

Snapshot Management

Management of snapshots is provided for virtual machines and instances via the snapshots subcollection:

```
/api/vms/:id/snapshots
```

```
/api/instances/:id/snapshots
```

Snapshot Actions

The following snapshot actions are available for the supported resources:

Snapshot Action	Vms	Instances
read	X	X
create	X	X

Snapshot Action	Vms	Instances
revert	X	
delete	X	X

- Querying Snapshots
- Creating Snapshots
- Reverting Snapshots
- Deleting Snapshots

Querying Snapshots

Snapshots of machines are queried via the subcollection of vms and instances as follows:

Querying all snapshots of a VM:

```
GET /api/vms/:id/snapshots
```

Querying all snapshots of an Instance:

```
GET /api/instances/:id/snapshots
```

Filtering, sorting and paging on the subcollection as mentioned on the Querying page.

For example:

```
GET /api/vms/11/snapshots?expand=resources&filter[]=name=test*
```

One can also fetch a single snapshot by id:

```
GET /api/vms/:id/snapshots/:snapshot_id
```

```
GET /api/instances/:id/snapshots/:snapshot_id
```

Creating Snapshots

Creating snapshots on Vms and Instances is done by posting the request to the snapshots subcollection.

```
POST /api/vms/:id/snapshots
```

```
POST /api/instances/:id/snapshots
```

```
{
  "name" : "name_of_new_snapshot",
  "description" : "Description of new Snapshot",
  "memory" : false
}
```

In the above example, only *name* is required. *memory* can be set to true to have the snapshot include physical memory.

Reverting Snapshots

Reverting snapshots is only available for Vms.

```
POST /api/vms/:id/snapshots/:snapshot_id
```

```
{
  "action" : "revert"
}
```

Deleting Snapshots

Individual snapshots of Vms and Instances can be deleted via the DELETE method as follows:

```
DELETE /api/vms/:id/snapshots/:snapshot_id
```

```
DELETE /api/instances/:id/snapshots/:snapshot_id
```

As well as the *delete* POST action as follows for VMs:

```
POST /api/vms/:id/snapshots/:snapshot_id
```

or instances:

```
POST /api/instances/:id/snapshots/:snapshot_id
```

```
{
  "action" : "delete"
}
```

Snapshots can also be deleted in bulk via the *delete* action on the snapshots subcollection as follows:

```
POST /api/vms/11/snapshots
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/11/snapshots/3" },
```

```
    { "href" : "http://localhost:3000/api/vms/11/snapshots/4" },
    { "href" : "http://localhost:3000/api/vms/11/snapshots/5" }
  ]
}
```

As well as for instances:

```
POST /api/instances/201/snapshots
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/instances/201/snapshots/7" },
    { "href" : "http://localhost:3000/api/instances/201/snapshots/8" }
  ]
}
```


- Tags & Tagging
  - Tag Management
  - Querying Tags
  - Querying Tags of a Resource
  - Assigning Tags
    - Tag specification
    - Assigning tags to a resource
      - Response:
    - Unassigning tags from a resource
      - Response:
    - Bulk Assigning Tags
      - Response:
    - Bulk Unassigning Tags
      - Response:

Tags & Tagging

- Tag Management
- Querying Tags
- Querying Tags of a Resource
- Assigning Tags
- Tag Specification
- Assigning Tags to a Resource
- Unassigning Tags from a Resource
- Bulk Assigning Tags
- Bulk Unassigning Tags

Tag Management

Management of Tags is provided in the following forms:

- Querying of tags via the primary collection:

```
/api/tags
```

- Assigning and Unassigning tags to resources via the tags subcollection of a resource

```
/api/:collection/:id/tags
```

and posting **assign** or **unassign** actions to the tags subcollection.

Querying Tags

Querying all tags in the system is simply:

```
GET /api/tags
```

However, that only gives us the href's of the tags:

```
{
  "name": "tags",
  "count": 1385,
  "subcount": 1385,
  "resources": [
    {
      "href": "http://localhost:3000/api/tags/1"
    },
    {
      "href": "http://localhost:3000/api/tags/2"
    },
```

```
    {
      "href": "http://localhost:3000/api/tags/3"
    }
    ...
  ]
}
```

Which is not too useful.

Expanding the resources, does provide us with the tag name as follows:

```
GET /api/tags?expand=resources
```

```
{
  "name": "tags",
  "count": 1385,
  "subcount": 1385,
  "resources": [
    {
      "href": "http://localhost:3000/api/tags/1",
      "id": "1",
      "name": "/managed/roles/change_managers"
    },
    {
      "href": "http://localhost:3000/api/tags/2",
      "id": "2",
      "name": "/managed/roles/operators"
    },
    {
      "href": "http://localhost:3000/api/tags/3",
      "id": "3",
      "name": "/managed/roles/cfme_administrators"
    },
    {
      "href": "http://localhost:3000/api/tags/4",
      "id": "4",
      "name": "/managed/location"
    },
    {
      "href": "http://localhost:3000/api/tags/5",
      "id": "5",
      "name": "/managed/location/ny"
    }
    ...
  ]
}
```

For additional details on tags, the **category** and **classification** attributes can be queried on tags as follows:

```
GET /api/tags/5?attributes=category,classification
```

Additionally, a compound attribute called **categorization** is provided to return key information from both **category** and **classification** of tags as follows:

```
GET /api/tags/5?attributes=categorization
```

```
{
  "href": "http://localhost:3000/api/tags/5",
  "id": "5",
  "name": "/managed/location/ny",
  "categorization": {
    "name": "ny",
    "description": "New York",
    "category": {
      "name": "location",
      "description": "Location"
    },
    "display_name": "Location: New York"
  }
}
```

Querying Tags of a Resource

To find out which tags are assigned to a resource, the tags subcollection can be expanded for that particular resource as follows:

```
GET /api/vms/320?expand=tags
```

This will provide the id, href and name of the tags:

```
{
  "href": "http://localhost:3000/api/vms/320",
  "id": "320",
  "vendor": "vmware",
  "name": "aab-vm1",
  "description": "this is a test",
  "raw_power_state": "poweredOn",
  ...
  "tags": [
    {
      "href": "http://localhost:3000/api/vms/320/tags/81",
      "id": "81",
      "name": "/managed/quota_max_cpu/4"
```

```
    },
    {
      "href": "http://localhost:3000/api/vms/320/tags/61",
      "id": "61",
      "name": "/managed/cc/001"
    },
    ...
  ]
}
```

Optionally, one can also query just the tags subcollection of the resource to get additional information about them like the categorization:

```
GET /api/vms/320/tags?expand=resources&attributes=categorization
```

```
{
  "name": "tags",
  "count": 1385,
  "subcount": 5,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/320/tags/81",
      "id": "81",
      "name": "/managed/quota_max_cpu/4",
      "categorization": {
        "name": "4",
        "description": "4",
        "category": {
          "name": "quota_max_cpu",
          "description": "Quota - Max CPUs"
        },
        "display_name": "Quota - Max CPUs: 4"
      }
    },
    {
      "href": "http://localhost:3000/api/vms/320/tags/61",
      "id": "61",
      "name": "/managed/cc/001",
      "categorization": {
        "name": "001",
        "description": "Cost Center 001",
        "category": {
          "name": "cc",
          "description": "Cost Center"
        },
        "display_name": "Cost Center: Cost Center 001"
      }
    },
    ...
  ]
}
```

Assigning Tags

Tag management on resources can be done by POSTing **assign** and **unassign** actions to the tags subcollection of resources. Tags can be managed as subcollections of the following primary collections:

Collection
/api/categories
/api/clusters
/api/data_stores
/api/generic_objects
/api/groups
/api/hosts
/api/lans
/api/providers
/api/resource_pools
/api/services
/api/service_templates
/api/templates
/api/tenants
/api/users
/api/vms

Tag specification

Tags can be specified using one of the following forms:

By Tag category and name:

```
{
  "category" : "department",
  "name"     : "finance"
}
```

By Tag path:

```
{
  "path" : "/managed/department/finance"
}
```

By Tag href:

```
{
  "href" : "http://localhost:3000/api/tags/10"
}
```

Assigning tags to a resource

```
POST /api/vms/320/tags
```

```
{
  "action" : "assign",
  "resources" : [
    { "category" : "department", "name" : "finance" },
    { "category" : "cc", "name" : "001" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
}
```

Unassigning tags from a resource

```
POST /api/vms/320/tags
```

```
{
  "action" : "unassign",
  "resources" : [
    { "category" : "department", "name" : "finance" },
    { "category" : "cc", "name" : "001" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
}
```

Additional Tag Management examples can be found on the main REST API Examples section.

Bulk Assigning Tags

While the general URL for tag assignment via the subcollection as follows:

```
POST /api/vms/320/tags
```

allows one to assign multiple tags to a single resource in one call, it does not provide the ability to assign tags to multiple resources in one call.

Bulk assignment of tags is available via the *assign_tags* action to the following collections:

Collection
/api/vms
/api/services

The *assign_tags* action allows adding one or more tags to one or more resources.

```
POST /api/vms

{
  "action" : "assign_tags",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/vms/11",
      "tags" : [
        { "category" : "department", "name" : "finance" }
      ]
    },
    {
      "href" : "http://localhost:3000/api/vms/12",
      "tags" : [
        { "category" : "cc", "name" : "001" }
      ]
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/vms/11",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/vms/12",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
  ]
}
```

Bulk Unassigning Tags

Tags can also be unassigned in bulk from multiple resources for the following collections via the *unassign_tags* action:

Collection
/api/vms
/api/services

```
POST /api/services

{
  "action" : "unassign_tags",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/5",
      "tags" : [
        { "href" : "http://localhost:3000/api/services/5/tags/81" }
      ]
    },
    {
      "href" : "http://localhost:3000/api/services/6",
      "tags" : [
        { "category" : "cc", "name" : "001" }
      ]
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
```

```
      "href": "http://localhost:3000/api/services/5",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/services/6",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
  ]
}
```


- Tasks
  - Querying Tasks
  - Deleting Tasks

Tasks

Management of Tasks is available via the following endpoint:

```
/api/tasks
```

Querying Tasks

Querying all Tasks:

```
GET /api/tasks
```

Getting details of a specific Task:

```
GET /api/tasks/:id
```

The following actions are available on tasks:

- Deleting Tasks

Deleting Tasks

Tasks can be deleted via either the **delete** POST action or the DELETE HTTP method.

```
POST /api/tasks/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/tasks/101
```

Deleting multiple Tasks can be done as follows:

```
POST /api/tasks

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/tasks/101" },
    { "href" : "http://localhost:3000/api/tasks/102" },
    ...
  ]
}
```


- Tenant Management
- Tenant Quota Management
  - Creating Tenants
  - Editing Tenants
  - Deleting Tenants
  - Creating Tenant Quotas
  - Editing Tenant Quotas
  - Deleting Tenant Quotas

Tenant Management

Tenant management including CRUD operations is available via the REST API. Management of tenants is through:

```
/api/tenants
```


- Creating Tenants
- Editing Tenants

- Deleting Tenants

Tenant Quota Management

Tenant Quota management is made available via the *quotas* subcollection of tenant resources as follows:

```
/api/tenants/:id/quotas
```

- Creating Tenant Quotas
- Editing Tenant Quotas
- Deleting Tenant Quotas

Creating Tenants

Creating a Tenant is done via the **create** action as follows:

```
POST /api/tenants

{
  "action" : "create",
  "resource" : {
    # New Tenant JSON
  }
}
```

And like other **create** actions, the new tenant data can simply be posted to the /api/tenants URL.

```
POST /api/tenants

{
  # New Tenant JSON
}
```

For example, creating a new tenant:

```
POST /api/tenants

{
  "name" : "Test Tenant",
  "description" : "Test Tenant Description",
  "parent" : { "href" : "http://localhost:3000/api/tenants/1" }
}
```

Parent can be specify by *id* or *href*

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Tenants.

Editing Tenants

Editing a Tenant is available via the **edit** action with the **resource** specifying the attributes to update for that tenant.

```
POST /api/tenants/:id

{
  "action" : "edit" ,
  "resource" : {
    "description" : "Updated Description for Tenant"
  }
}
```

Note:

type is a restricted attribute and cannot be edited.

Deleting Tenants

Deleting a Tenant is done via the delete method:

```
DELETE /api/tenants/:id
```

or via the **delete** action as follows:

```
POST /api/tenants/:id

{
  "action" : "delete"
}
```

or by href:

```
POST /api/tenants

{
  "action" : "delete",
  "resources" : [
```

```
    { "href" : "http://localhost:3000/api/tenants/:id" }
  ]
}
```

Creating Tenant Quotas

Creating a quota for a specific tenant is done as follows:

```
POST /api/tenants/:id/quotas
```

```
{
  "name" : "cpu_allocated",
  "value" : 1
}
```

Editing Tenant Quotas

```
POST /api/tenants/:id/quotas/:quota_id
```

```
{
  "action" : "edit",
  "resource" : {
    "value" : 5
  }
}
```

Multiple quotas for a tenant can also be edited in a single request as follows:

```
POST /api/tenants/1/quotas
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/tenants/1/quotas/5",
      "value" : 11
    },
    { "href" : "http://localhost:3000/api/tenants/1/quotas/6",
      "value" : 12
    },
    ...
  ]
}
```

Deleting Tenant Quotas

Quotas of a tenant can be deleted via the *delete* action as follows:

```
POST /api/tenants/:id/quotas/:quota_id
```

```
{
  "action" : "delete"
}
```

or via the DELETE HTTP method

```
DELETE /api/tenants/:id/quotas/:quota_id
```

Multiple quotas of a tenant can be deleted in a single request as shown in the following example:

```
POST /api/tenants/1/quotas
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/tenants/1/quotas/7" },
    { "href" : "http://localhost:3000/api/tenants/1/quotas/8" },
    ...
  ]
}
```


- User Management
  - Querying Users
  - Creating Users
  - Editing Users
  - Editing Authenticated User
  - Deleting Users

User Management

Management of Users is provided via the following collection:

```
/api/users
```

Full CRUD actions on users is available:

- Querying Users

- Creating Users
- Editing Users
- Editing Authenticated User
- Deleting Users

Querying Users

Querying all users in the system is simply:

```
GET /api/users
```

Getting details on a specific user:

```
GET /api/users/:id
```

Creating Users

Users can be created via a POST to the users collection or via the **create** action signature which also allows creation of multiple users in a single request.

```
POST /api/users

{
  "userid" : "testuser",
  "password" : "secret",
  "name" : "Test User",
  "group" : { "id" : "11" }
}
```

Note:

group can be specified via **description**, **href**, or **id**

or creating multiple users:

```
{
  "action" : "create",
  "resources" : [
    { "userid" : "testuser1", "name" : "Test User1", ... },
    { "userid" : "testuser2", "name" : "Test User2", ... },
    ...
  ]
}
```

Note:

Please refer to the Resource Attributes page for a list of available attributes when creating Users.

Editing Users

```
POST /api/users/:id

{
  "action" : "edit",
  "resource" : {
    "group" : { "id" : "12" }
  }
}
```

or editing multiple users:

```
POST /api/users

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/users/101",
      "group" : { "href" : "http://localhost:3000/api/groups/15" }
    },
    {
      "href" : "http://localhost:3000/api/users/102",
      "group" : { "href" : "http://localhost:3000/api/groups/15" }
    },
    {
      "href" : "http://localhost:3000/api/users/103",
      "group" : { "href" : "http://localhost:3000/api/groups/15" }
    },
    ...
  ]
}
```

Editing Authenticated User

An API Authenticated User can edit some of their own attributes without having general user edit permissions. Editable attributes for the authenticated user are:

password
email
settings

Users can also update their currently selected group with the **set_current_group** action as follows:

```
POST /api/users/:id

{
  "action" : "set_current_group",
  "resource" : {
    "group" : { "description" : "EvmGroup-auditor" }
  }
}
```

Note:

The group reference can be via **description**, **href** or **id**. Also, only a group that the user belongs to can be specified.

Deleting Users

users can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/users/101

{
  "action" : "delete"
}
```

or simply:

```
DELETE /api/users/101
```

Deleting multiple users can be done as follows:

```
POST /api/users

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/users/101" },
    { "href" : "http://localhost:3000/api/users/102" },
    ...
  ]
}
```


- Virtual Machine Management
  - Targeting Virtual Machines
  - Querying Virtual Machines
  - Starting a Virtual Machine
  - Stopping a Virtual Machine
  - Suspending a Virtual Machine
  - Refreshing a Virtual Machine
  - Retiring a Virtual Machine
  - Resetting a Virtual Machine
  - Rebooting guest in Virtual Machine
  - Shutting Down Guest in a Virtual Machine
  - Scanning a Virtual Machine
  - Setting Server of a Virtual Machine
  - Setting Owner of a Virtual Machine
  - Adding an Event to a Virtual Machine
  - Editing a Virtual Machine
  - Deleting a Virtual Machine

Virtual Machine Management

Management of virtual machines adds support of the following actions:

Action	Description
start	Starts a virtual machine
stop	Stops a virtual machine
suspend	Suspends a virtual machine
refresh	Refreshes a virtual machine
request_retire	Retires a virtual machine
reset	Resets a virtual machine
reboot_guest	Reboots guest in virtual machine
shutdown_guest	Shuts down guest in virtual machine
scan	Scans a virtual machine (Perform SmartState Analysis)

Action	Description
set_miq_server	Sets the server of a virtual machine
set_owner	Sets the owner of a virtual machine
add_event	Adding an event to a virtual machine
edit	Edits a virtual machine
delete	Deletes a virtual machine in the appliance

- Targeting VMs
- Querying VMs
- Starting a VM
- Stopping a VM
- Suspending a VM
- Refreshing a VM
- Retiring a VM
- Resets a VM
- Reboots guest in VM
- Shuts down guest in VM
- Scanning a VM
- Setting Server of a VM
- Setting Owner of a VM
- Adding an Event to a VM
- Editing a VM
- Deleting a VM

Targeting Virtual Machines

These actions can be triggered on individual virtual machine resources:

```
/api/vms/:id
```

As simply as POSTing the following action to a virtual machine.

```
{
  "action" : "start"
}
```

Requests can also be made on multiple vms by targetting the primary collection:

```
/api/vms
```

```
{
  "action" : "start",
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/11" },
    { "href" : "http://localhost:3000/api/vms/12" },
    ...
  ]
}
```

Querying Virtual Machines

Virtual machines are queried via the primary collection URL:

```
/api/vms
```

Filtering, sorting and paging as mentioned on the Querying page.

When querying vms, expanding the resources themselves as well as the following subcollections:

accounts
software

For example:

```
GET /api/vms?expand=resources,accounts,software
```

Or querying an individual virtual machine

```
GET /api/vms/:id?expand=accounts,software
```

Starting a Virtual Machine

```
{
  "action" : "start"
}
```

Stopping a Virtual Machine

```
{
  "action" : "stop"
}
```

Suspending a Virtual Machine

```
{
  "action" : "suspend"
}
```

Refreshing a Virtual Machine

```
{
  "action" : "refresh"
}
```

Retiring a Virtual Machine

```
{
  "action" : "request_retire"
}
```

Resetting a Virtual Machine

```
{
  "action" : "reset"
}
```

Rebooting guest in Virtual Machine

```
{
  "action" : "reboot_guest"
}
```

Shutting Down Guest in a Virtual Machine

```
{
  "action" : "shutdown_guest"
}
```

Scanning a Virtual Machine

```
{
  "action" : "scan"
}
```

Setting Server of a Virtual Machine

```
{
  "action" : "set_miq_server",
  "resource" : {
    "miq_server" : { "href" : "http://localhost:3000/api/servers/5" }
  }
}
```

```
{
  "action" : "set_miq_server",
  "resource" : {
    "miq_server" : { "id" : "6" }
  }
}
```

To remove the server from the virtual machine, pass in an empty reference as follows:

```
{
  "action" : "set_miq_server",
  "resource" : {
    "miq_server" : {}
  }
}
```

Setting Owner of a Virtual Machine

```
{
  "action" : "set_owner",
  "resource" : {
    "owner" : "admin"
  }
}
```

Adding an Event to a Virtual Machine

```
{
  "action" : "add_event",
  "resource" : {
    "event_type" : "...",
    "event_message" : "...",
    "event_time" : "UTC Time"
  }
}
```

Note:

event_time above is optional. If skipped, current time will be used.

Editing a Virtual Machine

Basic information of virtual machines can be edited. This includes the following:

VM Info	Attribute
description	description
custom attributes	custom_1 ... custom_9
parent resource	parent_resource - resource href reference
child resources	child_resources - array of resource href references

Virtual machine resources can be edited as follows:

```
POST /api/vms/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated VM Description",
    "custom_1" : "custom_attribute_1",
    "parent_resource" : { "href" : "http://localhost:3000/api/vms/11" },
    "child_resources" : [
      { "href" : "http://localhost:3000/api/vms/101" },
      { "href" : "http://localhost:3000/api/vms/102" },
      { "href" : "http://localhost:3000/api/vms/103" },
      { "href" : "http://localhost:3000/api/vms/104" }
    ]
  }
}
```

Virtual machines can also be edited in bulk as follows:

```
POST /api/vms
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/vms/11",
      "custom_9" : "vm_class_a"
    },
    {
      "href" : "http://localhost:3000/api/vms/12",
      "custom_9" : "vm_class_a"
    },
    {
      "href" : "http://localhost:3000/api/vms/13",
      "custom_9" : "vm_class_a"
    }
  ]
}
```

Deleting a Virtual Machine

```
{
  "action" : "delete"
}
```

Or simply doing the following:

```
DELETE /api/vms/:id
```

See the main REST API examples section for additional virtual machine management examples.

Infrastructure Management API Appendices

In this guide you will find appendices for the Infrastructure Automation – Infrastructure Management REST API.

- Quick Reference
- Collection Metadata
- Resource Attributes

- Provider Types
- Provision Request Supported Attributes
- Advanced Provisioning Attributes
- SOAP to REST Mapping guide
- Deprecated Types
- Deprecated Resource Attributes

- Authentication
- HTTP Headers
- Listing and Querying Collections and Sub-Collections
- Collection Queries:
- Sub-Collection Queries

This document presents the currently available features and capabilities of the REST API.

Authentication

Type	Mechanism
Basic Authentication	Basic HTTP Authorization with user and password
Token Based Authentication	
- Acquiring Token	/api/auth with Basic Authentication
- Authenticating with Token	X-Auth-Token Header

HTTP Headers

Header	Value
Authorization	Basic base64_encoded(user:password)
X-Auth-Token	Token provided by /api/auth
X-MIQ-Group	Group to authorize externally authenticated user
Accept	application/json
Content-Type	application/json

Listing and Querying Collections and Sub-Collections

Feature	Path
Listing Available Collections	/api
Listing Collections	/api/<collection>
Listing Sub-Collections	/api/<collection>/<id>/<sub-collection>

Querying Capability	Query Parameters	
Paging	offset, limit	
Sorting	sort_by=attr, sort_order=asc	desc
	Can sort by database columns and SQL friendly virtual attributes	
Filtering	filter[]=...	
Querying by Tag	i.e. by_tag=/department/finance	
Querying by Multiple Tags	i.e. by_tag=/department/finance,/department/engineering	
Expanding Results	expand=<what>, i.e. expand=resources,tags,service_templates,...	
Selecting Attributes	attributes=<attr1>,<attr2>,... i.e. attributes=id,name,type,...	
	Attributes can be:	
	Database columns	
	Virtual attributes	
	Relationships	

Example Queries

Collection Queries:

For a complete list of supported primary collections please refer to the Primary Collections

Sub-Collection Queries

When applicable, the following subcollections are available:

Sub-Collection	URL
Tagging	/api/<collection>/:id/tags
Policies	/api/<collection>/:id/policies
Policy Profiles	/api/<collection>/:id/policy_profiles
Service Requests	/api/service_templates/:id/service_requests

Sub-Collection	URL
Request Tasks	
	/api/service_requests/:id/request_tasks
	/api/automation_requests/:id/request_tasks
	/api/provision_requests/:id/request_tasks
Request Tasks can also be accessed via the tasks alias	
	/api/service_requests/:id/tasks
	/api/automation_requests/:id/tasks
	/api/provision_requests/:id/tasks

- Collection Metadata
  - Optional Metadata

Collection Metadata

Collection specific metadata can be queried by sending an HTTP OPTIONS request on the specific collection:

```
OPTIONS /api/:collection
```

What gets returned is the following structure:

```
{
  "attributes": [
  ],
  "virtual_attributes": [
  ],
  "relationships": [
  ],
  "subcollections": [
  ],
  "data": {
  }
}
```

Entry	Description
attributes	Physical attributes of the collection
virtual_attributes	Virtual attributes of the collection
relationships	Physical or Virtual associations to other resources
subcollections	Subcollections that can be queried
data	Optional meta-data for the collection

The following is an example for the **groups** collection:

```
OPTIONS /api/groups
```

```
{
  "attributes": [
    "created_on",
    "description",
    "group_type",
    "id",
    "sequence",
    "settings",
    "tenant_id",
    "updated_on"
  ],
  "virtual_attributes": [
    "allocated_memory",
    "allocated_storage",
    "allocated_vcpu",
    "custom_1",
    "custom_2",
    "custom_3",
    "custom_4",
    "custom_5",
    "custom_6",
    "custom_7",
    "custom_8",
    "custom_9",
    "href_slug",
    "miq_user_role_name",
    "provisioned_storage",
    "read_only",
    "region_description",
    "region_number",
    "user_count"
  ],
  "relationships": [
    "active_vms",
    "custom_action_buttons",
    "custom_actions",
    "custom_attributes",
    "custom_button_sets",
    "custom_buttons",
```

```
    "entitlement",
    "miq_custom_attributes",
    "miq_report_results",
    "miq_reports",
    "miq_templates",
    "miq_user_role",
    "miq_widget_contents",
    "miq_widget_sets",
    "taggings",
    "tags",
    "tenant",
    "users",
    "vms"
  ],
  "subcollections": [
    "tags"
  ],
  "data": {
  }
}
```

Optional Metadata

Most collections do not declare additional metadata as shown by the empty *data* hash entry above. Optional metadata is made available as needed for API clients. They are generally used to provide additional information needed for user-interfaces, help complete or validate data for additional API calls, etc.

Currently, optional metadata is made available for the following collections:

Collection
/api/authentications
/api/clusters
/api/container_deployments
/api/custom_buttons
/api/generic_object_definitions
/api/hosts
/api/providers

There is no enforced structure for the *data* element as each collection may need to provide different types of metadata. The following is a partial meta-data example:

```
OPTIONS /api/generic_object_definitions
```

```
{
  "attributes": [
    "created_at",
    "description",
    "id",
    "name",
    "properties",
    "updated_at"
  ],
  "virtual_attributes": [
    "generic_objects_count",
    "href_slug",
    "region_description",
    "region_number"
  ],
  "relationships": [
    "custom_action_buttons",
    "custom_actions",
    "custom_button_sets",
    "custom_buttons",
    "generic_objects",
    "picture"
  ],
  "subcollections": [
    "generic_objects"
  ],
  "data": {
    "allowed_association_types": [
      [
        "Availability Zone",
        "AvailabilityZone"
      ],
      ...
      [
        "VM and Instance",
        "Vm"
      ],
      [
        "Zone",
        "Zone"
      ]
    ],
    "allowed_types": [
      [
        "Boolean",
        "boolean"
```

```
      ],
      [
        "Date/Time",
        "datetime"
      ],
      [
        "Float",
        "float"
      ],
      [
        "Integer",
        "integer"
      ],
      [
        "String",
        "string"
      ],
      [
        "Time",
        "time"
      ]
    ]
  }
}
```

- Resource Attributes
  - Actions
    - Required attributes:
    - Optional attributes:
  - Alert Definitions
    - Required attributes:
    - Optional attributes:
  - Authentications
    - Required attributes:
    - Optional attributes:
  - Category Attributes
    - Required attributes:
    - Optional attributes:
  - Chargeback Rate Attributes
    - Required attributes:
    - Optional attributes:
  - Conditions
    - Required attributes:
    - Optional attributes:
  - Configuration Script Sources
    - Required attributes:
    - Optional attributes:
  - Generic Objects
    - Required attributes:
    - Required attributes:
    - Optional attributes:
  - Generic Object Definitions
    - Required attributes:
    - Optional attributes:
  - Group Attributes
    - Required attributes:
    - Optional attributes:
      - Filters hash:
  - Orchestration Templates
    - Required attributes:
    - Optional attributes:
  - Policies
    - Required attributes:
  - Provider Attributes
    - Required attributes:
    - Optional attributes:
  - Role Attributes
    - Required Attributes:
    - Optional Attributes:
  - Service Attributes
    - Required attributes:
    - Optional attributes:
  - Service Catalog Attributes
    - Optional attributes:
  - Service Dialogs
    - Required attributes:

        - Optional attributes:
    - Tag Attributes
        - Required attributes:
    - Tenant Attributes
        - Required attributes:
        - Optional attributes:
    - User Attributes
        - Required attributes:
        - Optional attributes:

Resource Attributes

This page lists the available attributes when creating resources with the REST API

- Actions
- Alert Definitions
- Authentications
- Categories
- Chargeback Rates
- Conditions
- Configuration Script Sources
- Generic Objects
- Generic Object Definitions
- Groups
- Orchestration Templates
- Policies
- Providers
- Roles
- Services
- Service Catalogs
- Service Dialogs
- Tags
- Tenants
- Users

Actions

`POST /api/actions`

Required attributes:

Attribute	Type	Description
action_type	string	Action Type
description	string	Description (unique)

Optional attributes:

Attribute	Type	Description
name	string	Name
options	hash	Options

Alert Definitions

`POST /api/alert_definitions`

Required attributes:

Attribute	Type	Description
description	string	Description
db	string	Database

Attribute	Type	Description
expression	hash	Expression
options	hash	Options

Optional attributes:

Attribute	Type	Description
enabled	boolean	Enable the Alert Definition or not - Defaults to true

Authentications

`POST /api/authentications`

Required attributes:

Attribute	Type	Description
manager_resource	hash	**href** of the provider

Optional attributes:

Attribute	Type	Description
type	string	Authentication class

Note:

For a complete list of authentication classes supported and the related attributes for each type, send the *OPTIONS /api/authentications* request.

Category Attributes

`POST /api/categories`

Required attributes:

Attribute	Type	Description
name	string	Name
description	string	Description

Optional attributes:

Attribute	Type	Description
example_text	string	Detailed description
show	boolean	Show Category flag
single_value	boolean	Single Value flag
syntax	string	Syntax of value, **string**, **integer** or **boolean**

Chargeback Rate Attributes

`POST /api/rates`

Required attributes:

Attribute	Type	Description
chargeback_rate_id	integer	Reference to parent Chargeback. **<chargeback_rate>** can be used instead.
group	string	Group rate belongs to, i.e. **cpu**, **memory**, **net_io**, **disk_io**, etc.
source	string	The input value for calculation, i.e. **allocated**, **used**, etc.
chargeback_tiers	Array of hashes	Each tier is represented by hash `{“start”: 0.0, “finish”: ‘infinity’, “fixed_rate”: 2.0, “variable_rate”: 0.0}]

Optional attributes:

Attribute	Type	Description
description	string	Description of the chargeback rate
enabled	boolean	Rate enabled flag
friendly_rate	string	Friendly name of the rate
metric	string	Metrics, i.e. **derived_memory_available**, etc.
per_time	string	Measured per time, **hourly**, **daily**, **weekly**, **monthly** or **yearly**
per_unit	string	Measured per unit, i.e. **megabytes**, **gigabytes**, etc.
chargeback_rate	hash	**id** or **href** from **api/chargebacks** collection.
detail_currency	hash	**id** or **href** from **api/currencies** collection. Details of related currency.
chargeable_field	hash	**id** or **href** from **api/chargeable_fields** collection. Record representing metric. Contains equivalent columns: **metric, group, source**

Conditions

`POST /api/conditions`

Required attributes:

Attribute	Type	Description
description	string	Condition Description
expression	hash	Expression
modifier	string	Modifier
towhat	string	Class name

Optional attributes:

Attribute	Type	Description
name	string	Condition name

Configuration Script Sources

`POST /api/configuration_script_sources`

Required attributes:

Attribute	Type	Description
manager_resource	hash	**href** or **id** of the provider

Optional attributes:

Attribute	Type	Description
description	string	Description
name	string	Project Name
related	hash	Related attributes and values

Generic Objects

`POST /api/generic_objects`

Required attributes:

Attribute	Type	Description
name	string	Name of the generic object
generic_object_definition	hash	Generic object definition to create new generic object for, requires an **href** or **id** attribute in the hash

`POST /api/generic_object_definitions/:id/generic_objects`

Required attributes:

Attribute	Type	Description
name	string	Name of the generic object

Optional attributes:

Attribute	Type	Description
uid	string	Uid string
property_attributes	hash	Attribute/Value pairs to define for the generic object
associations	hash	Hash containing association keys each being an array of resource references

Generic Object Definitions

Required attributes:

Attribute	Type	Description
name	string	Name of the generic object definition

Optional attributes:

Attribute	Type	Description
description	string	Description for the generic object definition
properties	hash	Hash containing the **attributes**, **associations** and **methods** of the generic object definition
picture	hash	Picture to use, either an **href** or **id** of the picture resource, or the actual picture including the **extension** and Base64 encoded **content**

Group Attributes

`POST /api/groups`

Required attributes:

Attribute	Type	Description
description	string	Description
role	hash	Role the group is assigned to, requires an **href**, **id** or **name** attribute in the hash
tenant	hash	Tenant the group belongs to, requires an **href** or **id** attribute in the hash

Optional attributes:

Attribute	Type	Description
filters	hash	Hash that contains the **belongsto** and **managed** arrays of filters to assign to the group
filter_expression	hash	Hash representing the expression used for filtering by Tag

Filters hash:

The `filters` hash can include the following keys:

- **belongsto**
  - Array of tags representing the following types of objects:
    - VMs & Templates
    - Clusters, Datastores, Hosts, Managers & Providers
- **managed**
  - Array of tag arrays representing:
    - My Company Tags

Example:

```
"filters" : {
  "managed" : [
    ["/managed/location/ny"],
    ["/managed/environment/prod"]
  ],
  "belongsto" : [
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|vm/EmsFolder|Alberto-
Dev,
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|host/EmsCluster|dev-
vc60-cluster"
  ]
}
```

Alternatively, company tags can be represented with an expression via the *filter_expression* instead of the *managed* tag array in the *filters* element as follows:

Example with Tags based on expression:

```
"filters" : {
  "managed" : null,
  "belongsto" : [
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|vm/EmsFolder|Alberto-
Dev,
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|host/EmsCluster|dev-
vc60-cluster"
  ]
},
"filter_expression" : {
  "exp": {
    "and": [
      {
        "CONTAINS": {
          "tag": "managed-location",
          "value": "ny"
        }
      },
      {
        "CONTAINS": {
          "tag": "managed-environment",
          "value": "prod"
        }
      }
    ]
  }
}
```

Orchestration Templates

```
POST /api/orchestration_templates
```

Required attributes:

Attribute	Type	Description

Attribute	Type	Description
name	string	Name
content	text	Template content

Optional attributes:

Attribute	Type	Description
type	string	Type
description	string	Description
draft	boolean	**true** or **false**
ems_id	integer	Id of the provider
orderable	boolean	**true** or **false**

Policies

`POST /api/policies`

Required attributes:

Attribute	Type	Description
name	string	Name
description	string	Description
mode	string	Mode (i.e. *compliance*)
towhat	string	Target Class
condition_ids	array	Ids of Conditions to assign to the Policy
policy_contents	array	Array of hashes including *event_id* and *actions*, an array of hashes comprising of an *action_id* and *opts*.

Provider Attributes

`POST /api/providers`

Required attributes:

Attribute	Type	Description
name	string	Name of the provider
type	string	Provider class
hostname	string	Hostname of the Provider
credentials or connection_configurations	hash or array	Credentials information for the provider either via the **credentials** hash or **connection_configurations** array

Optional attributes:

Attribute	Type	Description
api_version	string	API Version for communicating with the Provider
uid_ems	string	Domain for OpenStack provider. Required when “api_version” is “v3”
certificate_authority	string	CA for the Provider
connection_configurations	array	Endpoints for the Provider
credentials	hash	Credentials to use for communicating with the Provider, see Provider Support for examples
host_default_vnc_port_start	integer	Starting VNC port
host_default_vnc_port_end	integer	Ending VNC port
ipaddress	string	IP Address of the Provider
port	string	Port of the Provider
realm	string	Realm of the Provider
security_protocol	string	Security Protocol
url	string	URL of the Provider
zone	hash	In which zone to add provider to, hash needs to include an **href** or **id** attribute.

Credentials hash specification attributes are:

Type	Auth Types	Auth Attributes
ManageIQ::Providers::Amazon::CloudManager	default	userid, password
ManageIQ::Providers::Atomic::ContainerManager	default, password, bearer	userid, password
ManageIQ::Providers::Azure::CloudManager	default	userid, password
ManageIQ::Providers::Kubernetes::ContainerManager	default, password, bearer	userid, password
ManageIQ::Providers::Microsoft::InfraManager	default	userid, password
ManageIQ::Providers::Openshift::ContainerManager	default, password, bearer	userid, password, auth_key
ManageIQ::Providers::Openstack::InfraManager	default, amqp, ssh_keypair	userid, password, auth_key
ManageIQ::Providers::Openstack::CloudManager	default, amqp	userid, password
ManageIQ::Providers::Redhat::InfraManager	default, metrics	userid, password
ManageIQ::Providers::Vmware::InfraManager	default	userid, password

Type	Auth Types	Auth Attributes
ManageIQ::Providers::Foreman::Provider	default	userid, password
ManageIQ::Providers::Openstack::Provider	default	userid, password

Role Attributes

`POST /api/roles`

Required Attributes:

Attribute	Type	Description
name	string	Name of user role

Optional Attributes:

Attribute	Type	Description
features	array	Features to assign to the role. Array of { “identifier” OR “href” OR “id” : ... }
settings	hash	Settings for the role, normally: { “restrictions” : { “vms” : “user” OR “user_or_group” } }

Service Attributes

`POST /api/services`

Required attributes:

Attribute	Type	Description
name	string	Name

Optional attributes:

Attribute	Type	Description
description	string	Description
orchestration_template	hash	**href** or **id** of the Orchestration Template
orchestration_manager	hash	**href** or **id** of the Orchestration Provider
parent_service	hash	Parent service **href** or **id** reference
type	string	Type of Service

Service Catalog Attributes

`POST /api/service_catalogs`

Attribute	Type	Description
name	string	Name of service catalog
description	string	Description of service catalog

Optional attributes:

Attribute	Type	Description
service_templates	array	Array of Service Template hashes to assign to the new Service Catalog. Hash entries to include the **href** of the service_templates to assign and must not currently be assigned to any Service Catalog.

Service Dialogs

`POST /api/service_dialogs`

Required attributes:

Attribute	Type	Description
label	string	Label
dialog_tabs	array	Dialog content - array of dialog tabs

Optional attributes:

Attribute	Type	Description
description	string	Description

Tag Attributes

`POST /api/tags`

Required attributes:

Attribute	Type	Description
name	string	Name of tag

Attribute	Type	Description
description	string	Description of tag
category	hash	Category identifier for which the tag is getting added to, requires an **href**, **id** or **name** attribute in the hash

Note:

When adding with a **create** action on the tags subcollection of the category via /api/categories/:id/tags, the category should not be specified in the resource.

Tenant Attributes

```
POST /api/tenants
```

Required attributes:

Attribute	Type	Description
name	string	Name of Tenant
parent	hash	Parent tenant identifier for which the sub-tenant is getting added to, requires an **href** or **id** attribute in the hash

Optional attributes:

Attribute	Type	Description
description	string	Description of Tenant
domain	string	Domain
logo_file_name	string	Logo file name, must be available via /uploads/**logo_file_name**
login_logo_file_name	string	Login Logo file name, must be available via /uploads/**login_logo_file_name**
login_text	string	Login text
subdomain	string	Subdomain

User Attributes

```
POST /api/users
```

Required attributes:

Attribute	Type	Description
userid	string	Username
password	string	Password
name	string	Full Name
group	hash	Group the user belongs to, requires a **description**, **href** or **id** attribute in the hash

Optional attributes:

Attribute	Type	Description
email	string	E-mail Address

- Provider Types

Provider Types

The provider types available with earlier appliances have changed with this release. This is what gets specified with the *type* attribute when creating a new provider:

```
POST /api/providers
```

```
{
  "name" : "vCenter 5",
  "type" : "ManageIQ::Providers::Vmware::InfraManager",
  "hostname" : "my_vcenter_50",
  ...
}
```

The following table lists the type from previous appliance releases (*v2.0.0* of the REST API), and their mapping to the new *v2.1.0* types:

v2.0.0 Provider Type	v2.1.0 Provider Type
EmsAmazon	ManageIQ::Providers::Amazon::CloudManager
	ManageIQ::Providers::Atomic::ContainerManager (**new**)
	ManageIQ::Providers::Azure::CloudManager (**new**)
EmsKubernetes	ManageIQ::Providers::Kubernetes::ContainerManager
EmsMicrosoft	ManageIQ::Providers::Microsoft::InfraManager
	ManageIQ::Providers::Openshift::ContainerManager (**new**)
EmsOpenstackInfra	ManageIQ::Providers::Openstack::InfraManager
EmsOpenstack	ManageIQ::Providers::Openstack::CloudManager
EmsRedhat	ManageIQ::Providers::Redhat::InfraManager
EmsVmware	ManageIQ::Providers::Vmware::InfraManager

And when *provider-class* is set to **provider**, the following mapping table applies:

v2.0.0 Provider Type	v2.1.0 Provider Type
ProviderForeman	ManageIQ::Providers::Foreman::Provider
ProviderOpenstack	ManageIQ::Providers::Openstack::Provider

- Provision Request Supported Attributes
  - Requester Attributes
  - Custom Attributes
  - Service Catalog Attributes
  - Schedule Attributes
  - Network Attributes
  - Hardware Attributes
  - Environment Attributes

Provision Request Supported Attributes

Attribute Class	Provision Request Attribute Group
Requester Attributes	requester
Custom Attributes	vm_fields
Service Catalog Attributes	vm_fields
Schedule Attributes	vm_fields
Network Attributes	vm_fields
Hardware Attributes	vm_fields
Environment Attributes	additional_values

Requester Attributes

Attribute	Type	Description
owner_phone	string	Phone
owner_country	string	Country/Region
owner_phone_mobile	string	Mobile phone
owner_title	string	Title
owner_first_name	string	First name
owner_manager	string	Manager name
owner_address	string	Address
owner_company	string	Company
owner_last_name	string	Last name
owner_manager_email	string	Manager e-mail address
owner_city	string	City
owner_department	string	Department
owner_load_ldap	button	Look up LDAP e-mail address
owner_manager_phone	string	Manager phone
owner_state	string	State
owner_office	string	Office
owner_zip	string	Zip code
owner_email	string	E-Mail
request_notes	string	Notes

Custom Attributes

Attribute	Type	Description	Values	Default
dns_servers	string	DNS server list		
sysprep_organization	string	Organization		
sysprep_password	string	New Administrator Password		
sysprep_custom_spec	string	Name		
sysprep_server_license_mode	string	Identification	perServer, perSeat	
ldap_ous	string	LDAP Group		
sysprep_timezone	string	Timezone		
dns_suffixes	string	DNS Suffix list		
sysprep_product_id	string	ProductID		
sysprep_identification	string	Identification	domain, workgroup	
sysprep_per_server_max_connections	string	Maximum Connections		5
sysprep_computer_name	string	Computer Name		
sysprep_workgroup_name	string	Workgroup Name		WORKGROUP
sysprep_spec_override	boolean	Override Specification		false
addr_mode	string	Address Mode	static, dhcp	dhcp
linux_host_name	string	Computer Name		
sysprep_domain_admin	string	Domain Admin		

Attribute	Type	Description	Values	Default
sysprep_change_sid	boolean	Change SID		true
sysprep_domain_name	string	Domain Name		
sysprep_upload_file	string	Upload		
gateway	string	Gateway		
ip_addr	string	IP Address		
linux_domain_name	string	Domain Name		
sysprep_domain_password	string	Domain Password		
sysprep_auto_logon	boolean	Auto Logon		true
sysprep_enabled	string	Customize		disabled
sysprep_delete_accounts	boolean	Delete Accounts		false
sysprep_upload_text	string	Sysprep Text		
wins_servers	string	WINS Server list		
subnet_mask	string	Subnet Mask		
sysprep_full_name	string	Full Name		
sysprep_auto_logon_count	integer	Auto Logon Count	1, 2, 3	1
customization_template_id	integer	Script Name		
root_password	string	Root Password		
hostname	string	Host Name		
customization_template_script	string	Script Text		

Service Catalog Attributes

Attribute	Type	Description	Default
number_of_vms	integer	Count. *Max*: 50	1
vm_description	string	VM Description. *Max length*: 100	
vm_prefix	string	VM Name Prefix/Suffix	
src_vm_id	integer	Name	
vm_name	string	VM Name	
pxe_image_id	string	Image	
pxe_server_id	integer	Server	
host_name	string	Host Name	
provision_type	string	Provision Type	vmware
linked_clone	boolean	Linked Clone	false
snapshot	string	Snapshot	
vm_filter	integer	Filter	

Schedule Attributes

Attribute	Type	Description	Values	Default
schedule_type	string	When to Provision	schedule, immediately (*On Approval*)	immediately
vm_auto_start	boolean	Power on virtual machines after creation		true
schedule_time	time	Time to Provision on		
retirement	integer	Time until Retirement	0 (*Indefinite*), 1.month, 3.months, 6.months	0
retirement_warn	integer	Retirement Warning	1.week, 2.weeks, 30.days	1.week

Network Attributes

Attribute	Type	Description
vlan	string	vLan
mac_address	string	MAC Address

Hardware Attributes

Attribute	Type	Description	Values	Default
disk_format	string	Disk Format	thick, thin, unchanged	unchanged
cpu_limit	integer	CPU (MHz)		
memory_limit	integer	Memory (MB)		
number_of_sockets	integer	Number of Sockets	1, 2, 4, 8	1
cores_per_socket	integer	Cores per Socket	1, 2, 4, 8	1
cpu_reserve	integer	CPU (MHz)		
vm_memory	string	Memory (MB)	1024, 2048, 4096	1024
memory_reserve	integer	Memory (MB)		
network_adapters	integer	Network Adapters	1, 2, 3, 4	1

Environment Attributes

These attributes cannot be passed directly. To use these attributes, provide them in the **additional_values** attribute group and have customization methods use them and modify the request accordingly.

Attribute	Type	Description	Values	Default
new_datastore_grow_increment	integer	Grow Increment (GB)		
new_datastore_create	boolean	Create Datastore		false
placement_cluster_name	integer	Name		
new_datastore_aggregate	string	Aggregate		
new_datastore_max_size	integer	Max Size (GB)		
new_datastore_storage_controller	string	Controller		
cluster_filter	integer	Filter		
host_filter	integer	Filter		
ds_filter	integer	Filter		
new_datastore_volume	string	Volume		
placement_host_name	integer	Name		
placement_ds_name	integer	Name		
new_datastore_fs_type	string	FS Type	NFS, VMFS	NFS
rp_filter	integer	Filter		
new_datastore_thin_provision	string	Thin Provision		
placement_auto	boolean	Choose Automatically		false
new_datastore_size	integer	Size (GB)		
new_datastore_autogrow	string	Autogrow		false
placement_folder_name	integer	Name		
new_datastore_name	string	Name		
placement_rp_name	integer	Name		
placement_dc_name	integer	Name		

- Advanced Provisioning Attributes
  - Defining new vLans adapters
  - Defining new SCSI Controller
  - Defining new Disks
  - Cloning Type
  - Limit Template Selection

Advanced Provisioning Attributes

This appendix lists parameters that apply to REST API initiated provisioning via /api/provision_requests.

- Defining new vLans adapters
- Defining new SCSI Controller
- Defining new Disks
- Cloning Type
- Limit Template Selection

Defining new vLans adapters

These attributes allow adding new vLan adapters during the provisioning request.

Attribute	Type	Required	Values	Default
network##	string	Required	Network Name	
network##.mac_address	string	Optional	MAC Address	
network##.deviceType	string	Optional	VirtualE1000, VirtualPCNet32, VirtualVmxnet	VirtualPCNet32
network##.connectable.allowGuestControl	boolean	Optional		true
network##.connectable.startConnected	boolean	Optional		true
network##.connectable.connected	boolean	Optional		true

Example:

```
"network0" : "VM Network",
"network0.mac_address" : "00:50:56:57:58:59"
```

Defining new SCSI Controller

The SCSI Controller attributes are optional. If a controller is required, it will be added with the defined defaults. These settings are only required if you need to override a default value.

Attribute	Type	Required	Values	Default
ctrlscsi##	string	Required	VirtualBusLogicController, VirtualLsiLogicController	VirtualLsiLogicController
ctrlscsi##.sharedBus	string	Optional	noSharing, physicalSharing, virtualSharing	noSharing

Note:

The ## value defines the controller bus number

Example:

```
"ctrlscsi00" : "VirtualBusLogicController"
"ctrlscsi00.sharedBus" : "virtualSharing"
```

Defining new Disks

The Disk attributes allow adding disks as part of the provisioning request.

Attribute	Type	Required	Values	Default
diskscsi##	string	Required	“controller_#:controller_position_#:size_in_mb”	
diskscsi##.datastore	string	Optional	Datastore Name	
diskscsi##.filename	string	Optional	Filename	
diskscsi##.connectable.allowGuestControl	boolean	Optional		false
diskscsi##.connectable.startConnected	boolean	Optional		true
diskscsi##.connectable.connected	boolean	Optional		true
diskscsi##.backing.diskmode	string	Optional	persistent, independent_persistent, independent_nonpersistent, nonpersistent	persistent
diskscsi##.backing.split	boolean	Optional		false
diskscsi##.backing.thinProvisioned	boolean	Optional		false
diskscsi##.backing.writeThrough	boolean	Optional		false

Example:

```
"diskscsi0" : "0:0:10"
"diskscsi1" : "0:0:8"
"diskscsi1.datastore" : "storage1"
"diskscsi1.backing.diskmode" : "independent_persistent"
```

Cloning Type

This attribute sets the type of cloning request, either clone to vm or clone to template. It is an attribute to set with the *template_fields* hash.

Attribute	Type	Required	Values	Default
request_type	string	Optional	clone_to_vm, clone_to_template	clone_to_vm

Example:

```
"template_fields" : {
  "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742",
  "request_type" : "clone_to_template"
}
```

Limit Template Selection

This attribute value limits the template selection to the Datacenters contained in the specified cluster. It is an attribute to be specified with the *additional_values* hash.

Attribute	Type	Required
cluster	string	Optional

Example:

```
"additional_values" : {
  "request_id" : "87458",
  "cluster" : "sales_prod"
}
```


- SOAP to REST Mapping guide
  - Insight Web Services
  - Control Web Services
  - Automate Web Services

SOAP to REST Mapping guide

- Insight Web Services
- Control Web Services
- Automate Web Services

Insight Web Services

SOAP API	HTTP Method	REST API URLs	Actions	Examples
ClusterGetTags	GET	/api/cluster/:id/tags		
DatastoreGetTags	GET	/api/data_stores/:id/tags		
EmsGetTags	GET	/api/providers/:id/tags		
EVMDeleteVmByName	DELETE	/api/vms/:id		Delete VM
	POST	/api/vms/:id	delete	Delete VMs

SOAP API	HTTP Method	REST API URLs	Actions	Examples
EVMClusterList	GET	/api/clusters		
EVMDatastoreList	GET	/api/data_stores		
EVMHostList	GET	/api/hosts		
EVMResourcePoolList	GET	/api/resource_pools		
EVMPing	GET	/api		
EVMVmAccounts	GET	/api/vms/:id/accounts		
EVMVmList	GET	/api/vms		
	GET	/api/hosts/:id?attributes=vms		
	GET	/api/hosts?filter[]=guid='....'\&expand=resources\&attributes=vms		
EVMVmSoftware	GET	/api/vms:id/software		
FindClusterById	GET	/api/clusters/:id		
FindClustersById	GET	/api/clusters/:id		
FindDatastoreById	GET	/api/data_stores/:id		
FindDatastoresById	GET	/api/data_stores/:id		
FindEmsByGuid	GET	/api/providers?filter[]=guid='....'		
FindHostByGuid	GET	/api/hosts?filter[]='guid='....'"		
FindHostsByGuid	GET	/api/hosts?filter[]='guid='....'"		
FindResourcePoolById	GET	/api/resource_pools/:id		
FindResourcePoolsById	GET	/api/resource_pools		
	GET	/api/resource_pools?filter[]=id=:id\&filter[]=or id=:id&...		
FindVmByGuid	GET	/api/vms?filter[]=guid='....'		
FindVmsByGuid	GET	/api/vms?filter[]=guid='....'\&filter[]=or guid='....'&...		
GetClusterList	GET	/api/clusters		
	GET	/api/clusters?filter[]=ems_id=:id		
	GET	/api/providers/:id?attributes=ems_clusters		
GetClustersByList	GET	/api/clusters/:id		
	GET	/api/clusters?filter[]=id=:id\&filter[]=or id=:id&...		
GetClustersByTag	GET	/api/clusters?by_tag=:tag_path		
GetDatastoreByList	GET	/api/data_stores		
GetDatastoreByTag	GET	/api/data_stores?by_tag=:tag_path		
GetEmsByList	GET	/api/providers		
	GET	/api/providers/:id		
	GET	/api/providers?filter[]=....\&expand=resources		
GetEmsList	GET	/api/providers		
GetHostsByList	GET	/api/hosts		
	GET	/api/hosts?filter[]=....\&expand=resources		
GetHostList	GET	/api/hosts		
	GET	/api/providers/:id?attributes=hosts		
	GET	/api/providers?filter[]=guid='....'\&expand=resources\&attributes=hosts		
GetResourcePoolList	GET	/api/providers/:id?attributes=resource_pools		
	GET	/api/providers?filter[]=guid='....'\&expand=resources\&attributes=resource_pools		
GetResourcePoolsByList	GET	/api/resource_pools		
	GET	/api/resource_pools/:id		
	GET	/api/resource_pools?filter[]=....\&expand=resources		
GetResourcePoolsByTag	GET	/api/resource_pools?by_tag=:tag_path		
GetTemplatesByTag	GET	/api/templates?by_path=:tag_path		
GetVmList	GET	/api/hosts/:id?attributes=vms		
	GET	/api/hosts?filter[]=guid='....'\&expand=resources\&attributes=vms		
GetVmsByList	GET	/api/vms		
	GET	/api/vms/:id		
	GET	/api/vms?filter[]=....\&expand=resources		
GetVmsByTag	GET	/api/vms?by_path=:tag_path		
HostGetTags	GET	/api/hosts/:id/tags		
	GET	/api/hosts/:id?expand=tags		
	GET	/api/hosts?filter[]=guid='....'\&expand=resources,tags		
ResourcePoolGetTags	GET	/api/resource_pools/:id/tags		
	GET	/api/resource_pools/:id?expand=tags		
VmGetTags	GET	/api/vms/:id/tags		
	GET	/api/vms/:id?expand=tags		

Control Web Services

SOAP API	HTTP Method	REST API URLs	Actions	Examples
ClusterSetTag	POST	/api/clusters/:id/tags	assign	Assigning Tags
DatastoreSetTag	POST	/api/data_stores/:id/tags	assign	Assigning Tags

SOAP API	HTTP Method	REST API URLs	Actions	Examples
EmsSetTag	POST	/api/providers/:id/tags	assign	Assigning Tags
EVMActionList	GET	/api/policies/:id/policy_actions		Policy Actions
	GET	/api/policies/:id?expand=policy_actions		
	GET	/api/policies?filter[]=guid='... .'\&expand=resources,policy_actions		
EVMAssignPolicy	POST	/api/hosts/:id/policies	assign	Assigning Policies
EVMConditionList	GET	/api/policies/:id/conditions		Policy Conditions
	GET	/api/policies/:id?expand=conditions		
	GET	/api/policies?filter[]=guid='....'\&expand=resources,conditions		
EVMEventList	GET	/api/policies/:id/events		Policy Events
	GET	/api/policies/:id?expand=events		
	GET	/api/policies?filter[]=guid='....'\&expand=resources,events		
EVMGetPolicy	GET	/api/policies		Querying Policies
	GET	/api/policies/:id		
	GET	/api/policies?filter[]=name='....'		
EVMPolicyList	GET	/api/hosts/:id/policies		Policy Management
	GET	/api/hosts/:id?expand=policies		
	GET	/api/hosts?filter[]=guid='....'\&expand=resources,policies		
EVMVmEventByProperty	POST	/api/vms/:id	add_event	Add VM Event
EVMVmScanByProperty	POST	/api/vms/:id	scan	Scan VM
EVMSmartStart	POST	/api/vms/:id	start	Start VM
EVMSmartStop	POST	/api/vms/:id	stop	Stop VM
EVMSmartSuspend	POST	/api/vms/:id	suspend	Suspend VM
EVMUnassignPolicy	POST	/api/hosts/:id/policies	unassign	Unassigning Policies
EVMVmRsop	POST	/api/vms/:id/policies/:id	resolve	Resolving Policies
	POST	/api/vms/:id/policy_profiles/:id	resolve	Resolving Policy Profiles
HostSetTag	POST	/api/hosts/:id/tags	assign	Assigning Tags
ResourcePoolSetTag	POST	/api/resource_pools/:id/tags	assign	Assigning Tags
VmAddCustomAttribute	POST	/api/vms/:id/custom_attributes	add	Adding Custom Attributes
VmAddCustomAttributes	POST	/api/vms/:id/custom_attributes	add	
VmAddCustomAttributeByFields	POST	/api/vms/:id/custom_attributes	add	
VmDeleteCustomAttribute	POST	/api/vms/:id/custom_attributes	delete	Deleting Custom Attributes
VmDeleteCustomAttributes	POST	/api/vms/:id/custom_attributes	delete	
VmSetTag	POST	/api/vms/:id/tags	assign	Assigning Tags
VmSetOwner	POST	/api/vms/:id	set_owner	Set Owner of VM

Automate Web Services

SOAP API	HTTP Method	REST API URLs	Actions	Examples
CreateAutomationRequest	POST	/api/automation_requests	create	Create Automation Request, Create Automation Requests
EVMProvisionRequestEx	POST	/api/provision_requests	create	Create Provision Request, Create Provision Requests
GetAutomationRequest	GET	/api/automation_requests/:id		
GetAutomationTask	GET	/api/automation_requests/:id?expand=request_tasks		
	GET	/api/automation_requests/:id/request_tasks/:id		
GetVmProvisionRequest	GET	/api/provision_requests/:id		
	GET	/api/provision_requests/:id?expand=request_tasks		
VmProvisionRequest	POST	/api/provision_requests	create	Create Provision Request, Create Provision Requests

- Deprecated Types
  - API v2.4.0 ⇒ v3.0.0

Deprecated Types

This page lists the types that have been deprecated in previous releases and have been updated to newer types. They are *temporarily* supported in the listed releases but will be removed in future releases.

API v2.4.0 ⇒ v3.0.0

For Orchestration Templates:

Old Type	New Type

Old Type	New Type
OrchestrationTemplateAzure	ManageIQ::Providers::Azure::CloudManager::OrchestrationTemplate
OrchestrationTemplateHot	ManageIQ::Providers::Openstack::CloudManager::OrchestrationTemplate
OrchestrationTemplateCfn	ManageIQ::Providers::Amazon::CloudManager::OrchestrationTemplate
OrchestrationTemplateVnfd	ManageIQ::Providers::Openstack::CloudManager::VnfdTemplate

- Deprecated Resource Attributes
  - API v2.2.0 ⇒ v2.3.0

Deprecated Resource Attributes

This page lists the resource attributes that have been deprecated in earlier releases and have been either renamed to newer attribute names or simply removed.

API v2.2.0 ⇒ v2.3.0

For Hardware:

Old Attribute Name	New Attribute Name
numvcpus	cpu_sockets
memory_cpu	memory_mb

For Metrics:

Old Attribute Name	New Attribute Name
v_derived_logical_cpus_used	v_derived_cpu_total_cores_used

For Provision Workflows:

Old Attribute Name	New Attribute Name
allowed_datastore_aggregate	**removed**
allowed_datastore_storage_controller	**removed**

For all:

Old Attribute Name	New Attribute Name
cores_per_socket	cpu_cores_per_socket
logical_cpus	cpu_total_cores
aggregate_logical_cpus	aggregate_cpu_total_cores

Infrastructure Management API Examples

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Foundation
- Automation Requests
- Providers
- Provisioning Requests
- Service Catalogs
- Service Templates
- Services
- VM Management
- Tag Management
- Usage Flows

Infrastructure Management API Examples - Foundation

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Querying Examples
- Paging Queries

- Example Queries:
  - Query All VMs
  - Query A Specific VM
  - Query all VMs, but return only return name and vendor
  - Query VMs named sample* and return name and vendor
  - Query VMs but only return the first 500
  - Query VMs but return the second 500
  - Query first 1000 VMs, named test*, get name, vendor and guid and sort by name in ascending order
  - Query Services tagged for the finance department
  - Query Services tagged with the finance or engineering department

- Get details on the Tags of the first service
- Get details of the first service catalog and related details on the assigned service templates
- Get services requests created from a specific template and get details on all related request tasks.
- Get a specific provision request with expanded details on the associated provision request tasks

Example Queries:

Query All VMs

```
GET /api/vms
```

Query A Specific VM

```
GET /api/vms/1386
```

Query all VMs, but return only return name and vendor

```
GET /api/vms?expand=resources&attributes=name,vendor
```

Query VMs named sample* and return name and vendor

```
GET /api/vms?expand=resources&attributes=name,vendor&filter[]=name='sample%'
```

Query VMs but only return the first 500

```
GET /api/vms?offset=0&limit=500
```

Query VMs but return the second 500

```
GET /api/vms?offset=500&limit=500
```

Query first 1000 VMs, named test*, get name, vendor and guid and sort by name in ascending order

```
GET /api/vms?offset=0&limit=1000&filter[]=name='test%'
    &expand=resources&attributes=name,vendor,guid
    &sort_by=name&sort_order=asc
```

Query Services tagged for the finance department

```
GET /api/services?by_tag=/department/finance
```

Query Services tagged with the finance or engineering department

```
GET /api/services?by_tag=/department/finance,/department/engineering
```

Get details on the Tags of the first service

```
GET /api/services/1/tags?expand=resources
```

Get details of the first service catalog and related details on the assigned service templates

```
GET /api/service_catalogs/1?expand=service_templates
```

Get services requests created from a specific template and get details on all related request tasks.

```
GET /api/service_templates/25/service_requests?expand=resources,request_tasks
```

Get a specific provision request with expanded details on the associated provision request tasks

```
GET /api/provision_requests/120?expand=request_tasks
```


- Paging Queries
  - Request:
  - Response:
  - Request:
  - Response:
  - Request:
  - Response:
  - Request:
  - Response:

Paging Queries

In this series of requests, we will show the paging queries and the expected responses for each subsequent page.

Request:

```
GET /api/vms?offset=0&limit=500
    &sort_order=asc&sort_by=name
    &expand=resources&attributes=name
```

Response:


```
{
  "name": "vms",
  "count": 1912,
  "subcount": 500,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/176",
      "id": "176",
      "name": "53 Zone1"
    },
    ...
    {
      "href": "http://localhost:3000/api/vms/1575",
      "id": "1575",
      "name": "VmEmpty-3de98f0f-c6f3-4f8b-a932-554713a61067"
    }
  ],
  "actions": [
  ]
}
```


Request:

```
GET /api/vms?offset=500&limit=500
    &sort_order=asc&sort_by=name
    &expand=resources&attributes=name
```

Response:


```
{
  "name": "vms",
  "count": 1912,
  "subcount": 500,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/1574",
      "id": "1574",
      "name": "VmEmpty-3e13ff43-6907-4a22-8f95-58aeb1bffa0b"
    },
    ...
    {
      "href": "http://localhost:3000/api/vms/1076",
      "id": "1076",
      "name": "VmEmpty-9a885181-7771-4f91-9805-245c7606d833"
    }
  ],
  "actions": [
  ]
}
```


Request:

```
GET /api/vms?offset=1000&limit=500
    &sort_order=asc&sort_by=name
    &expand=resources&attributes=name
```

Response:


```
{
  "name": "vms",
  "count": 1912,
  "subcount": 500,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/1074",
      "id": "1074",
      "name": "VmEmpty-9ab9e101-92b0-4b6b-864e-e196538da8a8"
    },
    ...
    {
      "href": "http://localhost:3000/api/vms/575",
      "id": "575",
      "name": "VmEmpty-f251f135-01c8-4d44-b8e1-37b30844a9dd"
    }
  ],
  "actions": [
  ]
}
```


Request:

```
GET /api/vms?offset=1500&limit=500
    &sort_order=asc&sort_by=name
    &expand=resources&attributes=name
```

Response:

```
{
  "name": "vms",
  "count": 1912,
  "subcount": 412,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/574",
      "id": "574",
      "name": "VmEmpty-f28912f3-b096-487f-9763-97b39b67364b"
    },
    ...
    {
      "href": "http://localhost:3000/api/vms/1907",
      "id": "1907",
      "name": "yy_vm"
    }
  ],
  "actions": [
  ]
}
```

Note:

In this last request, the **subcount** was less than the requested page size thus denoting the last page of data being returned.

Infrastructure Management API Examples - Automation Requests

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Create an Automation Request
- Create multiple Automation Requests
- Trigger a single Automation Request
  - Request:
  - Request:
  - Response:

Trigger a single Automation Request

In the automation requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

```
POST /api/automation_requests
```

```
{
  "version" : "1.1",
  "uri_parts" : {
    "namespace" : "System",
    "class" : "Request",
    "instance" : "InspectME",
    "message" : "create"
  },
  "parameters" : {
    "var1" : "xxxxx",
    "var2" : "yyyyy",
    "var3" : 1024,
    "var4" : true,
    "var5" : "last value"
  },
  "requester" : {
    "user_name" : "admin",
    "auto_approve" : true
  }
}
```

Optionally, the action based request format is also supported:

Request:

```
POST /api/automation_requests
```

```
{
  "action" : "create",
  "resource" : {
    "version" : "1.1",
    "uri_parts" : {
      "namespace" : "System",
```

```
      "class" : "Request",
      "instance" : "InspectME",
      "message" : "create"
    },
    "parameters" : {
      "var1" : "xxxxx",
      "var2" : "yyyyy",
      "var3" : 1024,
      "var4" : true,
      "var5" : "last value"
    },
    "requester" : {
      "user_name" : "admin",
      "auto_approve" : true
    }
  }
}
```

Response:

```
{
  "results": [
    {
      "id": "12",
      "href" : "http://localhost:3000/api/automation_requests/12",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-16T21:49:55Z",
      "updated_on": "2015-04-16T21:49:55Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",
        "instance_name": "InspectME",
        "user_id": "1",
        "attrs": {
          "var1": "xxxxx",
          "var2": "yyyyy",
          "var3": 1024,
          "var4": true,
          "var5": "last value",
          "userid": "admin"
        }
      },
      "userid": "admin"
    }
  ]
}
```


- Trigger multiple Automation Requests
  - Request:
  - Response:

Trigger multiple Automation Requests

In the automation requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

```
POST /api/automation_requests
```

```
{
  "action" : "create",
  "resources" : [
    {
      "version" : "1.1",
      "uri_parts" :  {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
      },
      "parameters" : {
        "vm_name" : "test_1",
        "var2" : "yyyyy",
        "var3" : 1024,
        "var4" : true,
        "var5" : "last value"
      },
```

```
      "requester" :  {
        "user_name" : "jdoe",
        "auto_approve" : true
      }
    },
    {
      "uri_parts" :  {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
      },
      "parameters" : {
        "vm_name" : "test_2",
        "vm_memory" : 1024,
        "memory_limit" : 16384
      },
      "requester" :  {
        "auto_approve" : true
      }
    },
    {
      "uri_parts" :  {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
      },
      "parameters" : {
        "vm_name" : "test_3",
        "vm_memory" : 2048,
        "memory_limit" : 16384
      },
      "requester" :  {
        "auto_approve" : true
      }
    },
    {
      "uri_parts" :  {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
      },
      "parameters" : {
        "vm_name" : "test_4",
        "vm_memory" : 4096,
        "memory_limit" : 16384
      },
      "requester" :  {
        "auto_approve" : true
      }
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "id": "14",
      "href" : "http://localhost:3000/api/automation_requests/14",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-16T21:59:42Z",
      "updated_on": "2015-04-16T21:59:42Z",
      "requester_id": "13",
      "requester_name": "aab",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",
        "instance_name": "InspectME",
        "user_id": "13",
        "attrs": {
          "vm_name": "test_1",
          "var2": "yyyyy",
          "var3": 1024,
          "var4": true,
          "var5": "last value",
          "userid": "aab"
        }
      },
      "userid": "aab"
    },
```

```
    {
      "id": "15",
      "href" : "http://localhost:3000/api/automation_requests/15",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-16T21:59:42Z",
      "updated_on": "2015-04-16T21:59:42Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",
        "instance_name": "InspectME",
        "user_id": "1",
        "attrs": {
          "vm_name": "test_2",
          "vm_memory": 1024,
          "memory_limit": 16384,
          "userid": "admin"
        }
      },
      "userid": "admin"
    },
    {
      "id": "16",
      "href" : "http://localhost:3000/api/automation_requests/16",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-16T21:59:42Z",
      "updated_on": "2015-04-16T21:59:42Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",
        "instance_name": "InspectME",
        "user_id": "1",
        "attrs": {
          "vm_name": "test_3",
          "vm_memory": 2048,
          "memory_limit": 16384,
          "userid": "admin"
        }
      },
      "userid": "admin"
    },
    {
      "id": "17",
      "href" : "http://localhost:3000/api/automation_requests/17",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-16T21:59:42Z",
      "updated_on": "2015-04-16T21:59:42Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",
        "instance_name": "InspectME",
        "user_id": "1",
        "attrs": {
          "vm_name": "test_4",
          "vm_memory": 4096,
          "memory_limit": 16384,
          "userid": "admin"
        }
      },
      "userid": "admin"
    }
  ]
}
```


Infrastructure Management API Examples - Providers

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Create a Provider
- Create a Provider with multiple credentials
- Update a Provider
- Refresh a Provider
- Delete a Provider

- Create Provider
  - Request:
  - Response:

Create Provider

Request:

```
POST /api/providers
```

```
{
  "type"       : "ManageIQ::Providers::Redhat::InfraManager",
  "name"       : "rhevm101",
  "hostname"   : "rhevm101",
  "ipaddress"  : "100.200.300.101",
  "credentials" : {
    "userid"     : "admin_account",
    "password"   : "admin_password"
  }
}
```

Response:

```
{
  "results": [
    {
      "id": "105",
      "href" : "http://localhost:3000/api/providers/105",
      "name": "rhevm101",
      "hostname": "rhevm101",
      "ipaddress": "100.200.300.101",
      "created_on": "2015-05-05T15:47:41Z",
      "updated_on": "2015-05-05T15:47:41Z",
      "guid": "10360312-f33e-11e4-86c7-b8e85646e742",
      "zone_id": "1",
      "type": "ManageIQ::Providers::Redhat::InfraManager"
    }
  ]
}
```


- Create Provider
  - Request:
  - Response:

Create Provider

Request:

```
POST /api/providers
```

```
{
  "type"       : "ManageIQ::Providers::Redhat::InfraManager",
  "name"       : "rhevm102",
  "hostname"   : "rhevm102",
  "ipaddress"  : "100.200.300.102",
  "credentials" : [
    {
      "userid"     : "default_userid",
      "password"   : "default_password"
    },
    {
      "userid"     : "metrics_userid",
      "password"   : "metrics_password",
      "auth_type"  : "metrics"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "id": "106",
      "href" : "http://localhost:3000/api/providers/106",
      "name": "rhevm102",
      "hostname": "rhevm102",
      "ipaddress": "100.200.300.102",
```

```
      "created_on": "2015-05-05T15:57:44Z",
      "updated_on": "2015-05-05T15:57:44Z",
      "guid": "acbd610e-f3f6-11e4-aaba-b8e85646e742",
      "zone_id": "1",
      "type": "ManageIQ::Providers::Redhat::InfraManager"
    }
  ]
}
```

- Update Provider
  - Request:
  - Response:

Update Provider

Request:

```
POST /api/providers/106
```

```
{
  "action"    : "edit",
  "ipaddress" : "100.200.300.112",
  "credentials" : [
    {
      "userid"    : "updated_metrics_userid",
      "password"  : "updated_metrics_password",
      "auth_type" : "metrics"
    }
  ]
}
```

Response:

```
{
  "href": "http://localhost:3000/api/providers/106",
  "id": "106",
  "name": "rhevm102",
  "hostname": "rhevm102",
  "ipaddress": "100.200.300.112",
  "created_on": "2015-05-06T13:49:11Z",
  "updated_on": "2015-05-06T13:53:06Z",
  "guid": "acbd610e-f3f6-11e4-aaba-b8e85646e742",
  "zone_id": "1",
  "type": "ManageIQ::Providers::Redhat::InfraManager"
}
```

- Refresh Provider
  - Request:
  - Response:

Refresh Provider

Request:

```
POST /api/providers/105
```

```
{
  "action" : "refresh"
}
```

Response:

```
{
  "success": true,
  "message": "Provider id:105 name:'sample_provider' refreshing",
  "href": "http://localhost:3000/api/providers/105",
  "task_id" : "23",
  "task_href" : "http://localhost:3000/api/tasks/23",
  "tasks" : [
    { "id" : "23", "href" : "http://localhost:3000/api/tasks/23"},
    { "id" : "24", "href" : "http://localhost:3000/api/tasks/24"}
  ]
}
```

Note: In the above example a multi-manager provider was refreshed and in the response, a task id and href pair is returned for each of the provider managers. This allows the caller to check the status of each manager refresh for completion.

- Delete Provider
  - Deleting a single Provider
    - Request:
    - Response:
    - Request:
    - Response:
  - Deleting multiple Providers

- Request:
- Response:

Delete Provider

Deleting a single Provider

Request:

```
DELETE /api/provider/105
```

Or via the delete action as follows:

```
POST /api/provider/105

{
  "action" : "delete"
}
```

Response:

```
{
  "success": true,
  "message": "Provider id:106 name:'rhevm102' deleting",
  "task_id": "625",
  "task_href": "http://localhost:3000/api/tasks/625",
  "href": "http://localhost:3000/api/providers/106"
}
```

Note:

Delete actions are done asynchronously as it could take a while to complete. The delete task can be queried if needed as follows

Request:

```
GET /api/tasks/625
```

Response:

```
{
  "href": "http://localhost:3000/api/tasks/625",
  "id": "625",
  "name": "Provider id:106 name:'rhevm102' deleting",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2015-05-06T14:02:26Z",
  "updated_on": "2015-05-06T14:02:32Z"
}
```

Deleting multiple Providers

Request:

```
POST /api/providers

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/providers/107" },
    { "href" : "http://localhost:3000/api/providers/108" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Provider id:107 name:'rhevm102' deleting",
      "task_id": "626",
      "task_href": "http://localhost:3000/api/tasks/626",
      "href": "http://localhost:3000/api/providers/107"
    },
    {
      "success": true,
      "message": "Provider id:108 name:'rhevm103' deleting",
      "task_id": "627",
      "task_href": "http://localhost:3000/api/tasks/627",
      "href": "http://localhost:3000/api/providers/108"
    }
  ]
}
```

Infrastructure Management API Examples - Provisioning Requests

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Create a Provision Request
- Create multiple Provision Requests
- Trigger a single Provision Request
  - Request:
  - Response:

Trigger a single Provision Request

In the provision requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

```
POST /api/provision_requests
```

```
{
  "version" : "1.1",
  "template_fields" : {
    "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742"
  },
  "vm_fields" : {
    "number_of_cpus" : 1,
    "vm_name" : "aab_rest_vm1",
    "vm_memory" : "1024",
    "vlan" : "rhevm"
  },
  "requester" : {
    "user_name" : "jdoe",
    "owner_first_name" : "John",
    "owner_last_name" : "Doe",
    "owner_email" : "jdoe@sample.com",
    "auto_approve" : true
  },
  "tags" : {
    "network_location" : "Internal",
    "cc" : "001"
  },
  "additional_values" : {
    "request_id" : "1001"
  },
  "ems_custom_attributes" : { },
  "miq_custom_attributes" : { }
}
```

Optionally, the action based request format is also supported:

```
POST /api/provision_requests
```

```
{
  "action" : "create",
  "resource" : {
    "version" : "1.1",
      "template_fields" : {
        "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742"
      },
      "vm_fields" : {
        "number_of_cpus" : 1,
        "vm_name" : "aab_rest_vm1",
        "vm_memory" : "1024",
        "vlan" : "rhevm"
      },
      "requester" : {
        "user_name" : "jdoe",
        "owner_first_name" : "John",
        "owner_last_name" : "Doe",
        "owner_email" : "jdoe@sample.com",
        "auto_approve" : true
      },
      "tags" : {
        "network_location" : "Internal",
        "cc" : "001"
      },
      "additional_values" : {
        "request_id" : "1001"
      },
      "ems_custom_attributes" : { },
      "miq_custom_attributes" : { }
```

```
  }
}
```

Response:


```
{
  "results": [
    {
      "href" : "http://localhost:3000/api/provision_requests/18",
      "id": "18",
      "description": "Provision from [bd-clone-template] to [aab_rest_vm1]",
      "approval_state": "pending_approval",
      "type": "MiqProvisionRequest",
      "created_on": "2015-05-08T17:42:55Z",
      "updated_on": "2015-05-08T17:42:55Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "template",
      "request_state": "pending",
      "message": "VM Provisioning - Request Created",
      "status": "Ok",
      "options": {
        "use_pre_dialog": false,
        "request_type": "template",
        "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
        "owner_first_name": "John",
        "owner_last_name": "Doe",
        "owner_email": "jdoe@sample.com",
        "vm_tags": [
          62,
          58
        ],
        "addr_mode": [
          "static",
          "Static"
        ],
        "placement_cluster_name": [
          null,
          null
        ],
        "cluster_filter": [
          null,
          null
        ],
        "placement_auto": [
          true,
          1
        ],
        "placement_dc_name": [
          null,
          null
        ],
        "number_of_vms": [
          1,
          "1"
        ],
        "src_vm_id": [
          1947,
          "bd-clone-template"
        ],
        "provision_type": [
          "native_clone",
          "Native Clone"
        ],
        "linked_clone": [
          null,
          null
        ],
        "vm_name": "aab_rest_vm1",
        "pxe_server_id": [
          null,
          null
        ],
        "schedule_type": [
          "immediately",
          "Immediately on Approval"
        ],
        "vm_auto_start": [
          true,
          1
        ],
        "schedule_time": "2015-05-09T13:42:54-04:00",
        "retirement": [
          0,
          "Indefinite"
        ],
        "retirement_warn": [
          604800,
          "1 Week"
        ],
```

```
        "stateless": [
          false,
          0
        ],
        "vlan": [
          "rhevm",
          "rhevm"
        ],
        "disk_format": [
          "default",
          "Default"
        ],
        "number_of_sockets": [
          1,
          "1"
        ],
        "cores_per_socket": [
          1,
          "1"
        ],
        "vm_memory": [
          "1024",
          "1024"
        ],
        "network_adapters": [
          1,
          "1"
        ],
        "placement_host_name": [
          null,
          null
        ],
        "placement_ds_name": [
          null,
          null
        ],
        "src_vm_nics": [

        ],
        "src_vm_lans": [

        ],
        "customize_enabled": [
          "enabled"
        ],
        "src_ems_id": [
          105,
          "rhevm230"
        ],
        "auto_approve": false,
        "ws_values": {
          "request_id": "1001"
        },
        "ws_ems_custom_attributes": {
        },
        "ws_miq_custom_attributes": {
        }
      },
      "userid": "jdoe",
      "source_id": 1947,
      "source_type": "VmOrTemplate"
    }
  ]
}
```


- Trigger multiple Provision Requests
  - Request:
  - Response:

Trigger multiple Provision Requests

In the provision requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

```
POST /api/provision_requests
```

```
{
  "action" : "create",
  "resources" : [
    {
      "version" : "1.1",
      "template_fields" : { "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742" },
      "vm_fields" : {
        "vm_name" : "jdoe_rest_vm1",
```

```
        "number_of_cpus" : 1,
        "vm_memory" : "1024",
        "vlan" : "nic1"
      },
      "requester" : {
        "user_name" : "jdoe",
        "owner_first_name" : "John",
        "owner_last_name" : "Doe",
        "owner_email" : "jdoe@sample.com",
        "auto_approve" : true
      },
      "tags" : {
        "network_location" : "Internal",
        "cc" : "001"
      },
      "additional_values" : { "request_id" : "1001" },
      "ems_custom_attributes" : { },
      "miq_custom_attributes" : { }
    },
    {
      "template_fields" : { "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742" },
      "vm_fields" : {
        "vm_name" : "jdoe_rest_vm2",
        "number_of_cpus" : 1,
        "vm_memory" : "2048",
        "vlan" : "nic1"
      },
      "requester" : {
        "owner_first_name" : "John",
        "owner_last_name" : "Doe",
        "owner_email" : "jdoe@sample.com",
        "auto_approve" : true
      },
      "tags" : {
        "network_location" : "Internal",
        "cc" : "001"
      },
      "additional_values" : { "request_id" : "1002" }
    },
    {
      "template_fields" : { "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742" },
      "vm_fields" : {
        "vm_name" : "jdoe_rest_vm3",
        "number_of_cpus" : 1,
        "vm_memory" : "4096",
        "vlan" : "nic1"
      },
      "requester" : {
        "owner_first_name" : "John",
        "owner_last_name" : "Doe",
        "owner_email" : "jdoe@sample.com",
        "auto_approve" : true
      },
      "tags" : {
        "network_location" : "Internal",
        "cc" : "001"
      },
      "additional_values" : { "request_id" : "1003" }
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/provision_requests/19",
      "id": "19",
      "description": "Provision from [bd-clone-template] to [aab_rest_vm1]",
      "approval_state": "pending_approval",
      "type": "MiqProvisionRequest",
      "created_on": "2015-05-08T18:25:25Z",
      "updated_on": "2015-05-08T18:25:26Z",
      "requester_id": "1",
      "requester_name": "jdoe",
      "request_type": "template",
      "request_state": "pending",
      "message": "VM Provisioning - Request Created",
      "status": "Ok",
      "options": {
        "use_pre_dialog": false,
        "request_type": "template",
        "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
        "owner_first_name": "John",
        "owner_last_name": "Doe",
        "owner_email": "jdoe@sample.com",
        "vm_tags": [
          62,
          58
```

```
      ],
      "addr_mode": [
        "static",
        "Static"
      ],
      "placement_cluster_name": [
        null,
        null
      ],
      "cluster_filter": [
        null,
        null
      ],
      "placement_auto": [
        true,
        1
      ],
      "placement_dc_name": [
        null,
        null
      ],
      "number_of_vms": [
        1,
        "1"
      ],
      "src_vm_id": [
        1947,
        "bd-clone-template"
      ],
      "provision_type": [
        "native_clone",
        "Native Clone"
      ],
      "linked_clone": [
        null,
        null
      ],
      "vm_name": "aab_rest_vm1",
      "pxe_server_id": [
        null,
        null
      ],
      "schedule_type": [
        "immediately",
        "Immediately on Approval"
      ],
      "vm_auto_start": [
        true,
        1
      ],
      "schedule_time": "2015-05-09T14:25:25-04:00",
      "retirement": [
        0,
        "Indefinite"
      ],
      "retirement_warn": [
        604800,
        "1 Week"
      ],
      "stateless": [
        false,
        0
      ],
      "vlan": [
        "rhevm",
        "rhevm"
      ],
      "disk_format": [
        "default",
        "Default"
      ],
      "number_of_sockets": [
        1,
        "1"
      ],
      "cores_per_socket": [
        1,
        "1"
      ],
      "vm_memory": [
        "1024",
        "1024"
      ],
      "network_adapters": [
        1,
        "1"
      ],
      "placement_host_name": [
        null,
        null
      ],
```

```
        "placement_ds_name": [
          null,
          null
        ],
        "src_vm_nics": [

        ],
        "src_vm_lans": [

        ],
        "customize_enabled": [
          "enabled"
        ],
        "src_ems_id": [
          105,
          "rhevm230"
        ],
        "auto_approve": false,
        "ws_values": {
          "request_id": "1001"
        },
        "ws_ems_custom_attributes": {
        },
        "ws_miq_custom_attributes": {
        }
      },
      "userid": "jdoe",
      "source_id": "1947",
      "source_type": "VmOrTemplate"
    },
    {
      "href": "http://localhost:3000/api/provision_requests/20",
      "id": "20",
      "description": "Provision from [bd-clone-template] to [aab_rest_vm2]",
      "approval_state": "pending_approval",
      "type": "MiqProvisionRequest",
      "created_on": "2015-05-08T18:25:28Z",
      "updated_on": "2015-05-08T18:25:29Z",
      "requester_id": "1",
      "requester_name": "jdoe",
      "request_type": "template",
      "request_state": "pending",
      "message": "VM Provisioning - Request Created",
      "status": "Ok",
      "options": {
        "use_pre_dialog": false,
        "request_type": "template",
        "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
        "owner_first_name": "John",
        "owner_last_name": "Doe",
        "owner_email": "jdoe@sample.com",
        "vm_tags": [
          62,
          58
        ],
        "addr_mode": [
          "static",
          "Static"
        ],
        "placement_cluster_name": [
          null,
          null
        ],
        "cluster_filter": [
          null,
          null
        ],
        "placement_auto": [
          true,
          1
        ],
        "placement_dc_name": [
          null,
          null
        ],
        "number_of_vms": [
          1,
          "1"
        ],
        "src_vm_id": [
          1947,
          "bd-clone-template"
        ],
        "provision_type": [
          "native_clone",
          "Native Clone"
        ],
        "linked_clone": [
          null,
          null
        ],
```

```
      "vm_name": "aab_rest_vm2",
      "pxe_server_id": [
        null,
        null
      ],
      "schedule_type": [
        "immediately",
        "Immediately on Approval"
      ],
      "vm_auto_start": [
        true,
        1
      ],
      "schedule_time": "2015-05-09T14:25:28-04:00",
      "retirement": [
        0,
        "Indefinite"
      ],
      "retirement_warn": [
        604800,
        "1 Week"
      ],
      "stateless": [
        false,
        0
      ],
      "vlan": [
        "rhevm",
        "rhevm"
      ],
      "disk_format": [
        "default",
        "Default"
      ],
      "number_of_sockets": [
        1,
        "1"
      ],
      "cores_per_socket": [
        1,
        "1"
      ],
      "vm_memory": [
        "1024",
        "1024"
      ],
      "network_adapters": [
        1,
        "1"
      ],
      "placement_host_name": [
        null,
        null
      ],
      "placement_ds_name": [
        null,
        null
      ],
      "src_vm_nics": [

      ],
      "src_vm_lans": [

      ],
      "customize_enabled": [
        "enabled"
      ],
      "src_ems_id": [
        105,
        "rhevm230"
      ],
      "auto_approve": false,
      "ws_values": {
        "request_id": "1001"
      },
      "ws_ems_custom_attributes": {
      },
      "ws_miq_custom_attributes": {
      }
    },
    "userid": "jdoe",
    "source_id": "1947",
    "source_type": "VmOrTemplate"
  },
  {
    "href": "http://localhost:3000/api/provision_requests/21",
    "id": "21",
    "description": "Provision from [bd-clone-template] to [aab_rest_vm3]",
    "approval_state": "pending_approval",
    "type": "MiqProvisionRequest",
    "created_on": "2015-05-08T18:25:32Z",
```

```
    "updated_on": "2015-05-08T18:25:32Z",
    "requester_id": "1",
    "requester_name": "jdoe",
    "request_type": "template",
    "request_state": "pending",
    "message": "VM Provisioning - Request Created",
    "status": "Ok",
    "options": {
      "use_pre_dialog": false,
      "request_type": "template",
      "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
      "owner_first_name": "John",
      "owner_last_name": "Doe",
      "owner_email": "jdoe@sample.com",
      "vm_tags": [
        62,
        58
      ],
      "addr_mode": [
        "static",
        "Static"
      ],
      "placement_cluster_name": [
        null,
        null
      ],
      "cluster_filter": [
        null,
        null
      ],
      "placement_auto": [
        true,
        1
      ],
      "placement_dc_name": [
        null,
        null
      ],
      "number_of_vms": [
        1,
        "1"
      ],
      "src_vm_id": [
        1947,
        "bd-clone-template"
      ],
      "provision_type": [
        "native_clone",
        "Native Clone"
      ],
      "linked_clone": [
        null,
        null
      ],
      "vm_name": "aab_rest_vm3",
      "pxe_server_id": [
        null,
        null
      ],
      "schedule_type": [
        "immediately",
        "Immediately on Approval"
      ],
      "vm_auto_start": [
        true,
        1
      ],
      "schedule_time": "2015-05-09T14:25:31-04:00",
      "retirement": [
        0,
        "Indefinite"
      ],
      "retirement_warn": [
        604800,
        "1 Week"
      ],
      "stateless": [
        false,
        0
      ],
      "vlan": [
        "rhevm",
        "rhevm"
      ],
      "disk_format": [
        "default",
        "Default"
      ],
      "number_of_sockets": [
        1,
        "1"
```

```
        ],
        "cores_per_socket": [
          1,
          "1"
        ],
        "vm_memory": [
          "1024",
          "1024"
        ],
        "network_adapters": [
          1,
          "1"
        ],
        "placement_host_name": [
          null,
          null
        ],
        "placement_ds_name": [
          null,
          null
        ],
        "src_vm_nics": [

        ],
        "src_vm_lans": [

        ],
        "customize_enabled": [
          "enabled"
        ],
        "src_ems_id": [
          105,
          "rhevm230"
        ],
        "auto_approve": false,
        "ws_values": {
          "request_id": "1001"
        },
        "ws_ems_custom_attributes": {
        },
        "ws_miq_custom_attributes": {
        }
      },
      "userid": "jdoe",
      "source_id": "1947",
      "source_type": "VmOrTemplate"
    }
  ]
}
```

Infrastructure Management API Examples - Service Catalogs

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Add a Service Catalog
- Add multiple Service Catalogs
- Edit a Service Catalog
- Edit Service Catalogs
- Assign Service Templates
- Unassign Service Templates
- Delete Service Catalogs

- Adding a simple Service Catalog
  - Request:
  - Response:

Adding a simple Service Catalog

Request:

```
POST /api/service_catalogs

{
   "action" : "create",
   "resource" : {
        "name" : "Sample Service Catalog",
        "description" : "Description of Sample Service Catalog",
        "service_templates" : [
          { "href" : "http://localhost:3000/api/service_templates/3" },
          { "href" : "http://localhost:3000/api/service_templates/4" }
        ]
   }
}
```

Response:

```
{
  "results": [
    {
      "id": "7",
      "href" : "http://localhost:3000/api/service_catalogs/7",
      "name": "Sample Service Catalog",
      "description": "Description of Sample Service Catalog"
    }
  ]
}
```


- Adding multiple Service Catalogs
  - Request:
  - Response:

Adding multiple Service Catalogs

Request:

```
POST /api/service_catalogs
```

```
{
   "action" : "create",
   "resources" : [
      {
        "name" : "First Sample Service Catalog",
        "description" : "Description of First Sample Service Catalog",
        "service_templates" : [
          { "href" : "http://localhost:3000/api/service_templates/1" },
          { "href" : "http://localhost:3000/api/service_templates/2" }
        ]
      },
      {
        "name" : "Second Sample Service Catalog",
        "description" : "Description of Second Sample Service Catalog",
        "service_templates" : [
          { "href" : "http://localhost:3000/api/service_templates/3" },
          { "href" : "http://localhost:3000/api/service_templates/4" }
        ]
      },
      {
        "name" : "Third Sample Service Catalog",
        "description" : "Description of Third Sample Service Catalog",
        "service_templates" : [
          { "href" : "http://localhost:3000/api/service_templates/5" },
          { "href" : "http://localhost:3000/api/service_templates/6" }
        ]
      }
  ]
}
```

Response:

```
{
  "results": [
    {
      "id": "8",
      "href" : "http://localhost:3000/api/service_catalogs/8",
      "name": "First Sample Service Catalog",
      "description": "Description of First Sample Service Catalog"
    },
    {
      "id": "9",
      "href" : "http://localhost:3000/api/service_catalogs/9",
      "name": "Second Sample Service Catalog",
      "description": "Description of Second Sample Service Catalog"
    },
    {
      "id": "10",
      "href" : "http://localhost:3000/api/service_catalogs/10",
      "name": "Third Sample Service Catalog",
      "description": "Description of Third Sample Service Catalog"
    }
  ]
}
```


- Edit Service Catalog
  - Request:
  - Response:

Edit Service Catalog

Request:

```
POST /api/service_catalog/7
```

```
{
   "action" : "edit",
   "resource" : {
      "description" : "Updated Description of the Seventh Service Catalog"
   }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/service_catalogs/7",
  "id": "7",
  "name": "Sample Service Catalog",
  "description": "Updated Description of the Seventh Service Catalog",
  "service_templates": {
    "count": 1,
    "resources": [
      {
        "href": "http://localhost:3000/api/service_catalogs/7/service_templates/3"
      }
    ]
  }
}
```


- Edit Multiple Service Catalogs
  - Request:
  - Response:

Edit Multiple Service Catalogs

Request:

```
POST /api/service_catalogs
```

```
{
   "action" : "edit",
   "resources" : [
     {
       "href" : "http://localhost:3000/api/service_catalogs/3",
       "description" : "Updated Description for Third Service Catalog"
     },
     {
       "href" : "http://localhost:3000/api/service_catalogs/6",
       "description" : "Updated Description for Sixth Service Catalog"
     }
   ]
}
```

Response:

```
{
  "results": [
    {
      "id" : "3",
      "href" : "http://localhost:3000/api/service_catalogs/3",
      "name": "Service Catalog B Added from REST API",
      "description": "Updated Description for Third Service Catalog"
    },
    {
      "id" : "6",
      "href" : "http://localhost:3000/api/service_catalogs/6",
      "name": "Service Catalog E Added from REST API",
      "description": "Updated Description for Sixth Service Catalog"
    }
  ]
}
```


- Assign service templates to Service Catalog 1
  - Request:
  - Response:

Assign service templates to Service Catalog 1

Request:

```
POST /api/service_catalogs/1/service_templates
```

```
{
  "action" : "assign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/5" },
    { "href" : "http://localhost:3000/api/service_templates/1" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Service Template id:5 name:'template5'",
      "service_template_id": "5",
      "service_template_href": "http://localhost:3000/api/service_templates/5",
      "href": "http://localhost:3000/api/service_catalogs/1"
    },
    {
      "success": false,
      "message": "Service Template 1 is currently assigned to Service Catalog 2",
      "service_template_id": "1",
      "service_template_href": "http://localhost:3000/api/service_templates/1",
      "href": "http://localhost:3000/api/service_catalogs/1"
    }
  ]
}
```


- Unassign service templates from Service Catalog 2
  - Request:
  - Response:

Unassign service templates from Service Catalog 2

Request:

```
POST /api/service_catalogs/6/service_templates
```

```
{
  "action" : "unassign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/1" },
    { "href" : "http://localhost:3000/api/service_templates/5" },
    { "href" : "http://localhost:3000/api/service_templates/8" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": false,
      "message": "Service Template 1 is not currently assigned to Service Catalog 3",
      "service_template_id": "1",
      "service_template_href": "http://localhost:3000/api/service_templates/1",
      "href": "http://localhost:3000/api/service_catalogs/6"
    },
    {
      "success": true,
      "message": "Unassigning Service Template id:5 name:'template5'",
      "service_template_id": "5",
      "service_template_href": "http://localhost:3000/api/service_templates/5",
      "href": "http://localhost:3000/api/service_catalogs/6"
    },
    {
      "success": false,
      "message": "Couldn't find ServiceTemplate with id=8",
      "href": "http://localhost:3000/api/service_catalogs/6"
    }
  ]
}
```


- Delete multiple service catalogs
  - Request:
  - Response:

Delete multiple service catalogs

Request:

```
POST /api/service_catalogs
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_catalogs/8"  },
    { "href" : "http://localhost:3000/api/service_catalogs/9"  },
    { "href" : "http://localhost:3000/api/service_catalogs/10" }
  ]
}
```

Response:

```
{
  "results": [
```

```
    {
      "success": true,
      "message": "service_catalogs id: 8 deleting",
      "href": "http://localhost:3000/api/service_catalogs/8"
    },
    {
      "success": true,
      "message": "service_catalogs id: 9 deleting",
      "href": "http://localhost:3000/api/service_catalogs/9"
    },
    {
      "success": true,
      "message": "service_catalogs id: 10 deleting",
      "href": "http://localhost:3000/api/service_catalogs/10"
    }
  ]
}
```

Infrastructure Management API Examples - Service Templates

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Edit a Service Template
- Edit Service Templates
- Delete Service Templates

- Edit Service Template
  - Request:
  - Response:

Edit Service Template

Request:

```
POST /api/service_templates/2
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "updated_svc_template_02",
    "description" : "This is an updated description for service template 02"
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/service_templates/2",
  "id": "2",
  "name": "updated_svc_template_02",
  "description": "This is an updated description for service template 02",
  "guid": "6f7918b4-d6e7-11e4-9837-b8e85646e742",
  "options": {
  },
  "created_at": "2015-03-30T14:17:02Z",
  "updated_at": "2015-04-16T22:30:02Z",
  "service_type": "unknown",
  "service_template_catalog_id": "6"
}
```


- Edit Service Templates
  - Request:
  - Response:

Edit Service Templates

Request:

```
POST /api/service_templates
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_templates/1",
      "description" : "This is an updated description for the first sample service template"
    },
    {
      "href" : "http://localhost:3000/api/service_templates/2",
      "description" : "This is an updated description for the second sample service template"
    }
  ]
}
```

Response:


```
{
  "results": [
    {
      "id": "1",
      "name": "template1",
      "description": "This is an updated description for the first sample service template",
      "guid": "6a6fdf7e-d6e7-11e4-9837-b8e85646e742",
      "options": {
      },
      "created_at": "2015-03-30T14:16:53Z",
      "updated_at": "2015-04-16T22:33:09Z",
      "service_type": "unknown",
      "service_template_catalog_id": "3"
    },
    {
      "id": "2",
      "name": "updated_svc_template_02",
      "description": "This is an updated description for the second sample service template",
      "guid": "6f7918b4-d6e7-11e4-9837-b8e85646e742",
      "options": {
      },
      "created_at": "2015-03-30T14:17:02Z",
      "updated_at": "2015-04-16T22:33:09Z",
      "service_type": "unknown",
      "service_template_catalog_id": "6"
    }
  ]
}
```


- Delete multiple service templates
  - Request:
  - Response:

Delete multiple service templates

Request:

```
POST /api/service_templates
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/4" },
    { "href" : "http://localhost:3000/api/service_templates/5" }
  ]
}
```

Response:


```
{
  "results": [
    {
      "success": true,
      "message": "service_templates id: 4 deleting",
      "href": "http://localhost:3000/api/service_templates/4"
    },
    {
      "success": true,
      "message": "service_templates id: 5 deleting",
      "href": "http://localhost:3000/api/service_templates/5"
    }
  ]
}
```


Infrastructure Management API Examples - Services

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Order a Service
- Order Multiple Services
- Edit a Service
- Edit a Service with Patch Method
- Edit a Service with PUT
- Edit Services
- Reconfigure a Service
- Refresh Service Template Dialog Fields
- Refresh Service Dialog Fields
- Retire a Service
- Retire Multiple Services
- Trigger a Custom Action on a Service

- Delete Services
- Order a single service from Service Catalog 1
  - Request:
  - Response:

Order a single service from Service Catalog 1

Request:

```
POST /api/service_catalogs/1/service_templates
```

```
{
  "action" : "order",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "option_0_vm_target_name" : "test-vm-0001",
    "option_0_vm_target_hostname" : "test-vm-0001"
  }
}
```

Response:

```
{
  results: [
    {
      "approval_state": "pending_approval",
      "created_on": "2014-07-02T19:28:12Z",
      "description": "Provisioning Service [aws-ubuntu-api] from [aws-ubuntu-api]",
      "destination_id": null,
      "destination_type": null,
      "fulfilled_on": null,
      "id": "13",
      "message": "Service_Template_Provisioning - Request Created",
      "options": {
        "dialog": {
          "dialog_option_0_vm_target_name" : "test-vm-0001",
          "dialog_option_0_vm_target_hostname" : "test-vm-0001"
        }
    }
    "request_state": "pending",
    "request_type": "clone_to_service",
    "requester_id": "1",
    "requester_name": "Administrator",
    "source_id": "6",
    "source_type": "ServiceTemplate",
    "status": "Ok",
    "updated_on": "2014-07-02T19:28:12Z",
    "userid": "admin"
    }
  ]
}
```


- Order multiple services from Service Catalog 2
  - Request:
  - Response:

Order multiple services from Service Catalog 2

Request:

```
POST /api/service_catalogs/2/service_templates
```

```
{
  "action" : "order",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_templates/3",
      "option_1_vm_target_name" : "sample-vm-1201",
      "option_2_vm_target_hostname" : "sample-vm-1201"
    },
    {
      "href" : "http://localhost:3000/api/service_templates/3",
      "option_1_vm_target_name" : "sample-vm-1202",
      "option_2_vm_target_hostname" : "sample-vm-1202"
    },
    {
      "href" : "http://localhost:3000/api/service_templates/4",
      "option_1_vm_target_name" : "dev-vm1",
      "option_2_vm_target_hostname" : "dev-vm1",
      "option_3_vm_memory" : '16384'
    },
  ]
}
```

Response:

```
{
  results: [
    {
      "approval_state": "pending_approval",
      "created_on": "2014-07-02T19:28:12Z",
      "description": "Provisioning Service [sample-vm-1201] from [sample-vm-1201]",
      "destination_id": null,
      "destination_type": null,
      "fulfilled_on": null,
      "id": "13",
      "message": "Service_Template_Provisioning - Request Created",
      "options": {
        "dialog": {
          "dialog_option_0_vm_target_name" : "test-vm-0001",
          "dialog_option_0_vm_target_hostname" : "test-vm-0001"
        }
    }
    "request_state": "pending",
    "request_type": "clone_to_service",
    "requester_id": "1",
    "requester_name": "Administrator",
    "source_id": "6",
    "source_type": "ServiceTemplate",
    "status": "Ok",
    "updated_on": "2014-07-02T19:28:12Z",
    "userid": "admin"
    },
    {
      "approval_state": "pending_approval",
      "created_on": "2014-07-02T19:28:12Z",
      "description": "Provisioning Service [sample-vm-1202] from [sample-vm-1202]",
      "destination_id": null,
      "destination_type": null,
      "fulfilled_on": null,
      "id": "13",
      "message": "Service_Template_Provisioning - Request Created",
      "options": {
        "dialog": {
          "dialog_option_0_vm_target_name" : "test-vm-0001",
          "dialog_option_0_vm_target_hostname" : "test-vm-0001"
        }
    }
    "request_state": "pending",
    "request_type": "clone_to_service",
    "requester_id": "1",
    "requester_name": "Administrator",
    "source_id": "6",
    "source_type": "ServiceTemplate",
    "status": "Ok",
    "updated_on": "2014-07-02T19:28:12Z",
    "userid": "admin"
    },
    {
      "approval_state": "pending_approval",
      "created_on": "2014-07-02T19:28:12Z",
      "description": "Provisioning Service [sample-vm-1201] from [sample-vm-1201]",
      "destination_id": null,
      "destination_type": null,
      "fulfilled_on": null,
      "id": "13",
      "message": "Service_Template_Provisioning - Request Created",
      "options": {
        "dialog": {
          "dialog_option_0_vm_target_name" : "test-vm-0001",
          "dialog_option_0_vm_target_hostname" : "test-vm-0001"
        }
    }
    "request_state": "pending",
    "request_type": "clone_to_service",
    "requester_id": "1",
    "requester_name": "Administrator",
    "source_id": "6",
    "source_type": "ServiceTemplate",
    "status": "Ok",
    "updated_on": "2014-07-02T19:28:12Z",
    "userid": "admin"
    }
  ]
}
```


- Edit Service
  - Request:
  - Response:

Edit Service

Request:

```
POST /api/services/101
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "service_101",
    "description" : "This is an updated description for service 101"
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/services/101",
  "id": "101",
  "name": "service_101",
  "description": "This is an updated description for the service 101",
  "guid": "4a8f96de-a1a6-11e4-9f8d-b8e85646e742",
  "options": {
  },
  "created_at": "2015-01-21T19:47:11Z",
  "updated_at": "2015-04-16T22:17:15Z"
}
```


- Editing a resource via the Patch Method.
  - Request:
  - Response:

Editing a resource via the Patch Method.

Supported attribute actions for PATCH include, add, edit and remove.

Request:

```
PATCH /api/services/90
```

```
[
  { "action" : "edit", "path" : "name", "value" : "updated_service_90" },
  { "action" : "remove", "path" : "description"}
]
```

Response:

```
{
  "href": "http://localhost:3000/api/services/90",
  "id": "90",
  "name": "updated_service_90",
  "guid": "eb6fc61c-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-04-16T22:38:57Z"
}
```

Note:

Note that the description attribute is no longer defined for this service.

- Edit Service via PUT
  - Request:
  - Response:

Edit Service via PUT

Request:

```
PUT /api/services/90
```

```
{
  "name" : "new_service_90",
  "description" : "This is a new description for the service 90"
}
```

Response:

```
{
  "href": "http://localhost:3000/api/services/90",
  "id": "90",
  "name": "new_service_90",
  "description": "This is a new description for the service 90",
  "guid": "eb6fc61c-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-04-16T22:41:37Z"
}
```


- Edit Services

    - Request:
    - Response:

Edit Services

Request:

```
POST /api/services
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/81",
      "description" : "This is an updated description for service 81"
    },
    {
      "href" : "http://localhost:3000/api/services/82",
      "description" : "This is an updated description for service 82"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/services/81",
      "id": "81",
      "name": "api_gen_A_81",
      "description": "This is an updated description for service 81",
      "guid": "eb6daaf8-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-04-16T22:43:37Z"
    },
    {
      "href": "http://localhost:3000/api/services/82",
      "id": "82",
      "name": "api_gen_A_82",
      "description": "This is an updated description for service 82",
      "guid": "eb6de400-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-04-16T22:43:37Z"
    }
  ]
}
```


- Reconfigure Service
    - Request:
    - Response:

Reconfigure Service

Request:

```
POST /api/services/2
```

```
{
  "action" : "reconfigure",
  "resource" : {
     "SampleText" : "default_12",
     "SampleHostIp" : "100.200.300.12"
  }
}
```

Response:

```
{
  "success": true,
  "message": "Service id:2 name:'SampleStaticServiceCatalogItem' reconfiguring",
  "result": {
    "id": "74",
    "description": "Service Reconfigure for: SampleStaticServiceCatalogItem",
    "approval_state": "pending_approval",
    "type": "ServiceReconfigureRequest",
    "created_on": "2016-06-10T20:34:13Z",
    "updated_on": "2016-06-10T20:34:13Z",
    "requester_id": "1",
    "requester_name": "Administrator",
    "request_type": "service_reconfigure",
    "request_state": "pending",
    "message": "Service Reconfigure - Request Created",
```

```
    "status": "Ok",
    "options": {
      "dialog": {
        "dialog_SampleText": "default_12",
        "dialog_SampleHostName": "",
        "dialog_SampleHostIp": "100.200.300.12",
        "dialog_sample_button_one": "complex_stuff",
        "dialog_vm_host_name": ""
      },
      "workflow_settings": {
        "resource_action_id": "23",
        "dialog_id": "3"
      },
      "src_id": "2",
      "requester_group": "EvmGroup-super_administrator"
    },
    "userid": "admin",
    "source_id": "2",
    "source_type": "Service",
    "tenant_id": "1",
    "service_order_id": "9",
    "process": true
  },
  "href": "http://localhost:3000/api/services/2"
}
```


- Refresh Dialog Fields
  - Request:
  - Response:

Refresh Dialog Fields

Refresh fields of a service template dialog:

Request:

```
POST /api/service_catalogs/1/service_templates
```


```
{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/1",
    "dialog_fields" : {
      "my_field" : "current value for my_field"
    },
    "fields" : [ "my_field" ]
  }
}
```


Response:


```
{
  "results": [
    {
      "success": true,
      "message": "Refreshing dialog fields for Service Template id:1 name:'DynamicDialog'",
      "result": {
        "my_field": {
          "name": "my_field",
          "description": "my_field",
          "type": "DialogFieldDropDownList",
          "data_type": "string",
          "display": "edit",
          "display_method_options": {
          },
          "required": true,
          "required_method_options": {
          },
          "values": [
            [
              "",
              "Choose"
            ],
            [
              "6",
              "6"
            ],
            [
              "15",
              "15"
            ],
            [
              "2016-06-10T15:14:30Z",
              "2016-06-10T15:14:30Z"
            ]
          ],
          "values_method_options": {
          },
          "options": {
```

```
            "sort_by": "value",
            "sort_order": "ascending"
          },
          "label": "My Field",
          "position": 0,
          "dynamic": true,
          "show_refresh_button": true,
          "load_values_on_init": true,
          "read_only": false,
          "resource_action": {
            "resource_type": "DialogField",
            "ae_namespace": "Dialogs",
            "ae_class": "dynamic_dialogs",
            "ae_instance": "drop_down",
            "ae_attributes": {
            }
          }
        }
      },
      "service_template_id": "1",
      "service_template_href": "http://localhost:3000/api/service_templates/1",
      "href": "http://localhost:3000/api/service_catalogs/1"
    }
  ]
}
```


- Refresh Dialog Fields
  - Request:
  - Response:

Refresh Dialog Fields

Refresh fields of a service dialog:

Request:

```
POST /api/service_dialogs/2
```

```
{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "dialog_fields" : {
      "my_field" : "current value for my_field"
    },
    "fields" : [ "my_field" ]
  }
}
```

Response:


```
{
  "success": true,
  "message": "Refreshing dialog fields for Service Dialog id:2 label:'Dynamic Drop-down test'",
  "result": {
    "my_field": {
      "name": "my_field",
      "description": "my_field",
      "type": "DialogFieldDropDownList",
      "data_type": "string",
      "display": "edit",
      "display_method_options": {
      },
      "required": true,
      "required_method_options": {
      },
      "values": [
        [
          "",
          "Choose"
        ],
        [
          "7",
          "7"
        ],
        [
          "13",
          "13"
        ],
        [
          "2016-06-10T17:59:36Z",
          "2016-06-10T17:59:36Z"
        ]
      ],
      "values_method_options": {
      },
      "options": {
        "sort_by": "value",
        "sort_order": "ascending"
      },
```

```
      "label": "My Field",
      "position": 0,
      "dynamic": true,
      "show_refresh_button": true,
      "load_values_on_init": true,
      "read_only": false,
      "resource_action": {
        "resource_type": "DialogField",
        "ae_namespace": "Dialogs",
        "ae_class": "dynamic_dialogs",
        "ae_instance": "drop_down",
        "ae_attributes": {
        }
      }
    }
  },
  "href": "http://localhost:3000/api/service_dialogs/2"
}
```


- Retire Service Immediately
  - Request:
  - Response:
  - Response:
- Retire Service In the Future
  - Request:
  - Response:

Retire Service Immediately

Request:

```
POST /api/services
```

```
{
  "action" : "request_retire",
  "resource" : { "href" : "http://localhost:3000/api/services/35" }
}
```

Response:

```
{
  "results" : [
    {
      "href": "http://localhost:3000/api/services/95",
      "id": "95",
      "name": "sample_service_95",
      "description": "Description for sample_service_95",
      "guid": "eb713538-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-05-08T19:32:00Z"
    }
  ]
}
```

Or simply targeting the service directly:

```
POST /api/services/95
```

```
{
  "action" : "request_retire"
}
```

Response:

```
{
  "href": "http://localhost:3000/api/services/95",
  "id": "95",
  "name": "sample_service_95",
  "description": "Description for sample_service_95",
  "guid": "eb713538-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-05-08T19:32:00Z"
}
```

Retire Service In the Future

Request:

```
POST /api/services/93
```

```
{
  "action" : "request_retire",
```

```
  "resource" : { "date" : "10/31/2015", "warn" : "7"  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/services/93",
  "id": "93",
  "name": "sample_service_93",
  "description": "Description for sample_service_93",
  "guid": "eb709e02-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-05-08T19:40:19Z",
  "retired": false,
  "retires_on": "2015-10-30",
  "retirement_warn": 7
}
```


- Retire Multiple Services
  - Request:
  - Response:

Retire Multiple Services

Request:

```
POST /api/services

{
  "action" : "request_retire",
  "resources" : [
      { "href" : "http://localhost:3000/api/services/100" },
      { "href" : "http://localhost:3000/api/services/101", "date" : "11/01/2015", "warn" : "4" },
      { "href" : "http://localhost:3000/api/services/102", "date" : "11/02/2015", "warn" : "4" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/services/100",
      "id": "100",
      "name": "cloud_service_100",
      "description": "Description for cloud_service_100",
      "guid": "eb725f94-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-05-08T19:46:11Z"
    },
    {
      "href": "http://localhost:3000/api/services/101",
      "id": "101",
      "name": "cloud_service_101",
      "description": "Description for cloud_service_101",
      "guid": "4a8f96de-a1a6-11e4-9f8d-b8e85646e742",
      "options": {
      },
      "created_at": "2015-01-21T19:47:11Z",
      "updated_at": "2015-05-08T19:46:19Z",
      "retired": false,
      "retires_on": "2015-11-01",
      "retirement_warn": 4
    },
    {
      "href": "http://localhost:3000/api/services/102",
      "id": "102",
      "name": "cloud_service_102",
      "description": "Description for cloud_service_102",
      "guid": "e2da0cb0-e47e-11e4-a47f-b8e85646e742",
      "options": {
      },
      "created_at": "2015-04-16T21:23:55Z",
      "updated_at": "2015-05-08T19:46:19Z",
      "retired": false,
      "retires_on": "2015-11-02",
      "retirement_warn": 3
    }
  ]
}
```


- Trigger a Custom Action on a Service
  - Request:
  - Response:

- Request:
- Response:

Trigger a Custom Action on a Service

This page shows an ordered service that has a custom button with a dialog and how the custom action is triggered.

Here is the ordered service with the expanded custom actions:

Request:

```
GET /api/services/110?attributes=custom_actions
```

Response:


```
{
  "href": "http://localhost:3000/api/services/110",
  "id": "110",
  "name": "SampleServiceTemplate",
  "description": "Sample Service Template",
  "guid": "a3ab175c-8ca1-11e5-939f-000a27020067",
  "service_template_id": "9",
  "options": {
    "button_order": [
      "cb-10"
    ],
    "dialog": {
      "dialog_hostname": "myhostname",
      "dialog_ipaddress": "192.168.100.1",
      "dialog_environmenttype": "medium"
    }
  },
  "display": true,
  "created_at": "2015-11-16T20:35:56Z",
  "updated_at": "2015-11-16T20:35:56Z",
  "evm_owner_id": "1",
  "miq_group_id": "1",
  "tenant_id": "1",
  "custom_actions": {
    "buttons": [
      {
        "id": "10",
        "guid": "cdcec41c-8ca0-11e5-8c79-000a27020067",
        "description": "Query VM",
        "applies_to_class": "ServiceTemplate",
        "options": {
          "button_image": 4,
          "display": true
        },
        "userid": "admin",
        "created_on": "2015-11-16T20:29:58Z",
        "updated_on": "2015-11-16T20:29:58Z",
        "name": "QueryVM",
        "visibility": {
          "roles": [
            "_ALL_"
          ]
        },
        "applies_to_id": "9",
        "resource_action": {
          "id": "38",
          "dialog_id": "4",
          "resource_id": "10",
          "resource_type": "CustomButton",
          "created_at": "2015-11-16T20:29:58Z",
          "updated_at": "2015-11-16T20:29:58Z",
          "ae_namespace": "SYSTEM",
          "ae_class": "PROCESS",
          "ae_instance": "Request",
          "ae_attributes": {
            "attr1": "value1",
            "attr2": "value2",
            "request": "System/Project/InspectME"
          },
          "dialog": {
            "description": "Sample Dialog Description",
            "buttons": "submit",
            "label": "SampleDialog",
            "dialog_tabs": [
              {
                "description": "VM Tab Description",
                "display": "edit",
                "label": "VMTabLabel",
                "position": 0,
                "dialog_groups": [
                  {
                    "description": "Vm Info Description",
                    "display": "edit",
                    "label": "VmInfoLabel",
```

```
                "position": 0,
                "dialog_fields": [
                  {
                    "name": "hostname",
                    "description": "Hostname of the VM",
                    "type": "DialogFieldTextBox",
                    "display": "edit",
                    "display_method_options": {
                    },
                    "required": false,
                    "required_method_options": {
                    },
                    "default_value": "default_hostname",
                    "values_method_options": {
                    },
                    "options": {
                      "protected": false
                    },
                    "label": "HostNameLabel",
                    "position": 0,
                    "dynamic": false,
                    "read_only": false,
                    "resource_action": {
                      "resource_type": "DialogField",
                      "ae_attributes": {
                      }
                    }
                  },
                  {
                    "name": "ipaddress",
                    "description": "IP Address of VM",
                    "type": "DialogFieldTextBox",
                    "display": "edit",
                    "display_method_options": {
                    },
                    "required": false,
                    "required_method_options": {
                    },
                    "default_value": "",
                    "values_method_options": {
                    },
                    "options": {
                      "protected": false
                    },
                    "label": "IpaddressLabel",
                    "position": 1,
                    "dynamic": false,
                    "read_only": false,
                    "resource_action": {
                      "resource_type": "DialogField",
                      "ae_attributes": {
                      }
                    }
                  }
                ]
              }
            ]
          },
          {
            "description": "Environment Tab Description",
            "display": "edit",
            "label": "EnvironmentTab",
            "position": 1,
            "dialog_groups": [
              {
                "description": "Environment Box Description",
                "display": "edit",
                "label": "EnvironmentBoxLabel",
                "position": 0,
                "dialog_fields": [
                  {
                    "name": "environmenttype",
                    "description": "Environment Type Description",
                    "type": "DialogFieldRadioButton",
                    "data_type": "string",
                    "display": "edit",
                    "display_method_options": {
                    },
                    "required": true,
                    "required_method_options": {
                    },
                    "values": [
                      [
                        "large",
                        "Large Environment"
                      ],
                      [
                        "medium",
                        "Medium Environment"
                      ],
                      [
```

```
                          "small",
                          "Small Environment"
                        ]
                      ],
                      "values_method_options": {
                      },
                      "options": {
                        "sort_by": "description",
                        "sort_order": "ascending"
                      },
                      "label": "EnvTypeLabel",
                      "position": 0,
                      "dynamic": false,
                      "read_only": false,
                      "resource_action": {
                        "resource_type": "DialogField",
                        "ae_attributes": {
                        }
                      }
                    }
                  ]
                }
              ]
            }
          ]
        }
      }
    }
  ],
  "button_groups": [
```

```
    ]
  },
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/services/110"
    },
    {
      "name": "request_retire",
      "method": "post",
      "href": "http://localhost:3000/api/services/110"
    },
    {
      "name": "set_ownership",
      "method": "post",
      "href": "http://localhost:3000/api/services/110"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/services/110"
    },
    {
      "name": "queryvm",
      "method": "post",
      "href": "http://localhost:3000/api/services/110"
    }
  ]
}
```

Showing the **queryvm** custom action being returned in the actions list.

The following is the request triggering a **queryvm** action on the service:

Request:

```
POST /api/services/110

{
  "action" : "queryvm",
  "resource" : {
    "hostname": "updated_hostname",
    "ipaddress": "192.168.100.2",
    "environmenttype" : "medium"
  }
}
```

Response:

```
{
  "success": true,
  "message": "Invoked custom dialog action queryvm for services id: 110",
  "href": "http://localhost:3000/api/services/110"
}
```


- Delete Services
  - Request:

- Response:

Delete Services

Request:

```
POST /api/services
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/97" },
    { "href" : "http://localhost:3000/api/services/98" },
    { "href" : "http://localhost:3000/api/services/99" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "services id: 97 deleting",
      "href": "http://localhost:3000/api/services/97"
    },
    {
      "success": true,
      "message": "services id: 98 deleting",
      "href": "http://localhost:3000/api/services/98"
    },
    {
      "success": true,
      "message": "services id: 99 deleting",
      "href": "http://localhost:3000/api/services/99"
    }
  ]
}
```

Infrastructure Management API Examples - VM Management

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Start a VM
- Stop a VM
- Suspend a VM
- Scan a VM
- Set owner of a VM
- Add Event to a VM
- Add Lifecycle Event to VM
- Delete a VM

- Start a VM
  - Request:
  - Response:
- Querying task progress:
  - Request:
  - Response:

Start a VM

Request:

```
POST /api/vms/1921
```

```
{
  "action": "start"
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' starting",
  "task_id": "610",
  "task_href": "http://localhost:3000/api/tasks/610",
  "href": "http://localhost:3000/api/vms/1921"
}
```

Querying task progress:

Request:

```
GET /api/tasks/620
```

Response:

```
{
  "href": "http://localhost:3000/api/tasks/610",
  "id": "610",
  "name": "VM id:1921 name:'aab_demo_vm' starting",
  "state": "Queued",
  "status": "Ok",
  "message": "Queued the action: [VM id:1921 name:'aab_demo_vm' starting] being run for user: [admin]",
  "userid": "admin",
  "created_on": "2015-05-05T15:58:08Z",
  "updated_on": "2015-05-05T15:58:08Z"
}
```


- Stop a VM
  - Request:
  - Response:

Stop a VM

Request:

```
POST /api/vms/1921
```

```
{
  "action": "stop"
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' stopping",
  "task_id": "619",
  "task_href": "http://localhost:3000/api/tasks/619",
  "href": "http://localhost:3000/api/vms/1921"
}
```


- Suspend a VM
  - Request:
  - Response:

Suspend a VM

Request:

```
POST /api/vms/1921
```

```
{
  "action": "suspend"
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' suspending",
  "task_id": "620",
  "task_href": "http://localhost:3000/api/tasks/620",
  "href": "http://localhost:3000/api/vms/1921"
}
```


- Scan a VM
  - Request:
  - Response:
  - Response:

Scan a VM

Request:

```
POST /api/vms/1922
```

```
{
  "action": "scan"
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1922 name:'aab_test_vm' scanning",
  "task_id": "618",
  "task_href": "http://localhost:3000/api/tasks/618",
  "href": "http://localhost:3000/api/vms/1922"
}
```

And then optionally waiting for the task to complete as follows:

```
GET /api/tasks/618
```

Response:

```
{
  "href": "http://localhost:3000/api/tasks/618",
  "id": "618",
  "name": "VM id:1922 name:'aab_test_vm' scanning",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2015-05-05T19:37:32Z",
  "updated_on": "2015-05-05T19:37:38Z"
}
```


- Set owner of a VM
  - Request:
  - Response:

Set owner of a VM

Request:

```
POST /api/vms/1921
```

```
{
  "action": "set_owner",
  "resource" : {
    "owner" : "admin"
  }
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' setting owner to 'admin'",
  "href": "http://localhost:3000/api/vms/1921"
}
```


- Add event to a VM
  - Request:
  - Response:

Add event to a VM

Request:

```
POST /api/vms/1921
```

```
{
  "action": "add_event",
  "resource" : {
    "event_type" : "BadUserNameSessionEvent",
    "event_message" : "Cannot login user@test.domain"
  }
}
```

Response:

```
{
  "success": true,
  "message": "Adding Event type=BadUserNameSessionEvent message=Cannot login user@test.domain",
  "href": "http://localhost:3000/api/vms/1921"
}
```


- Delete VMs
  - Deleting a single VM
    - Request:
  - Deleting multiple VMs
    - Request:
    - Response:
    - Request:

- Response:

Delete VMs

Deleting a single VM

Request:

```
DELETE /api/vms/334
```

Upon success, Vm targeted for deletion asynchronously and No Response Content with HTTP status code 204

Deleting multiple VMs

Request:

```
DELETE /api/vms
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/348" },
    { "href" : "http://localhost:3000/api/vms/349" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "VM id:348 name:'aab-temp1' deleting",
      "task_id": "616",
      "task_href": "http://localhost:3000/api/tasks/616",
      "href": "http://localhost:3000/api/vms/348"
    },
    {
      "success": true,
      "message": "VM id:349 name:'aab-temp2' deleting",
      "task_id": "617",
      "task_href": "http://localhost:3000/api/tasks/617",
      "href": "http://localhost:3000/api/vms/349"
    }
  ]
}
```

Optionally monitoring the asynchronous Vm deletion by accessing the related Task as follows:

Request:

```
GET /api/tasks/616
```

Response:

```
{
  "href": "http://localhost:3000/api/tasks/616",
  "id": "616",
  "name": "VM id:348 name:'aab-temp1' deleting",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2015-05-05T19:33:35Z",
  "updated_on": "2015-05-05T19:33:40Z"
}
```

Infrastructure Management API Examples - Tag Management

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Assign Tags
- Assign Tags to Service Template
- Unassign Tags
- Unassign Tags from Service Template

- Assign Tags to a Service
  - Request:
  - Response:
- Assign Tags by Name to a Service
  - Request:

    - Response:
- Assign a Tag by Reference to a Service
    - Request:
    - Response:

Assign Tags to a Service

Request:

```
POST /api/services/101/tags

{
  "action" : "assign",
  "resources" : [
    { "category" : "location", "name" : "ny" },
    { "category" : "department", "name" : "finance" },
    { "category" : "environment", "name" : "dev" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "location",
      "tag_name": "ny"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'environment' name:'dev'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "environment",
      "tag_name": "dev"
    }
  ]
}
```

Assign Tags by Name to a Service

Request:

```
POST /api/services/101/tags

{
  "action" : "assign",
  "resources" : [
    { "name" : "/department/finance" },
    { "name" : "/location/ny" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "location",
      "tag_name": "ny"
    }
  ]
}
```

Assign a Tag by Reference to a Service

Request:

```
POST /api/services/101/tags
```

```
{
  "action" : "assign",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/1/tags/49" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/49"
    }
  ]
}
```


- Assign Tags to service template 1
  - Request:
  - Response:

Assign Tags to service template 1

Request:

```
POST /api/service_templates/1/tags
```

```
{
  "action" : "assign",
  "resources" : [
    { "category" : "location", "name" : "ny" },
    { "category" : "department", "name" : "finance" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "location",
      "tag_name": "ny"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "department",
      "tag_name": "finance"
    }
  ]
}
```


- Unassign Tags from a Service
  - Request:
  - Response:
- Unassign a Tag by Name from a Service
  - Request:
  - Response:
- Unassign a Tag by Reference from a Service
  - Request:
  - Response:

Unassign Tags from a Service

Request:

```
POST /api/services/101/tags
```

```
{
  "action" : "unassign",
  "resources" : [
    { "category" : "department", "name" : "finance" },
    { "category" : "environment", "name" : "dev" }
```

```
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'environment' name:'dev'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "environment",
      "tag_name": "dev"
    }
  ]
}
```

Unassign a Tag by Name from a Service

Request:

```
POST /api/services/101/tags
```

```
{
  "action" : "unassign",
  "resources" : [
    { "name" : "/managed/department/finance" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    }
  ]
}
```

Unassign a Tag by Reference from a Service

Request:

```
POST /api/services/101/tags
```

```
{
  "action" : "unassign",
  "resources" : [
    { "href" : "tags/49" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/49"
    }
  ]
}
```


- Unassign Tags from service template 1
  - Request:
  - Response:

Unassign Tags from service template 1

Request:

```
POST /api/service_templates/1/tags

{
  "action" : "unassign",
  "resources" : [
    { "category" : "location", "name" : "ny" },
    { "category" : "department", "name" : "finance" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "location",
      "tag_name": "ny"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "department",
      "tag_name": "finance"
    }
  ]
}
```

Infrastructure Management API Examples - Usage Flows

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- Compliance Policy Flow
- Compliance Policy Flow
  - Create a couple of conditions for the new compliance policy
    - Request:
    - Response:
  - Find the Policy action to use
    - Request:
    - Response:
  - Find the Policy event to use
    - Response:
  - Create the Compliance Policy for Finance Servers
    - Request:
    - Response:
  - Create the Compliance Policy Profile for Finance Servers
    - Request:
    - Response:
  - Assigning new policy to policy profile
    - Request:
    - Response:
  - Assigning the policy profile to a virtual machine
    - Request:
    - Response:
  - Querying the policy profiles and related policies of the virtual machine
    - Request:
    - Response:
  - Requesting a compliance check on the virtual machine
    - Request:
    - Response:
  - Waiting for the compliance check to complete
    - Response:
  - Querying all compliances run for the virtual machine
    - Request:
    - Response:
  - Querying the virtual machine and its latest compliance result including the last compliance status
    - Request:
    - Response:

Compliance Policy Flow

Create a couple of conditions for the new compliance policy

Request:

```
POST /api/conditions
```

```
{
  "action" : "create",
  "resources" : [
    {
      "name" : "finance_mem_req",
      "description" : "System Memory Requirement for Finance Systems",
      "expression" : {
        ">=": {
          "field": "Vm-mem_cpu",
          "value": "4096"
        }
      },
      "towhat" : "Vm"
    },
    {
      "name" : "finance_cpu_req",
      "description" : "System CPU Requirement for Finance Systems",
      "expression" : {
        ">=": {
          "field": "Vm-num_cpu",
          "value": "2"
        }
      },
      "towhat" : "Vm"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/conditions/8",
      "id": "8",
      "name": "finance_mem_req",
      "description": "System Memory Requirement for Finance Systems",
      "expression": {
        "exp": {
          ">=": {
            "field": "Vm-mem_cpu",
            "value": "4096"
          }
        },
        "context_type": null
      },
      "towhat": "Vm",
      "file_mtime": null,
      "created_on": "2020-03-24T17:19:21Z",
      "updated_on": "2020-03-24T17:19:21Z",
      "guid": "b67a564d-5290-4245-b4bc-4d75d0890951",
      "filename": null,
      "applies_to_exp": null,
      "miq_policy_id": null,
      "notes": null,
      "read_only": null
    },
    {
      "href": "http://localhost:3000/api/conditions/9",
      "id": "9",
      "name": "finance_cpu_req",
      "description": "System CPU Requirement for Finance Systems",
      "expression": {
        "exp": {
          ">=": {
            "field": "Vm-num_cpu",
            "value": "2"
          }
        },
        "context_type": null
      },
      "towhat": "Vm",
      "file_mtime": null,
      "created_on": "2020-03-24T17:19:21Z",
      "updated_on": "2020-03-24T17:19:21Z",
      "guid": "2c3c6fd1-c6a4-44e7-8a86-b1f44c1e2572",
      "filename": null,
      "applies_to_exp": null,
      "miq_policy_id": null,
      "notes": null,
```

```
      "read_only": null
    }
  ]
}
```

Find the Policy action to use

Request:

```
GET /api/policy_actions?expand=resources&filter[]=name=compliance_failed
```

Response:

```
{
  "name": "policy_actions",
  "count": 42,
  "subcount": 1,
  "subquery_count": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/policy_actions/34",
      "id": "34",
      "name": "compliance_failed",
      "description": "Mark as Non-Compliant",
      "created_on": "2019-08-08T19:40:10Z",
      "updated_on": "2019-08-08T19:40:10Z",
      "guid": "47969a3f-4eb8-4e7b-b7d2-542e4c0cb0d5",
      "action_type": "default",
      "options": {
      }
    }
  ],
  "actions": [
    {
      "name": "query",
      "method": "post",
      "href": "http://localhost:3000/api/policy_actions"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/policy_actions?
attributes=name%2Cname&expand=resources&filter%5B%5D=name%3Dcompliance_failed&offset=0",
    "first": "http://localhost:3000/api/policy_actions?
attributes=name%2Cname&expand=resources&filter%5B%5D=name%3Dcompliance_failed&offset=0",
    "last": "http://localhost:3000/api/policy_actions?
attributes=name%2Cname&expand=resources&filter%5B%5D=name%3Dcompliance_failed&offset=0"
  }
}
```

Find the Policy event to use

```
GET /api/events?expand=resources&filter[]=name=vm_compliance_failed
```

Response:

```
{
  "name": "events",
  "count": 191,
  "subcount": 1,
  "subquery_count": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/events/119",
      "id": "119",
      "name": "vm_compliance_check",
      "description": "VM Compliance Check",
      "created_on": "2019-08-08T19:40:12Z",
      "updated_on": "2019-08-08T19:40:12Z",
      "guid": "f95fd930-b70d-4198-b018-7f2fe37a33f9",
      "event_type": "Default",
      "definition": null,
      "default": null,
      "enabled": null
    }
  ],
  "actions": [
    {
      "name": "query",
      "method": "post",
      "href": "http://localhost:3000/api/events"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/events?expand=resources&filter%5B%5D=name%3Dvm_compliance_check&offset=0",
    "first": "http://localhost:3000/api/events?expand=resources&filter%5B%5D=name%3Dvm_compliance_check&offset=0",
    "last": "http://localhost:3000/api/events?expand=resources&filter%5B%5D=name%3Dvm_compliance_check&offset=0"
```

```
  }
}
```

Create the Compliance Policy for Finance Servers

Request:

```
POST /api/policies
```

```
{
  "name" : "finance_server_policy",
  "description" : "Finance Server Policy",
  "mode" : "compliance",
  "towhat" : "Vm",
  "conditions_ids" : [ "8", "9" ],
  "policy_contents" : [
    {
      "event_id" : "119",
      "actions" : [
        {
          "action_id" : "34",
          "opts" : { "qualifier" : "failure" }
        }
      ]
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/policies/11",
      "id": "11",
      "name": "finance_server_policy",
      "description": "Finance Server Policy",
      "created_on": "2020-03-25T16:25:06Z",
      "updated_on": "2020-03-25T16:25:06Z",
      "expression": null,
      "towhat": "Vm",
      "guid": "95f28050-7947-41e4-8082-ee9cd512b5fd",
      "created_by": null,
      "updated_by": null,
      "notes": null,
      "active": true,
      "mode": "compliance",
      "read_only": null
    }
  ]
}
```

Create the Compliance Policy Profile for Finance Servers

Request:

```
POST /api/policy_profiles
```

```
{
  "name" : "finance_policy_profile",
  "description" : "Finance Servers Policy Profile",
  "mode" : "compliance"
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/policy_profiles/35",
      "id": "35",
      "name": "finance_policy_profile",
      "description": "Finance Servers Policy Profile",
      "set_type": "MiqPolicySet",
      "created_on": "2020-03-24T18:56:23Z",
      "updated_on": "2020-03-24T18:56:23Z",
      "guid": "d4546f30-15a7-48f3-a793-cfd6d1d849a2",
      "read_only": null,
      "set_data": null,
      "mode": "compliance",
      "owner_type": null,
      "owner_id": null,
      "userid": null,
      "group_id": null
    }
  ]
}
```

Assigning new policy to policy profile

Request:

```
POST /api/policy_profiles/35/policies

{
  "action" : "assign",
  "resources" : [
    { "id" : "11" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Policy: id:'11' description:'Finance Server Policy' guid:'95f28050-7947-41e4-8082-
ee9cd512b5fd'",
      "href": "http://localhost:3000/api/policy_profiles/35",
      "policy_id": "12",
      "policy_href": "http://localhost:3000/api/policies/11"
    }
  ]
}
```

Assigning the policy profile to a virtual machine

Request:

```
POST /api/vms/171/policy_profiles

{
  "action" : "assign",
  "resources" : [
    { "id" : "35" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Policy Profile: id:'35' description:'Finance Servers Policy Profile' guid:'d4546f30-15a7-48f3-
a793-cfd6d1d849a2'",
      "href": "http://localhost:3000/api/vms/171",
      "policy_profile_id": "35",
      "policy_profile_href": "http://localhost:3000/api/policy_profiles/35"
    }
  ]
}
```

Querying the policy profiles and related policies of the virtual machine

Request:

```
GET /api/vms/171/policy_profiles?expand=resources&attributes=miq_policies
```

Response:

```
{
  "name": "policy_profiles",
  "count": 6,
  "subcount": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/171/policy_profiles/35",
      "id": "35",
      "name": "finance_policy_profile",
      "description": "Finance Servers Policy Profile",
      "set_type": "MiqPolicySet",
      "created_on": "2020-03-24T18:56:23Z",
      "updated_on": "2020-03-24T18:56:23Z",
      "guid": "d4546f30-15a7-48f3-a793-cfd6d1d849a2",
      "read_only": null,
      "set_data": null,
      "mode": "compliance",
      "owner_type": null,
      "owner_id": null,
      "userid": null,
      "group_id": null,
```

```
      "miq_policies": [
        {
          "href": "http://localhost:3000/api/policies/12",
          "id": "12",
          "name": "finance_server_policy",
          "description": "Finance Server Policy",
          "created_on": "2020-03-25T16:44:19Z",
          "updated_on": "2020-03-25T16:44:19Z",
          "expression": null,
          "towhat": "Vm",
          "guid": "e56d468c-1b1b-42f1-a4cb-4ea11924d7b3",
          "created_by": null,
          "updated_by": null,
          "notes": null,
          "active": true,
          "mode": "compliance",
          "read_only": null
        }
      ]
    }
  ],
  "actions": [
    {
      "name": "assign",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171/policy_profiles"
    },
    {
      "name": "unassign",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171/policy_profiles"
    },
    {
      "name": "resolve",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171/policy_profiles"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/vms/171/policy_profiles?attributes=miq_policies&expand=resources&offset=0",
    "first": "http://localhost:3000/api/vms/171/policy_profiles?attributes=miq_policies&expand=resources&offset=0",
    "last": "http://localhost:3000/api/vms/171/policy_profiles?attributes=miq_policies&expand=resources&offset=0"
  }
}
```


At this point, the compliance policy is properly assigned to the virtual machine. If configured via a schedule for the encompassing region, the compliance checks will run as scheduled. They can also be executed on the specific virtual machine via the API as follows:

Requesting a compliance check on the virtual machine

Request:

```
POST /api/vms/171
```


```
{
  "action" : "check_compliance"
}
```


Response:


```
{
  "success" : true,
  "message": "VM id:171 name:'aab-ldap' check compliance requested",
  "task_id": "35",
  "task_href": "http://localhost:3000/api/tasks/41",
  "href": "http://localhost:3000/api/vms/171"
}
```


Waiting for the compliance check to complete

One can find out when the compliance check completes by querying the task generated for it and look for the state attribute as follows:

```
GET /api/tasks/41
```

Response:


```
{
  "href": "http://localhost:3000/api/tasks/41",
  "id": "41",
  "name": "VM id:171 name:'aab-ldap' check compliance requested",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2020-04-09T21:16:07Z",
  "updated_on": "2020-04-09T21:16:12Z",
  "pct_complete": null,
  "context_data": null,
```

```
  "results": null,
  "miq_server_id": "3",
  "identifier": null,
  "started_on": "2020-04-09T21:16:09Z",
  "zone": null,
  "actions": [
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/tasks/41"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/tasks/41"
    }
  ]
}
```


Querying all compliances run for the virtual machine

Request:

```
GET /api/vms/171/compliances?expand=resources
```

Response:


```
{
  "name": "compliances",
  "count": 430,
  "subcount": 2,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/171/compliances/40",
      "id": "40",
      "resource_id": "171",
      "resource_type": "VmOrTemplate",
      "compliant": false,
      "timestamp": "2020-03-26T14:30:24Z",
      "updated_on": "2020-03-26T14:30:24Z",
      "event_type": "vm_compliance_check"
    },
    {
      "href": "http://localhost:3000/api/vms/171/compliances/41",
      "id": "41",
      "resource_id": "171",
      "resource_type": "VmOrTemplate",
      "compliant": false,
      "timestamp": "2020-03-26T20:11:53Z",
      "updated_on": "2020-03-26T20:11:53Z",
      "event_type": "vm_compliance_check"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/vms/171/compliances?expand=resources&offset=0",
    "first": "http://localhost:3000/api/vms/171/compliances?expand=resources&offset=0",
    "last": "http://localhost:3000/api/vms/171/compliances?expand=resources&offset=0"
  }
}
```


Querying the virtual machine and its latest compliance result including the last compliance status

Request:

```
GET /api/vms/171?attributes=last_compliance,last_compliance_status
```

Response:


```
{
  "href": "http://localhost:3000/api/vms/171",
  "id": "171",
  "vendor": "vmware",
  "format": null,
  "version": null,
  "name": "aab-ldap",
  "description": null,
  "location": "aab-ldap/aab-ldap.vmx",
  "config_xml": null,
  "autostart": null,
  "host_id": 2,
  "last_sync_on": null,
  "created_on": "2020-03-10T18:53:36Z",
  "updated_on": "2020-03-26T01:58:16Z",
  "storage_id": "7",
  "guid": "30be2fa7-6b95-4a62-a163-d3107fc79d94",
  "ems_id": "2",
  "last_scan_on": "2020-03-18T21:13:32Z",
```

```
  "last_scan_attempt_on": "2020-03-18T21:12:25Z",
  "uid_ems": "4cda25de-c41c-4f68-8b3d-0cca0dc993c4",
  "retires_on": null,
  "retired": null,
  "boot_time": null,
  "tools_status": null,
  "standby_action": null,
  "power_state": "on",
  "state_changed_on": "2020-03-10T18:53:36Z",
  "previous_state": null,
  "connection_state": "connected",
  "last_perf_capture_on": "2020-03-26T01:58:16Z",
  "registered": null,
  "busy": null,
  "smart": null,
  "memory_reserve": null,
  "memory_reserve_expand": null,
  "memory_limit": null,
  "memory_shares": null,
  "memory_shares_level": null,
  "cpu_reserve": null,
  "cpu_reserve_expand": null,
  "cpu_limit": -1,
  "cpu_shares": 200,
  "cpu_shares_level": "normal",
  "cpu_affinity": null,
  "ems_created_on": null,
  "template": false,
  "evm_owner_id": null,
  "miq_group_id": "1",
  "linked_clone": true,
  "fault_tolerance": null,
  "type": "ManageIQ::Providers::Vmware::InfraManager::Vm",
  "ems_ref": "vm-825",
  "ems_cluster_id": "3",
  "retirement_warn": null,
  "retirement_last_warn": null,
  "vnc_port": null,
  "flavor_id": "960",
  "availability_zone_id": "23",
  "cloud": true,
  "retirement_state": null,
  "cloud_network_id": null,
  "cloud_subnet_id": null,
  "cloud_tenant_id": "2",
  "raw_power_state": "ACTIVE",
  "publicly_available": null,
  "orchestration_stack_id": null,
  "retirement_requester": null,
  "tenant_id": "1",
  "resource_group_id": null,
  "deprecated": null,
  "storage_profile_id": null,
  "cpu_hot_add_enabled": null,
  "cpu_hot_remove_enabled": false,
  "memory_hot_add_enabled": true,
  "memory_hot_add_limit": null,
  "memory_hot_add_increment": null,
  "hostname": null,
  "ems_ref_type": "VirtualMachine",
  "restart_needed": null,
  "last_compliance_status": false,
  "last_compliance": {
    "id": "41",
    "resource_id": "1000000132824",
    "resource_type": "VmOrTemplate",
    "compliant": false,
    "timestamp": "2020-03-26T20:11:53Z",
    "updated_on": "2020-03-26T20:11:53Z",
    "event_type": "vm_compliance_check"
  },
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "add_event",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "refresh",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "shutdown_guest",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
```

```
    },
    {
      "name": "reboot_guest",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "start",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "stop",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "suspend",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "shelve",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "shelve_offload",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "pause",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "request_console",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "reset",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "request_retire",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "set_owner",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "set_ownership",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "set_miq_server",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "scan",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/vms/171"
    }
  ]
}
```


References: Infrastructure Management

View reference information about Infrastructure Automation Infrastructure Management.

- Support Matrix
- Methods Available for Automation
- Managing Authentication for Infrastructure Management (stand-alone)
- Integration with OpenShift Container Platform
- Self Service User Interface Guide

Supported platforms and features in Infrastructure Management

Infrastructure Management includes support for different platforms and features.

- Introduction
- Supported Browsers
- Supported Platforms
- Supported Providers
- Provisioning Support on Infrastructure Providers
- Provisioning Support on Cloud Providers
- Feature Support on Infrastructure Providers
- Feature Support on Cloud Providers
- Feature Support on Container Providers
- Feature Support on Network Providers
- Feature Support on Configuration Management Providers
- Remote Console Support
- Smart State Analysis Support

Chapter 1. Introduction

Infrastructure Management is also delivered as a virtual appliance that includes the following key components:

- CentOS Linux release (Core)
- Rails
- Ruby
- PostgreSQL
- Infrastructure Management Engine
- Apache

Chapter 2. Supported Browsers

For more information about browsers that are supported for viewing the Infrastructure Management web user interface, see the table in Supported browsers.

Chapter 3. Supported Platforms for Infrastructure Management

The following table lists platforms that can host the Infrastructure Management appliance, and platforms that can be managed by Infrastructure Management as providers.

Table 3.1. Supported Platforms

Table 3.1. Supported Platforms

Platform	Support for Infrastructure Management Appliance
Amazon EC2 [a]	Yes
IBM Cloud	Yes
Google Compute Engine	Yes
OpenShift Container Platform (OCP) 4.2, 4.3, and 4.4	No
OpenShift Container Platform (OCP) 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.13, 4.14, 4.15, and 4.16	No
Red Hat OpenShift Virtualization	Yes
Red Hat OpenStack Platform (OSP) 10, 13, 16 [b]	Yes
Red Hat Virtualization (RHV) 4.2, 4.3, 4.4, 4.4SP1 [c]	Yes
VMware vSphere 6.0, 6.5, 6.7, 7, 8	Yes

[a] As the pace of change in public cloud features is high, constantly changing and evolving, the compatibility for Infrastructure Management is as of the time of the GA of the product. Best effort basis during errata cycle for public cloud vendor introduced breaking changes (intended or unintended) to existing functionality. Any other changes or additions would be considered as a Request For Enhancement (RFE) which is evaluated for potential inclusion in future releases. RFE can be requested by way of your standard support mechanisms.

[b] See Red Hat OpenStack Platform Life Cycle for product lifecycle information.

[c] See Red Hat Enterprise Virtualization Life Cycle for product lifecycle information.

Chapter 4. Supported Providers

Provider support is divided into tiers:

- Tier 1: These providers are developed, tested, maintained, and supported by IBM. IBM accepts support and enhancement requests.
- Tier 2: These providers are delivered by an IBM partner, or by various sources into the ManageIQ open source community. IBM packages and ships these providers and accepts support cases. All support and enhancement updates need to come from the open source community.

Table 4.1. Supported Providers

Table 4.1. Supported Providers

Support tier	Provider
Tier 1	Amazon
	Amazon EKS
	Ansible Automation Platform
	Azure
	Azure Kubernetes Service
	Cisco Intersight
	Google Compute Engine
	Google Kubernetes Engine
	IBM Cloud
	IBM Cloud Automation Manager
	IBM Cloud Infrastructure Center
	IBM Cloud Kubernetes Service
	IBM Power HMC
	IBM Power Systems Virtual Servers
	IBM PowerVC
	Red Hat OpenShift
	OpenShift Virtualization
	OpenStack
	Oracle Cloud
	Oracle Kubernetes Engine
	Red Hat Satellite
	Red Hat Virtualization
	VMware Tanzu
	VMware vCenter
Tier 2	AWX
	Azure Stack
	Foreman
	Kubernetes
	KubeVirt
	Kyndryl AutoSDE
	Lenovo XClarity
	Nuage Network Manager
	oVirt
	Redfish
	VMware NSX-T
	VMware vCloud
	OpenStack Platform Director

Chapter 5. Provisioning Support

5.1. Infrastructure Providers

The following table outlines the status of support for provisioning on infrastructure providers.

Table 5.1. Infrastructure Provider Provisioning Support

Table 5.1. Infrastructure Provider Provisioning Support

Feature	Red Hat Virtualization (RHV) Manager	VMware vCenter	IBM Power HMC	OpenShift Virtualization (KubeVirt)	OpenStack undercloud
Provision VM/Instance using PXE	Yes	No	No	No	No
Provision VM/Instance using ISO	Yes	No	No	No	No
Clone from VM/Instance to VM/Instance	No	Yes	No	No	No
Provision from Template/Image to VM/Instance	Yes	Yes	Yes	Yes	No
Provision from VM/Instance to Template/Image	No	Yes	Yes	No	No

5.2. Cloud Providers

The following table outlines the status of support for provisioning on cloud providers.

Table 5.2. Cloud Provider Provisioning Support

Table 5.2. Cloud Provider Provision

Feature	OpenStack Platform (OSP)	Amazon EC2	Microsoft Azure	Google Compute Engine	IBM Cloud VPC	IBM PowerVS	IBM Power Systems Virtual Server
Provision VM/Instance using PXE	No	No	No	No	No	No	No
Provision VM/Instance using ISO	No	No	No	No	No	No	No
Clone from VM/Instance to VM/Instance	No	No	No	No	No	No	No
Provision from Template/Image to VM/Instance	Yes	Yes	Yes	Yes	No	Yes	Yes
Provision from VM/Instance to Template/Image	No	No	No	No	Yes	No	No
Orchestration Template Provisioning	Yes	Yes	Yes	No	No	No	No

Chapter 6. Feature Support

Infrastructure Management allows you to perform actions on the providers that it manages, and provides features for interacting with and providing information about those providers. This section outlines the status of support for actions that you can perform on providers, and the Infrastructure Management features that allow you to interact with those providers.

Note: You can run a rake command to retrieve a list of provider features supported across all platforms. Run the following command in your Infrastructure Management appliance to retrieve the supported features in a comma-separated values (.csv) file format. The comma-separated values are displayed on the command-line interface (CLI) by default. You can use a pipe to redirect the output to a file.

```
ruby /var/www/miq/vmdb/tools/feature_support_matrix.rb
```

6.1. Infrastructure Providers

The following tables outline the status of support for Infrastructure Management features on infrastructure providers.

6.1.1. Red Hat Virtualization

The following table outlines the status of support for Infrastructure Management features on Red Hat Virtualization (RHV) infrastructure providers.

Table 6.1.1 Red Hat Virtualization Provider Feature Support

Table 6.1.1. Red Hat Virtualization Provider Feature Support

Feature	RHV 4.2	RHV 4.3	RHV 4.4
Relationship Discovery	Yes	Yes	Yes
VM Drift Comparison	Yes	Yes	Yes
Track VM Genealogy	Yes	Yes	Yes
Capacity & Utilization	Yes	Yes	Yes
Capture VM/Instance Event Timelines	Yes	Yes	Yes
Reporting	Yes	Yes	Yes
Right Sizing	Yes	Yes	Yes
Chargeback	Yes	Yes	Yes
Remote Console VM Access	Yes [a]	Yes [a]	Yes [a]
Snapshot Creation and Removal	Yes	Yes	Yes
VM / Instance Compliance Enforcement	Yes	Yes	Yes
VM / Instance Policy Enforcement	Yes	Yes	Yes
VM / Instance Power Operations	Yes	Yes	Yes
VM / Instance Retirement	Yes	Yes	Yes
Alerts - VM Value Changed	Yes	Yes	Yes
Alerts - VM Reconfigured	Yes	Yes	Yes
Integrate with Service Catalogs	Yes	Yes	Yes
Virtual Machine Reconfiguration	Yes	Yes	Yes
Automation Work Flows	Yes	Yes	Yes
Tag Mapping from Provider	No	No	No
Tag Mapping to Provider	No	No	No
Provision VM/Instance using PXE	Yes	Yes	Yes
Provision VM/Instance using ISO	Yes	Yes	Yes

Feature	RHV 4.2	RHV 4.3	RHV 4.4
Provision from Template/Image to VM/Instance	Yes	Yes	Yes
Provision from VM/Instance to Template/Image	Yes	Yes	Yes
Host Power Operations	No	No	No
Provision Host	No	No	No
Clone from VM/Instance to VM/Instance	No	No	No
Cloud-Init	Yes	Yes	Yes
Customize Windows Templates with Sysprep during Provisioning	Yes	Yes	Yes
OVN Provider	Yes	Yes [b]	Yes [b]

[a] On some operating system and browser combinations.

[b] Technology preview.

6.1.2. VMware vSphere

The following table outlines the status of support for Infrastructure Management features on VMware vSphere infrastructure providers.

Table 6.1.2. VMware vSphere Provider Feature Support

Table 6.1.2. VMware vSphere Provider Feature Support

Feature	vSphere 6.0	vSphere 6.5	vSphere 6.7	vSphere 7	vSphere 8
Relationship Discovery	Yes	Yes	Yes	Yes	Yes
VM Drift Comparison	Yes	Yes	Yes	Yes	Yes
Track VM Genealogy	Yes	Yes	Yes	Yes	Yes
Capacity & Utilization	Yes	Yes	Yes	Yes	Yes
Capture VM/Instance Event Timelines	Yes	Yes	Yes	Yes	Yes
Disk Addition to VM/Instance	Yes	Yes	Yes	Yes	Yes
Reporting	Yes	Yes	Yes	Yes	Yes
Right Sizing	Yes	Yes	Yes	Yes	Yes
Chargeback	Yes	Yes	Yes	Yes	Yes
Remote Console VM Access	Yes [a]	Yes [a]	Yes [a]	Yes [a]	Yes [a]
Snapshot Creation and Removal	Yes	Yes	Yes	Yes	Yes
VM / Instance Compliance Enforcement	Yes	Yes	Yes	Yes	Yes
VM / Instance Policy Enforcement	Yes	Yes	Yes	Yes	Yes
VM / Instance Power Operations	Yes	Yes	Yes	Yes	Yes
VM / Instance Retirement	Yes	Yes	Yes	Yes	Yes
Alerts - Real Time	Yes	Yes	Yes	Yes	Yes
Alerts - VM Value Changed	Yes	Yes	Yes	Yes	Yes
Alerts - VM Reconfigured	Yes	Yes	Yes	Yes	Yes
Integrate with Service Catalogs	Yes	Yes	Yes	Yes	Yes
Virtual Machine Reconfiguration	Yes	Yes	Yes	Yes	Yes
Automation Work Flows	Yes	Yes	Yes	Yes	Yes
Tag Mapping from Provider	No	No	No	No	No
Tag Mapping to Provider	No	No	No	No	No
Provision VM/Instance using PXE	Yes	Yes	Yes	Yes	Yes
Provision VM/Instance using ISO	No	No	No	No	No
Provision from Template/Image to VM/Instance	Yes	Yes	Yes	Yes	Yes
Provision from VM/Instance to Template/Image	Yes	Yes	Yes	Yes	Yes
Host Power Operations	Yes	Yes	Yes	Yes	Yes
Customize Windows Templates with Sysprep during Provisioning	Yes	Yes	Yes	Yes	Yes
Clone from VM/Instance to VM/Instance	Yes	Yes	Yes	Yes	Yes

[a] On some operating system and browser combinations

6.1.3. IBM Power HMC

The following table outlines the status of support for IBM Power HMC infrastructure providers.

Table 6.1.3. IBM Power HMC Provider Feature Support

Table 6.1.3. IBM Power HMC Provider Feature Support

Feature	IBM Power HMC
Relationship Discovery	Yes
VM Drift Comparison	No
Track VM Genealogy	No
Capacity & Utilization	Yes
Capture VM/Instance Event Timelines	Yes
Reporting	Yes

Feature	IBM Power HMC
Right Sizing	Yes
Chargeback	No
Remote Console VM Access	No
Snapshot Creation and Removal	No
VM / Instance Compliance Enforcement	No
VM / Instance Policy Enforcement	No
VM / Instance Power Operations	Yes
VM / Instance Retirement	Yes
Alerts - Real Time	No
Alerts - VM Value Changed	No
Alerts - VM Reconfigured	No
Integrate with Service Catalogs	No
Virtual Machine Reconfiguration	No
Automation Work Flows	No
Tag Mapping from Provider	No
Tag Mapping to Provider	No
Provision VM/Instance using PXE	No
Provision VM/Instance using ISO	No
Provision from Template/Image to VM/Instance	Yes
Provision from VM/Instance to Template/Image	Yes
Host Power Operations	Yes
Provision Host	No
Clone from VM/Instance to VM/Instance	No

6.2. OpenStack Platform Director Infrastructure Providers

The following table outlines the status of support for Infrastructure Management features on OpenStack Platform (OSP) director infrastructure providers.

Table 6.2. OpenStack Platform Director Infrastructure Provider Feature Support

Table 6.2. OpenStack Platform Director Infrastructure Provider Feature Support

Feature	OpenStack Platform (OSP) Director
Relationship Discovery	Yes
Nodes Inventory	Yes
OpenStack Services Inventory	Yes
Nodes Drift Comparison	Yes
Nodes Smart State	Yes
Capacity & Utilization	Yes
Capture Infrastructure Event Timelines	Yes
Node Power Operation	Yes
Capacity Planning	Yes
Reporting	Yes
Add/Remove Node	Yes
Scale Down Node	Yes (Compute nodes only)
Scale Up Nodes	Yes (Compute nodes only)
Nodes Policy Enforcement	Yes
Nodes Evacuate	Yes
OpenStack Upgrade	No

6.3. Cloud Providers

The following section outlines the status of support for Infrastructure Management features on cloud providers.

6.3.1. OpenStack Platform

The following table outlines the status of support for Infrastructure Management features on OpenStack Platform (OSP) cloud providers.

- The following feature support information for IBM Cloud is based on:
  - IBM Cloud PowerVC Manager 1.4.4.0 Build: 20191205-1007
  - IBM Cloud Infrastructure Center 1.2.0

Table 6.3.1. OpenStack Platform Feature Support

Table 6.3.1. OpenStack Platform Feature Support

Feature	Red Hat OSP 10	Red Hat OSP 13	Red Hat OSP 16	OpenStack	IBM PowerVC	IBM Cloud Infrastructure Center
Relationship Discovery	Yes	Yes	Yes	Yes	Yes	Yes
Instance Drift Comparison	Yes	Yes	Yes	Yes	No	No
Track Instance Genealogy	Yes	Yes	Yes	Yes	Yes	Yes
Capacity & Utilization	Yes	Yes	Yes	Yes	Yes	Yes
Capture Cloud Layer Timelines	Yes	Yes	Yes	Yes	Yes	Yes
Capture Instance Event Timelines	Yes	Yes	Yes	Yes	Yes	Yes
Disk Addition to Instance	Yes	Yes	Yes	Yes	N/A	Yes
Key Pairs Inventory	Yes	Yes	Yes	Yes	Yes	Yes
Key Pairs Management	Yes	Yes	Yes	Yes	Yes	Yes
Reporting	Yes	Yes	Yes	Yes	Yes	Yes
Right Sizing	Yes	Yes	Yes	Yes	Yes	Yes
Chargeback	Yes	Yes	Yes	Yes	Yes	Yes
Remote Console Instance Access	Yes	Yes	Yes	Yes (VNC)	Yes (for NovaLink-managed hosts only)	No
Snapshot Creation and Removal	Yes	Yes	Yes	Yes	Yes	Yes
Instance Compliance Enforcement	Yes	Yes	Yes	Yes	No	No
Instance Policy Enforcement	Yes	Yes	Yes	Yes	Yes	Yes
Instance Power Operations	Yes	Yes	Yes	Yes	Yes	Yes
Instance Retirement	Yes	Yes	Yes	Yes	Yes	Yes
Alerts - Real Time	Yes	Yes	Yes	Yes	Yes	Yes
Alerts - Instance Value Changed	No	No	No	No	No	No
Alerts - Reconfigured	No	No	No	No	No	No
Integrate with Service Catalogs	Yes	Yes	Yes	Yes	Yes	Yes
Virtual Machine Reconfiguration	Yes	Yes	Yes	Yes	Yes	Yes
Volume Inventory	Yes	Yes	Yes	Yes	Yes	Yes
Volume Creation/Deletion	Yes	Yes	Yes	Yes	Yes	Yes
Instance Migration	Yes	Yes	Yes	Yes	Yes	Yes**
Automation Work Flows	Yes	Yes	Yes	Yes	Yes	Yes
Tag Mapping from Provider	No	No	No	No	Yes	No
Tag Mapping to Provider	No	No	No	No	No	No
Storage Manager	Yes	Yes	Yes	Yes	Yes	Yes
Provision Instance using PXE	No	No	No	N/A	N/A	N/A
Provision Instance using ISO	No	No	No	N/A	N/A	N/A
Provision from Template/Image to Instance	Yes	Yes	Yes	Yes	Yes	Yes
Provision from Image Snapshot to Instance	Yes	Yes	Yes	Yes	No	N/A
Provision from Volume to Instance	Yes	Yes	Yes	Yes	No	N/A
Provision from Volume Snapshot to Instance	Yes	Yes	Yes	Yes	No	N/A
Clone from Instance to Instance	No	No	No	No	No	No
Cloud-Init	Yes	Yes	Yes	Yes	Yes	N/A
Sysprep Windows Templates	Yes	Yes	Yes	Yes	N/A	N/A

Notes:

- Features marked with "*" are only supported with z/VM hypervisor.
- Features marked with "**" are only supported with KVM hypervisor.
- Features not marked with "*" or "**" are valid and supported for both z/VM and KVM hypervisors.
- Features listed with a “NO” are not valid for both KVM and z/VM.

6.3.2. IBM Cloud Feature Support

The following table outlines the status of support for Infrastructure Management features on IBM Cloud providers.

Table 6.3.2. IBM Cloud Feature Support

Table 6.3.2. IBM Cloud Feature Support

Feature	IBM Cloud VPC	IBM Power Systems Virtual Server
Relationship Discovery	Yes	Yes
Instance Drift Comparison	No	No
Track Instance Genealogy	Yes	Yes
Capacity & Utilization	Yes	No
Capture Cloud Layer Timelines	No	No
Capture Instance Event Timelines	Yes	No
Disk Addition to Instance	No	Yes

Feature	IBM Cloud VPC	IBM Power Systems Virtual Server
Key Pairs Inventory	Yes	Yes
Key Pairs Management	No	Yes
Reporting	Yes	Yes
Right Sizing	No	No
Chargeback	Yes (Limited to inventory allocation)	Yes (Limited to inventory allocation)
Remote Console Instance Access	No	No
Snapshot Creation and Removal	No	No
Instance Compliance Enforcement	No	No
Instance Policy Enforcement	Yes	Yes
Instance Power Operations	Yes	Yes
Instance Retirement	No	Yes
Alerts - Real Time	No	No
Alerts - Instance Value Changed	No	No
Alerts - Reconfigured	No	No
Integrate with Service Catalogs	No	Yes
Virtual Machine Reconfiguration	No	No
Volume Inventory	Yes	Yes
Volume Creation/Deletion	Yes	Yes
Instance Migration	N/A	N/A
Automation Work Flows	No	No
Tag Mapping from Provider	Yes	No
Tag Mapping to Provider	No	No
Storage Manager	Yes	Yes
Provision Instance using PXE	No	No
Provision Instance using ISO	No	No
Provision from Template/Image to Instance	Yes	Yes
Provision from Image Snapshot to Instance	No	No
Provision from Volume to Instance	No	No
Provision from Volume Snapshot to Instance	No	No
Clone from Instance to Instance	No	No
Cloud-Init	No	Yes
Sysprep Windows Templates	N/A	N/A

6.3.3. Amazon EC2

The following table outlines the status of support for Infrastructure Management features on Amazon EC2 cloud providers.

Table 6.3.3. Amazon EC2 Feature Support

Table 6.3.3. Amazon EC2 Feature Support

Feature	Amazon EC2
Relationship Discovery	Yes
VM Drift Comparison	Yes
Track VM Genealogy	Yes
Capacity & Utilization	Yes
Capture VM/Instance Event Timelines	Yes
Disk Addition to VM/Instance	Yes
Key Pairs Inventory	Yes
Key Pairs Management	Yes
Reporting	Yes
Right Sizing	Yes
Chargeback	Yes
Remote Console VM Access	No
Snapshot Creation and Removal	No
VM / Instance Compliance Enforcement	Yes
VM / Instance Policy Enforcement	Yes
VM / Instance Power Operations	Yes
VM / Instance Retirement	Yes
Alerts - Real Time	No
Alerts - VM Value Changed	No
Alerts - Reconfigured	No
Integrate with Service Catalogs	Yes
Virtual Machine Reconfiguration	No
Volume Inventory	No

Feature	Amazon EC2
Volume Creation/Deletion	No
VM Migration	No
Automation Work Flows	Yes
Tag Mapping from Provider	Yes
Tag Mapping to Provider	No
Network Manager - Read Only	Yes
Network Manager - CRUD Actions Supported	No
Storage Manager	Yes
Provision VM/Instance using PXE	No
Provision VM/Instance using ISO	No
Provision from Template/Image to VM/Instance	Yes
Clone from VM/Instance to VM/Instance	No

6.3.4. Microsoft Azure

The following table outlines the status of support for Infrastructure Management features on Microsoft Azure cloud providers.

Table 6.3.4. Microsoft Azure Feature Support

Table 6.3.4. Microsoft Azure Feature Support

Feature	Microsoft Azure
Relationship Discovery	Yes
VM Drift Comparison	Yes
Track VM Genealogy	Yes
Capacity & Utilization	Yes
Capture Cloud Layer Timelines	N/A
Capture VM/Instance Event Timelines	Yes
Disk Addition to VM/Instance	Yes
Key Pairs Inventory	No
Key Pairs Management	Yes
Optimization - Bottleneck Identification	No
Reporting	Yes
Right Sizing	Yes
Chargeback	Yes
Remote Console VM Access	No
Snapshot Creation and Removal	No
VM / Instance Compliance Enforcement	Yes
VM / Instance Policy Enforcement	Yes
VM / Instance Power Operations	Yes
VM / Instance Retirement	Yes
Alerts - Real Time	No
Alerts - VM Value Changed	No
Alerts - Reconfigured	No
Integrate with Service Catalogs	Yes
Virtual Machine Reconfiguration	No
Volume Inventory	No
Volume Creation/Deletion	No
VM Migration	No
Automation Work Flows	No
Tag Mapping from Provider	Yes
Tag Mapping to Provider	No
Network Manager - Read Only	Yes
Network Manager - CRUD Actions Supported	No
Provision VM/Instance using PXE	No
Provision VM/Instance using ISO	No
Provision from Template/Image to VM/Instance	Yes
Clone from VM/Instance to VM/Instance	No

6.3.5. Google Compute Engine

The following table outlines the status of support for Infrastructure Management features on Google Compute Engine cloud providers.

Table 6.3.5. Google Compute Engine Feature Support

Table 6.3.5. Google Compute Engine Feature Support

Feature	Google Compute Engine
Relationship Discovery	Yes
VM Drift Comparison	No
Track VM Genealogy	Yes
Capacity & Utilization	Yes
Capture Cloud Layer Timelines	No
Capture VM/Instance Event Timelines	Yes
Disk Addition to VM/Instance	No
Key Pairs Inventory	Yes
Key Pairs Management	No
Optimization - Bottleneck Identification	No
Reporting	Yes
Right Sizing	Yes
Chargeback	Yes
Remote Console VM Access	No
Snapshot Creation and Removal	No
VM / Instance Compliance Enforcement	Yes
VM / Instance Policy Enforcement	Yes
VM / Instance Power Operations	Yes
VM / Instance Retirement	Yes
Alerts - Real Time	No
Alerts - VM Value Changed	No
Alerts - Reconfigured	No
Integrate with Service Catalogs	Yes
Virtual Machine Reconfiguration	No
Volume Inventory	Yes
Volume Creation/Deletion	No
VM Migration	No
Automation Work Flows	Yes
Tag Mapping from Provider	No
Tag Mapping to Provider	No
Network Manager - Read Only	Yes
Network Manager - CRUD Actions Supported	No
Storage Manager	No
Provider TLS	No
Provision VM/Instance using PXE	No
Provision VM/Instance using ISO	No
Provision from Template/Image to VM/Instance	Yes
Provision from VM/Instance to Template/Image	No
Host Power Operations	No
Provision Host	No
Clone from VM/Instance to VM/Instance	No

6.3.6. Oracle Cloud

The following table outlines the status of support for Infrastructure Management features on Oracle cloud providers.

Table 6.3.6. Oracle Cloud Feature Support

Table 6.3.6. Oracle Cloud Feature Support

Feature	Oracle Cloud
Relationship Discovery	Yes
VM Drift Comparison	No
Track VM Genealogy	Yes
Capacity & Utilization	Yes
Capture Cloud Layer Timelines	No
Capture VM/Instance Event Timelines	Yes
Disk Addition to VM/Instance	No
Key Pairs Inventory	Yes
Key Pairs Management	No
Optimization - Bottleneck Identification	No
Reporting	Yes
Right Sizing	No
Chargeback	No
Remote Console VM Access	No

Feature	Oracle Cloud
Snapshot Creation and Removal	No
VM / Instance Compliance Enforcement	No
VM / Instance Policy Enforcement	Yes
VM / Instance Power Operations	Yes
VM / Instance Retirement	Yes
Alerts - Real Time	No
Alerts - VM Value Changed	No
Alerts - Reconfigured	No
Integrate with Service Catalogs	Yes
Virtual Machine Reconfiguration	No
Volume Inventory	Yes
Volume Creation/Deletion	No
VM Migration	No
Automation Work Flows	Yes
Tag Mapping from Provider	No
Tag Mapping to Provider	No
Network Manager - Read Only	Yes
Network Manager - CRUD Actions Supported	No
Storage Manager	No
Provider TLS	No
Provision VM/Instance using PXE	No
Provision VM/Instance using ISO	No
Provision from Template/Image to VM/Instance	Yes
Provision from VM/Instance to Template/Image	No
Host Power Operations	No
Provision Host	No
Clone from VM/Instance to VM/Instance	No

6.4. Container Providers

The following section outlines the status of support for Infrastructure Management features on Red Hat OpenShift Container Platform container providers.

Table 6.4. Red Hat OpenShift Container Platform Provider Feature Support

Table 6.4. Red Hat OpenShift Container Platform Provider Feature Support

Feature	Red Hat OpenShift Container Platform 4.x
Relationship Discovery	Yes
Track Container Image Relationship	Yes
Capacity & Utilization	Yes
Capture Container Event Timelines	Yes
Reporting	Yes
Chargeback	Yes
Remote Cockpit Node Access	No
Container Node Compliance Enforcement	Yes
Container Node Policy Enforcement	Yes
Replicator Compliance Enforcement	Yes
Replicator Policy Enforcement	Yes
Pod Compliance Enforcement	Yes
Pod Policy Enforcement	Yes
Image Compliance Enforcement	Yes
Image Policy Enforcement	Yes
Template Provisioning	No
Smart State	No
OpenSCAP Execution and Report	No
Cross Linking Underlying Infrastructure	Yes
Tag Mapping From Provider	No
Tag Mapping To Provider	Yes

6.5. Network Providers

This section outlines the status of support for Infrastructure Management features on network providers.

Table 6.5. Network Provider Feature Support

Table 6.5. Network Provider Feature Support

Feature	Red Hat OpenStack Platform (OSP)	Amazon EC2	Microsoft Azure	Google Compute Engine	Nuage	VMware NSX-t
Network Manager Relationships	Yes	Yes	Yes	Yes	Yes	Yes
Relationship Discovery	Yes	Yes	Yes	Yes	Yes	Yes
Cloud Network Inventory	Yes	Yes	Yes	Yes	Yes	Yes
Cloud Subnet Inventory	Yes	Yes	Yes	Yes	Yes	Yes
Network Router Inventory	Yes	No	No	Yes	Yes	Yes
Security Groups	Yes	Yes	Yes	Yes	No	Yes
Floating IP Addresses	Yes	Yes	Yes	Yes	Yes	No
Network Ports	Yes	Yes	Yes	Yes	Yes	Yes
Load Balancer Inventory	No	Yes	Yes	Yes	No	No
Create/Update/Delete Network	Yes	No	No	No	Yes	Yes
Create/Update/Delete Subnet	Yes	No	No	No	Yes	No
Create/Update/Delete Routers	Yes	No	No	No	Yes	Yes

6.6. Configuration Management Providers

The following table outlines the status of support for Infrastructure Management features on configuration management providers.

Table 6.6. Configuration Management Provider Feature Support

Table 6.6. Configuration Management Provider Feature Support

Feature	Ansible
Groups Inventory	Yes
Configured Systems Inventory	Yes
Job Template Inventory	Yes
Integrate with Service Catalog	Yes
Automate Workflows	Yes

Note: Infrastructure Management supports Ansible 3.7 and later as a configuration management provider.

Chapter 7. Remote Console Support

Infrastructure Management supports the following remote consoles:

Table 7.1. Remote consoles

Table 7.1. Remote consoles

Provider	Connection type
vSphere 6.0	VNC
	VMRC
	WebMKS
vSphere 6.5+	VMRC
	WebMKS
RHV	SPICE
SCVMM	N/A
KubeVirt	SPICE
OSV	SPICE

Note: As of 1 September 2015, new versions of Chrome no longer support VMRC consoles. VMRC requires NPAPI plug-in support. Chrome removed support for NPAPI plug-ins in version 45.0.2454, which was released on this date.

Chapter 8. Smart State analysis support

Infrastructure Management allows you to analyze virtual machines and instances on the providers it manages to collect information about the base operating system on those virtual machines or instances, their virtual hardware, installed applications, and other details. This operation is known as SmartState analysis. The following tables outline the status of support for SmartState analysis of virtual machines and instances by guest operating system, file systems, volume managers, and file formats.

Table 8.1. Volume managers

Table 8.1. Volume managers

Volume manager	OS	Support
Logical Disk Manager (LDM)	Windows	Yes

Volume manager	OS	Support
Logical Volume Manager (Linux)	Linux	Yes

Table 8.2. File systems

Table 8.2. File systems

File system	OS	Support
FAT	Windows	Yes
FAT32	Windows	Yes
NTFS	Windows	Yes
ReFS	Windows	No
Ext3	Linux	Yes
Ext4	Linux	Yes
XFS	Linux	Yes
ReiserFS	Linux	Yes
CDfs	Linux	Yes

Table 8.3. File formats

Table 8.3. File formats

File format	OS	Support
Registry	Windows	Yes
Win32 Accounts	Windows	Yes
Win32 Event Log	Windows	Yes
Win32 Services	Windows	Yes
Win32 Software	Windows	Yes
Win32 System	Windows	Yes
SQLite	Linux	Yes
Berkeley DB (BDB)	Linux	Yes
Conary	Linux	Yes
RPM (RHEL 6, 7)	Linux	Yes
/etc/passwd	Linux	Yes
/etc/group	Linux	Yes
/etc/init.d	Linux	Yes
/etc/systemd	Linux	Yes

- Integration with OpenShift Container Platform
  - Overview
  - Prerequisites
  - Retrieving the OpenShift Container Platform Management Token
  - Configuring Infrastructure Automation
  - Configuring Infrastructure Automation Capacity and Utilization
    - Enabling SmartState Analysis
  - Adding OpenShift Container Platform as a Container Provider
  - Container Image Scanning
  - Configuring Image Scanning
  - Scheduling A Recurring Scan
  - Working with Images
    - Viewing Results
    - Manual Scanning
    - Evaluating Compliance
    - Generating a Report on Images
  - OpenSCAP Policy Profile
    - Customizing the Scanning Policy Profile
  - Controlling OpenShift Pod Execution
  - Reference

Integration with OpenShift Container Platform

Overview

This guide walks you through adding a Red Hat OpenShift Container Platform cluster to a Infrastructure Automation container provider catalog. This deployment focuses on enabling the OpenShift Container Platform cluster metrics plug-in, so that Infrastructure Automation can collect information from an OpenShift Container Platform cluster upon integration.

Each procedure in this guide is covered in greater detail in the Infrastructure Automation and OpenShift Container Platform product documentation. However, links to the corresponding sections are provided for more detail.

The following sections will describe the required configuration for both products prior to integration.

Prerequisites

This guide assumes that you have:

- Already deployed Infrastructure Automation on your chosen platform
- Already deployed OpenShift Container Platform

When enabling metrics on OpenShift Container Platform, you can store your metrics data on *persistent* or *non-persistent* storage. To use persistent storage, you need to provision a persistent volume specifically for this purpose before configuring the metrics components. See Persistent Volumes in the OpenShift Container Platform *Architecture* documentation for more information.

Retrieving the OpenShift Container Platform Management Token

After enabling cluster metrics on your OpenShift Container Platform deployment, retrieve the *management token* while you are still logged in to the OpenShift Container Platform host. This will be required later in Adding OpenShift Container Platform as a Container Provider.

Run the following to obtain the token needed to add a Red Hat OpenShift provider:

1. Obtain the **`management`** service account token name:

```
# oc describe sa -n $project_name $service_account_name
...
Tokens:  management-admin-token-0f3fh
         management-admin-token-q7a87
```

2. Select and describe one of the tokens to retrieve the full token output, replacing **`management-admin-token-0f3fh`** with the name of your token:

```
# oc describe secret -n $project_name management-admin-token-0f3fh
...
Data
====
token:  eyJhbGciOiJSUzI1NiI...
```

Configuring Infrastructure Automation

Configuring Infrastructure Automation involves two steps:

1. Configuring Infrastructure Automation Capacity and Utilization, and
2. Enabling SmartState Analysis

These steps are required to allow Infrastructure Automation to collect metrics from OpenShift Container Platform Enabling OpenShift Container Platform Metrics and use them to perform a SmartState analysis. You can choose different servers to perform either function; the following sections assume that you will.

Configuring Infrastructure Automation Capacity and Utilization

For metrics collection to work properly, you also need to configure Infrastructure Automation to allow for all three **Capacity & Utilization** server roles, which are available from the settings menu under **Configuration > Server > Server Control**.

To enable these server roles:

1. Click **Configuration**, then select the server to configure from menu: **Settings > Zone** in the accordion menu on the left.
2. Browse to the **Server Roles** list in the menu: **Server > Server Control** section. From there, set the required capacity and utilization roles to **ON**, namely:
    1. **Capacity & Utilization Coordinator**
    2. **Capacity & Utilization Data Collector**
    3. **Capacity & Utilization Data Processor**
3. Click **Save**.

Data collection is enabled immediately. However, the first collection begins 5 minutes after the server is started, and every 10 minutes after that. Therefore, the longest the collection takes after enabling the Capacity & Utilization Collector role is 10 minutes. The first collection from a particular provider may take a few minutes since Infrastructure Automation is gathering data points going one month back in time.

For more information, see Capacity and Utilization Collection in the *Deployment Planning Guide*.

Enabling SmartState Analysis

After enabling the required server roles, enable SmartState analysis. See Smart State Analysis Support from the Support Matrix.

Enabling SmartState analysis is similar to Assigning the Capacity and Utilization Server Roles, in that the procedure also involves enabling server roles on a specific server. To do so:

1. Click **Configuration**.

2. Select the server to configure from menu: **Settings**[Zone] in the left pane of the appliance.
3. Browse to the **Server Roles** list in the menu: **Server > Server Control** section. From there, set the appropriate SmartState roles to **ON**. Namely:
    1. **SmartProxy**
    2. **SmartState Analysis**
4. Click **Save**.

Adding OpenShift Container Platform as a Container Provider

At this point, you should now be ready to add your OpenShift cluster to Infrastructure Automation as a container provider. To do so, prepare the token you retrieved in Retrieving the OpenShift Container Platform Management Token and follow the procedure below:

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click (**Add a New Containers Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **OpenShift Container Platform**.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **`default`**.
6. In the **Default** tab, configure the following for the OpenShift provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.

          Note:

          You can obtain your OpenShift Container Platform provider's CA certificate for all endpoints (default, metrics, alerts) from **`/etc/origin/master/ca.crt`**. Paste the output (a block of text starting with **`-----BEGIN CERTIFICATE-----`**) into the **Trusted CA Certificates** field.
        - **SSL without validation**: Authenticate the provider insecurely (not recommended).
    2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.

       Important:

       The **Hostname** must use a unique fully qualified domain name.
    3. Enter the **API Port** of the provider. The default port is **`6443`**.
    4. Enter a token for your provider in the **Token** box.

       Note:

       To obtain a token for your provider, run the **`oc get secret`** command on your provider; see Obtaining an OpenShift Container Platform Management Token.

       For example:

       # oc get secret –namespace management-infra management-admin-token-8ixxs –template='' | base64 –decode
    5. Click **Validate** to confirm that Infrastructure Automation can connect to the OpenShift Container Platform provider.
7. In the **Metrics** endpoint tab optionally configure the metrics service details:
    1. Select a service type (e.g.: Prometheus) or leave disabled
    2. Select a **Security Protocol** method to specify how to authenticate the service:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
        - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)
    3. Enter the **Hostname** (or IPv4 or IPv6 address) of the metrics service.

       In order to find the hostname you can use the **`oc get route`** command.

       You can retrieve the Prometheus route by running:

```
oc get route prometheus-k8s -n openshift-monitoring
```

Example:

```
oc get route prometheus-k8s -n openshift-monitoring
NAME             HOST/PORT                                                PATH             SERVICES   PORT
TERMINATION          WILDCARD
prometheus-k8s   prometheus-k8s-openshift-monitoring.ocp.example.com  prometheus-k8s   web
reencrypt/Redirect   None
```

In this example you would use **`prometheus-k8s-openshift-monitoring.ocp.example.com`** as the metrics hostname.

4. Enter the **API Port** of your metrics endpoint uses a non-standard port for access. The default port is **`443`**.
5. Click **Validate** to confirm that Infrastructure Automation can connect to the metrics endpoint.

8. In the **Virtualization** endpoint tab optionally configure the virtualization service details:
    1. Select a service type (OpenShift Virtualization) or leave disabled
    2. Select a **Security Protocol** method to specify how to authenticate the service:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
        - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)
    3. Enter the **Hostname** (or IPv4 or IPv6 address) of the virtualization service.
    4. Enter the **API Port** of your virtualization service. The default port is **`443`**.
    5. Enter a token for your provider in the **Token** box.
9. Click the **Advanced** tab to add image inspector settings for scanning container images on your provider using OpenSCAP.

    Note:

    - These settings control downloading the image inspector container image from the registry and obtaining the Common Vulnerabilities and Exposures (CVE) information (for effective scanning) via a proxy.
    - CVE URL that Infrastructure Automation requires to be open for OpenSCAP scanning: https://www.redhat.com/security/data/metrics/ds/. This information is based on the source code of OpenSCAP.

    1. Enter the proxy information for the provider in either **HTTP**, **HTTPS**, or **NO Proxy** depending on your environment.
    2. Enter the **Image-Inspector Repository** information. For example, **`openshift3/image-inspector`**.
    3. Enter the **Image-Inspector Registry** information. For example, **`registry.access.redhat.com`**.
    4. Enter the **Image-Inspector Tag** value. A tag is a mark used to differentiate images in a repository, typically by the application version stored in the image.
    5. Enter **`https://www.redhat.com/security/data/metrics/ds/`** in **CVE location**.
10. Click **Add**.

Note:

You can also set global default image-inspector settings for all OpenShift providers in the advanced settings menu by editing the values under **`ems_kubernetes`**, instead of setting this for each provider.

For example:

```
:image_inspector_registry: registry.access.redhat.com
:image_inspector_repository: openshift3/image-inspector
```

Container Image Scanning

Configuring Image Scanning

Infrastructure Automation manages vulnerability scanning of container images. When an OpenShift provider is added, OpenShift images from the internal registry are discovered. To enable image scanning, perform the following configuration steps:

1. Browse to menu: **Compute > Containers > Providers**.
2. Select the checkboxes of the OpenShift providers on which to enable scanning.
3. From the **Policy** pull-down menu, click **Manage Policies**.
4. Select the **OpenSCAP profile** checkbox.

5. Click **Save**.

This action will trigger a SmartState analysis, or scan, of all images referenced by the OpenShift provider. The initial scan may take several hours to complete, depending on the number and size of images. The scan occurs in the OpenShift provider, which Infrastructure Automation receives and records in the database. OpenShift limits the number of scanning pods; only three images can be scanned simultaneously.

Scheduling A Recurring Scan

Software vulnerability databases are updated frequently. To apply these updates, a rescan is required. To schedule a recurring scan of container images:

 :schedule openscap scan

1. Click **Configuration**.
2. From **Settings > Zones** in the left pane of the appliance, select **Schedules**.
3. From the drop-down menu, click **Configuration > Add a new Schedule**.
4. Type an arbitrary **Name**.
5. Type an arbitrary **Description**.
6. Ensure the **Active** checkbox is selected.
7. In **Action**, select **Container Image Analysis**.
8. In **Filter**, select **All Container Images for Containers Provider**, **OpenShift**.
9. In **Run**, set the schedule as desired.
10. Set the **Time Zone**, **Starting Date**, and **Starting Time**.
11. Click **Add**.

Working with Images

Viewing Results

Image scanning results are displayed in each image summary page.

1. Select menu: **Compute > Containers > Container Images**.
2. Click the desired image.

For an OpenSCAP HTML report, locate the **Configuration** section and select **OpenSCAP HTML**.

 :container configuration

For compliance and scanning history information, locate the **Compliance** section and note the **Status** field or select **Available** from the **History** field.

 :container scan history

Manual Scanning

SmartState analysis scanning may be initiated manually for images. From an image summary page, select **Configuration > Perform SmartState Analysis**. Refreshing the image page will reflect the latest scan results and compliance history.

Evaluating Compliance

If the image scan policy has been updated since the last scan, compliance conditions may be re-evaluated. From an image summary page, select **Policy > Check Compliance of Last Known Configuration**. Refreshing the image page will reflect the latest compliance history.

Generating a Report on Images

You can output the results of an OpenSCAP scan of images to a report for an overview of the security risk level of images. The **Images by Failed OpenSCAP Rule Results are included with Infrastructure Automation and shows whether the image passed or failed OpenSCAP policy criteria, and the security risk.

Note: You can also create a copy of this report and edit it to contain additional information, such as the project name where the image is used to produce more useful results. For more information, see Editing a Report and Reportable Fields in Infrastructure Automation in *Monitoring, Alerts, and Reporting* for instructions on customizing reports.

Steps to create a report to show image compliance:

1. Browse to menu: **Overview > Reports**.
2. Click the menu: **Reports > All Reports** accordion.
3. Browse to menu: **Configuration Management > Containers > Images by Failed OpenSCAP Rule Results** to run a report that shows which images failed the OpenSCAP compliance.
4. Click **Queue**.

5. The report generation is placed in the queue and its status shows in the reports page.

		Queued At	Run At	Source	Username	Group	Status
☐	⏵	03/12/18 04:31:57 UTC		Requested by user	ocpadmin	EvmGroup-container_administrator_updated	Queued
☐	→	03/07/18 08:43:25 UTC	03/07/18 08:44:00 UTC	Requested by user	ocpadmin	EvmGroup-container_administrator_updated	Complete

6. Click **(Refresh this page)** to update the status.
7. Browse to the **Saved Reports** accordion, and click the report when it completes.
8. Click the report download buttons for the type of export you want. The report is automatically named with the type of report and date.
   - Click **Download this report in text format** to download as text.
   - Click **Download this report in CSV format** to download as a comma-separated file.
   - Click **Download this report in PDF format** to download as PDF.

OpenSCAP Policy Profile

Infrastructure Automation is pre-configured with a default scanning policy profile. This includes conditions to scan and identify compliance, as well as annotate compliance failure. SmartState analysis is performed when new images are added to OpenShift.

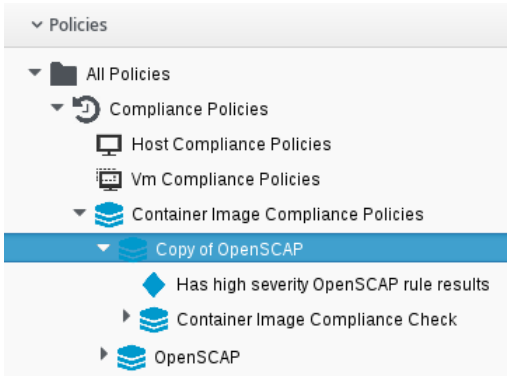
Customizing the Scanning Policy Profile

The built-in OpenSCAP policy profile cannot be edited. However, you can assign *edited* copies of these policies to a new policy profile. This provides you the ability to create a customized version of the built-in OpenSCAP policy profile.

To do so, you must first copy the policy that you want to customize:

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, select **Container Image Compliance Policies**, and click **OpenSCAP**.
3. Click **Configuration**, and an option to copy the policy appears; for example, **Copy this Container Image Policy**.
4. Click **OK** to confirm.

The new policy is created with a prefix of **Copy of** in its description, and it can be viewed in the Policies accordion.



You can now edit the copied policy. After editing copied policies, you can add them to a new policy profile. For instructions on how to edit policies, create a new policy profile, and add policies to it, see the Policies and Profiles guide. Once you have a customized policy profile, you can assign it to a containers provider.

Controlling OpenShift Pod Execution

Through the default policy profile, non-compliant images receive the control policy action **Mark as Non-Compliant**. This action annotates the **image** object (not to be confused with the **imagestream** object) with *images.openshift.io/deny-execution=true*. This annotation may be used to prevent nodes from running non-compliant images.

Reference

More information about OpenSCAP, see visit the OpenSCAP web site.

- Methods Available for Automation
  - Methods Available for Use with Infrastructure Automation
  - $evm.root
    - Method Hierarchy
    - Base Methods
    - Availability Zones (availability_zone)
    - Classifications (classification)
    - Cloud Networks (cloud_network)
    - Cloud Resource Quotas (cloud_resource_quota)
    - Cloud Subnets (cloud_subnet)
    - Cloud Tenants (cloud_tenant)
    - Customization Template (customization_template)
    - Clusters (ems_cluster)
    - Management System Events (ems_event)
    - Management System Folders (ems_folder)
    - Providers (ext_management_system)
      - Cloud Providers (ems_cloud)
    - Firewall Rules (firewall_rule)
    - Flavors (flavor)
    - Floating IPs (floating_ip)
    - Guest Applications (guest_application)
    - Guest Device (guest_device)
    - Hardware (hardware)
    - Hosts (host)
      - Hosts: Vmware ESX (host_vmware_esx)
    - LAN (lan)
    - Groups (miq_group)
    - Request (miq_request)
      - Automation Request (automation_request)
      - Host Provision Request (miq_host_provision_request)
      - VM Provision Request (miq_provision_request)
    - Request Task (miq_request_task)
      - Automation Task (automation_task)
      - Host Provision Task (miq_host_provision)
      - VM Provision Task (miq_provision)
        - Adding Disks to OpenStack Instance Provisioning
        - Booting OpenStack Instances from Volume
        - VM Provisioning for Clouds Task (miq_provision_cloud)
      - Service Template Provision Task (service_template_provision_task)
      - Service Reconfiguration Task (service_reconfigure_task)
      - VM Migrate Task (vm_migrate_task)
    - Proxies (miq_proxy)
    - Servers (miq_server)
    - Network (network)
    - PXE Image (pxe_image)
    - PXE Server (pxe_server)
    - Security Groups (security_group)
    - Service (service)
    - Service Resource (service_resource)
    - Service Template (service_template)
    - Snapshot (snapshot)
    - Datastores (storage)
    - Switch (switch)
    - User (user)
    - Virtual Machines and Templates (vm_or_template)
      - VMs (vm)
        - VMs for Clouds (vm_cloud)
    - Windows Image (windows_image)
    - Creating Categories and Tags
      - Category and Tags Methods Example
    - Quota
- Terminology

Methods Available for Automation

Methods Available for Use with Infrastructure Automation

Methods can be used from within Infrastructure Automation to create custom actions and workflows for the objects managed for your Infrastructure Automation Infrastructure. This document describes the methods available for use in Infrastructure Automation. This document is organized by the object hierarchy in the Automate Model.

Note:

Users of Infrastructure Automation can construct custom automation methods in Ruby to extend the product. Infrastructure Automation ships with a core set of Ruby gems used by the Infrastructure Automation Rails Application. The Ruby gems in this set are subject to change, and have changed since the previous version. If you are calling gems using Automate that are no longer in the Infrastructure Automation appliance, you can install them by using the **`gem install`** command.

While gems can be imported into automation methods using **`require`**, it is recommended that the authors of the automation methods clearly document the use of gems either in the core set or a custom set. It is the responsibility of the author of such custom automation to own the life cycle of any gem being referenced in those methods.

$evm.root

When an Automate method is launched, it has one global variable: **`$evm`**. The **`$evm`** variable allows the method to communicate back to Infrastructure Automation. The **`$evm.root`** is the root object in the workspace, it provides access to the data currently loaded in the Infrastructure Automation model. It uses the object's data to solve more complex problems by integrating with Infrastructure Automation methods.

The following is an excerpt from the **InspectMe** method that can be found in the **ManageIQ\System\Request** namespace. The dumpRoot method accesses the **`$evm.root`** object, and sends all of its attributes to the Infrastructure Automation Automate log for review. In the dumpServer Method, the inspect method is run based on the value of the miq_server obtained from the **`$evm.root`** object.

```
#########################
  #
  # Method: dumpRoot
  # Description: Dump Root information
  #
  #########################
  def dumpRoot
    $evm.log("info", "#{@log_prefix} - Root:<$evm.root> Begin Attributes")
    $evm.root.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - Root:<$evm.root> Attributes - #{k}: #{v}")}
    $evm.log("info", "#{@log_prefix} - Root:<$evm.root> End Attributes")
    $evm.log("info", "")
  end

  #########################
  #
  # Method: dumpServer
  # Inputs: $evm.root['miq_server']
  # Description: Dump MIQ Server information
  #
  #########################
  def dumpServer
    $evm.log("info","#{@log_prefix} - Server:<#{$evm.root['miq_server'].name}> Begin Attributes")
    $evm.root['miq_server'].attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - Server:<#
{$evm.root['miq_server'].name}> Attributes - #{k}: #{v.inspect}")}
    $evm.log("info","#{@log_prefix} - Server:<#{$evm.root['miq_server'].name}> End Attributes")
    $evm.log("info", "")
  end
```

The result of dumpRoot is below. The value of miq_server is what gets passed into the dumpServer method.

```
[----] I, [2012-10-23T13:53:54.517279 #5320:f329024]  INFO -- : <User-Defined Method> [InspectMe] - EVM Automate Method
Started
[----] I, [2012-10-23T13:53:54.523637 #5320:f329024]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root> Begin
Attributes
[----] I, [2012-10-23T13:53:54.527552 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - miq_server: #<MiqAeMethodService::MiqAeServiceMiqServer:0x0000001e76d900>
[----] I, [2012-10-23T13:53:54.528801 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - miq_server_id: 1
[----] I, [2012-10-23T13:53:54.529961 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - object_name: Request
[----] I, [2012-10-23T13:53:54.531067 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - request: inspectme
[----] I, [2012-10-23T13:53:54.534054 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - vm: DEV-JaneM
[----] I, [2012-10-23T13:53:54.535156 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - vm_id: 85
[----] I, [2012-10-23T13:53:54.536238 #5320:ef8c538]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root>
Attributes - vmdb_object_type: vm
[----] I, [2012-10-23T13:53:54.537159 #5320:f329024]  INFO -- : <User-Defined Method> [InspectMe] - Root:<$evm.root> End
Attributes
[----] I, [2012-10-23T13:53:54.537772 #5320:f329024]  INFO -- : <User-Defined Method>
```

Method Hierarchy

The Automate Model inline methods have a hierarchy. The sublevels in the hierarchy have access to the methods for itself and the levels above it. For example, Red Hat Hosts have access to the Red Hat Host methods, Host Methods, and Base Methods.

The following nested list displays the hierarchy.

- Top Level: Base

- Authentication (authentication)
  - Private Keys (auth_private_key)
    - Key Pair for Clouds (auth_key_pair_cloud)
      - Amazon (auth_key_pair_amazon)
      - OpenStack (auth_key_pair_openstack)
- Availability Zones (availability_zone)
  - Amazon (availability_zone_amazon)
  - OpenStack (availability_zone_openstack)
- Classification (classification)
- Cloud Networks (cloud_network)
- Cloud Resource Quotas (cloud_resource_quota)
  - OpenStack (openstack_resource_quota)
- Cloud Subnets (cloud_subnet)
- Cloud Tenants (cloud_tenant)
- Customization Templates (customization_template)
  - Cloud Init (customization_template_cloud_init)
  - Kickstart (customization_template_kickstart)
  - Sysprep (customization_template_sysprep)
- Cluster (ems_cluster)
- Event (ems_event)
- Folder (ems_folder)
- Providers (ext_management_system)
  - Cloud (ems_cloud)
    - Amazon (ems_amazon)
    - Openstack (ems_openstack)
  - Infrastructure (ems_infra)
    - Microsoft System Center VMM (ems_microsoft)
    - Red Hat Enterprise Virtualization (ems_redhat)
    - VMware vCenter (ems_vmware)
- Filesystems (filesystem)
- Firewall Rules (filewall_rule)
- Flavors
  - Amazon (flavor_amazon)
  - OpenStack (flavor_openstack)
- Floating IPs (floating_ip)
  - Amazon (floating_ip_amazon)
  - OpenStack (floating_ip_openstack)
- Guest Applications (guest_application)
- Guest Devices (guest_device)
- Hardware (hardware)
- Hosts (host)
  - Red Hat Enterprise Virtualization (host_redhat)
  - VMware (host_vmware)
    - VMware ESX (host_vmware_esx)
- ISO Images (iso_image)

- Jobs (job)
- LANs (lan)
- Groups (miq_group)
- Policies (miq_policy)
- Proxies (miq_proxy)
- Requests (miq_request)
  - Automation (automation_request)
  - Host Provisioning (miq_host_provision_request)
  - VM Provisioning (miq_provision_request)
    - VM Templates (miq_provision_request_template)
  - Service Reconfiguration (service_reconfigure_request)
  - Service Template Provisioning (service_template_provision_request)
  - VM Migration (vm_migrate_request)
  - VM Reconfiguration (vm_reconfigure_request)
- Request Task (miq_request_task)
  - Automation (automation_task)
  - Host Provisioning (miq_host_provision)
  - VM Provisioning (miq_provision)
    - Cloud (miq_provision_cloud)
      - Amazon (miq_provision_amazon)
      - OpenStack (miq_provision_openstack)
    - Red Hat Enterprise Virtualization (miq_provision_redhat)
      - Via ISO (miq_provision_redhat_via_iso)
      - Via PXE (miq_provision_redhat_via_pxe)
    - VMware (miq_provision_vmware)
      - Via NetApp RCU (miq_provision_vmware_via_net_app_rcu)
      - Via PXE (miq_provision_vmware_via_pxe)
  - Service Reconfiguration (service_reconfigure_task)
  - Service Template Provisioning (service_template_provision_task)
  - VM Migratation (vm_migrate_task)
  - VM Reconfiguration (vm_reconfigure_task)
- Servers (miq_server)
- Networks (network)
- Operating Systems (operating_system)
- PXE Images (pxe_image)
  - iPXE (pxe_image_ipxe)
  - PXELINUX (pxe_image_pxelinux)
- PXE Servers (pxe_server)
- Resource Pools (resource_pool)
- Security Groups (security_group)
  - Amazon (security_group_amazon)
  - OpenStack (security_group_openstack)
- Services (service)
- Service Resources (service_resource)

- Service Templates (service_template)
- Snapshots (snapshot)
- Storages (storage)
- Switches (switch)
- Users (user)
- VMs or Templates (vm_or_template)
  - Templates (miq_template)
    - Cloud (template_cloud)
      - Amazon (template_amazon)
      - OpenStack (template_openstack)
    - Infrastructure (template_infra)
      - Microsoft (template_microsoft)
      - Red Hat Enterprise Virtualization (template_redhat)
      - VMware (template_vmware)
  - VMs (vm)
    - Clouds (vm_cloud)
      - Amazon (vm_amazon)
      - OpenStack (vm_openstack)
    - Infrastructure (vm_infra)
      - Microsoft (vm_microsoft)
      - Red Hat Enterprise Virtualization (vm_redhat)
      - Vmware (vm_vmware)
- Windows Images (windows_images)

Base Methods

These methods may be used with all objects available in the Automate Model.

Method	Usage
inspect	Returns a string containing a list of attributes of the object. See the InspectMe method in Samples class
inspect_all	Returns all information for an object
virtual_column_names	Returns the objects virtual columns names
virtual_columns_inspect	Returns the objects virtual columns and values
reload	Returns to original object to prevent the internal object from being returned
model_suffix	Returns objects suffix. For an object of type MiqAeServiceVmVmware, returns “Vmware”
tagged_with?(category, name)	Is the object tagged with the category and name specified?
tags(category = nil)– this means that category is an optional parameter, with a default of nil	Returns the tags.
tag_assign(tag)	Assigns tag to the object, except for the miq_provision object, which uses add_tag(category, tag_name)
tag_unassign(tag)	Unassigns tag to the object, except for the miq_provision object, which uses clear_tag(category, tag_name)

The **InspectMe Sample Method** uses many of the Methods shown in this document. The method returns attributes of the Infrastructure Automation Server and then returns attributes for the host, cluster, and virtual machine from the provider of invocation. In many environments it is linked to a button.

```
###################################
# EVM Automate Method: InspectMe
#
# Notes: Dump the objects in storage to the automation.log
#
###################################
begin
  @method = 'InspectMe'
  @log_prefix = "[#{@method}]"
  $evm.log("info", "#{@log_prefix} - EVM Automate Method Started")

  # Turn on verbose logging
  @debug = true
```

```
  # List the types of object we will try to detect
  obj_types = %w{ vm host storage ems_cluster ext_management_system }
  obj_type = $evm.root.attributes.detect { |k,v| obj_types.include?(k)}

  # uncomment below to dump root object attributes
  dumpRoot

  # uncomment below to dump miq_server object attributes
  dumpServer


  # If obj_type is NOT nil
  unless obj_type.nil?
    rootobj = obj_type.first
    obj = obj_type.second
    $evm.log("info", "#{@log_prefix} - Detected Object:<#{rootobj}>")
    $evm.log("info","")

    case rootobj
    when 'host' then dumpHost(obj)
    when 'vm' then dumpVM(obj)
    when 'ems_cluster' then dumpCluster(obj)
    when 'ext_management_system' then dumpEMS(obj)
    when 'storage' then dumpStorage(obj)
    end
  end

  #
  # Exit method
  #
  $evm.log("info", "#{@log_prefix} - EVM Automate Method Ended")
  exit MIQ_OK

  #
  # Set Ruby rescue behavior
  #
rescue => err
  $evm.log("error", "#{@log_prefix} - [#{err}]\n#{err.backtrace.join("\n")}")
  exit MIQ_ABORT
end
```

Availability Zones (availability_zone)

Method	Use
ext_management_system	Returns object's Management System
vms	Returns object's VMs
vms_and_templates	Returns object's VMs and templates
cloud_subnets	Returns object's cloud subnets

Classifications (classification)

Method	Use
parent	Returns object's parent object
namespace	Returns object's namespace
category	Returns object's category
name	Returns object's name
to_tag	Returns object's tag mapping

Cloud Networks (cloud_network)

Method	Use
ext_management_system	Returns object's Management System
cloud_tenant	Returns object's cloud tenant
cloud_subnets	Returns object's cloud subnets
security_groups	Returns object's security groups
vms	Returns object's VMs

Cloud Resource Quotas (cloud_resource_quota)

Method	Use
ext_management_system	Returns object's Management System
cloud_tenant	Returns object's cloud tenant

Cloud Subnets (cloud_subnet)

Method	Use
cloud_network	Returns object's cloud network
availability_zone	Returns object's availability zone
vms	Returns object's VMs

Cloud Tenants (cloud_tenant)

Method	Use
ext_management_system	Returns object's Management System
security_groups	Returns object's security groups
cloud_networks	Returns object's cloud network
vms	Returns object's VMs
vms_and_templates	Returns object's VMs and templates
miq_templates	Returns object's templates
floating_ips	Returns object's floating IP addresses
cloud_resource_quotas	Returns object's quotas

Customization Template (customization_template)

Method	Use
Pxe_images	Returns customization templates pxe images

Clusters (ems_cluster)

Method	Use
all_resource_pools	Return all of the objects Resource Pools
all_vms	Return all of the objects Virtual Machines
default_resource_pool	Return the objects default Resource Pool
ems_events	Returns an array of EmsEvent records associated with the object
ext_management_system	Return objects Management System
hosts	Return objects Hosts
parent_folder	Return objects Parent Folder
register_host(host)	Register Host to this Cluster
resource_pools	Return objects Resource Pools
storages	Return objects datastores
vms	Return objects Virtual Machines

```
  ##########################
  #
  # Method: dumpCluster
  # Inputs: $evm.root['ems_cluster']
  # Description: Dump Cluster information
  #
  ##########################
  def dumpCluster(cluster)
    $evm.log("info","#{@log_prefix} - Cluster:<#{cluster.name}> Begin Attributes")
    cluster.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - Cluster:<#{cluster.name}> Attributes - #{k}:
#{v.inspect}")}
    $evm.log("info","#{@log_prefix} - Cluster:<#{cluster.name}> End Attributes")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Cluster:<#{cluster.name}> Begin Associations")
    cluster.associations.sort.each { |assc| $evm.log("info", "#{@log_prefix} - Cluster:<#{cluster.name}> Associations - #
{assc}")}
    $evm.log("info","#{@log_prefix} - Cluster:<#{cluster.name}> End Associations")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Cluster:<#{cluster.name}> Begin Virtual Columns")
    cluster.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@log_prefix} - Cluster:<#{cluster.name}> Virtual
Columns - #{vcn}: #{cluster.send(vcn)}")}
    $evm.log("info","#{@log_prefix} - Cluster:<#{cluster.name}> End Virtual Columns")
    $evm.log("info","")
  end
```

Management System Events (ems_event)

Method	Use
ext_management_system	Returns object's provider
ems	Shortcut to ext_management_system
src_vm	Source VM for the event
vm	VM for the event
src_host	Source Host for the event
host	Host for the event
dest_vm	Destination VM for the event
service	Service for the event
dest_host	Destination Host for the event
refresh(*targets)	Refresh the target types specified (ems, vm, host, src_vm, src_host, dest_vm, or dest_host)

Management System Folders (ems_folder)

Method	Use

Method	Use
hosts	Returns hosts that are in the folder
vms	Returns VMs that are in folder
register_host(host)	Registers specified host to the folder
folder_path(*options)	Returns folders path

Providers (ext_management_system)

Method	Use
authentication_password_encrypted	Returns credentials password encrypted
authentication_password	Returns credentials password unencrypted
authentication_userid	Returns credentials user id
ems_clusters	Returns objects clusters
ems_events	Returns an array of EmsEvent records associated with the object
ems_folders	Returns objects folders
hosts	Returns objects hosts
refresh	Refreshes relationships and power states for objects related to the object
resource_pools	Returns objects resource pools
storages	Returns objects storages
vms	Returns objects vms
to_s	Converts object to string

```
  ##########################
  #
  # Method: dumpEMS
  # Inputs: $evm.root['ext_management_system']
  # Description: Dump EMS information
  #
  ##########################
  def dumpEMS(ems)
    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> Begin Attributes")
    ems.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - EMS:<#{ems.name}> Attributes - #{k}: #
{v.inspect}")}
    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> End Attributes")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> Begin Associations")
    ems.associations.sort.each { |assc| $evm.log("info", "#{@log_prefix} - EMS:<#{ems.name}> Associations - #{assc}")}
    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> End Associations")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> Begin EMS Folders")
    ems.ems_folders.each { |ef| ef.attributes.sort.each { |k,v| $evm.log("info", "#{@log_prefix} - EMS:<#{ems.name}> EMS
Folder:<#{ef.name}> #{k}: #{v.inspect}")}}
    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> End EMS Folders")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> Begin Virtual Columns")
    ems.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@log_prefix} - EMS:<#{ems.name}> Virtual Columns - #
{vcn}: #{ems.send(vcn)}")}
    $evm.log("info","#{@log_prefix} - EMS:<#{ems.name}> End Virtual Columns")
    $evm.log("info","")
  end
```

Cloud Providers (ems_cloud)

Method	Use
availability_zones	Return the provider's availability zones
cloud_networks	Return the provider's available networks
cloud_tenants	Return the provider's available tenants
flavors	Return the provider's hardware flavors
floating_ips	Return the provider's floating IP addresses
key_pairs	Return the provider's key pairs
security_groups	Return the provider's security groups
cloud_resource_quotas	Return the provider's resource quotas

Firewall Rules (firewall_rule)

Method	Use
resource	Return object's resource
source_security_group	Return object's source security group

Flavors (flavor)

Method	Use
ext_management_system	Returns object's Management System
vms	Returns object's VMs

Floating IPs (floating_ip)

Method	Use
ext_management_system	Returns object's Management System
vm	Returns object's VMs
cloud_tenant	Returns object's cloud tenant

Guest Applications (guest_application)

Method	Use
vm	Returns objects VM
host	Returns objects Host

Guest Device (guest_device)

Method	Use
hardware	Returns objects hardware
switch	Returns objects switch
lan	Returns objects LAN
network	Returns objects network

Hardware (hardware)

Method	Use
ipaddresses	Returns objects IP addresses
guest_devices	Returns objects guest devices
storage_adapters	Returns objects storage adapters
nics	Returns objects nics
ports	Returns objects ports
vm	Returns objects Virtual Machine
host	Returns objects Host
mac_addresses	Returns objects MAC addresses

Hosts (host)

Method	Use
authentication_password	Returns credential password
authentication_userid	Returns credential user
datacenter	Returns datacenter
directories	Returns list of directories for the object
domain	Returns the domain portion of the hostname
ems_cluster	Returns cluster
ems_events	Returns an array of EmsEvent records associated with the object
ems_folder	Returns hosts folder on Management System
event_log_threshold? (options)	Searches event log records to determine if an event has occurred x number of times within a defined time frame. Returns true if the number of matching records found are greater or equal to the specified freq_threshold, otherwise it returns false Options values: :message_filter_type - Must be one of "STARTS WITH", "ENDS WITH", "INCLUDES", "REGULAR EXPRESSION" :message_filter_value - <string value to search for> :time_threshold - Options time interval to search. Example: 2.days (Search the past 2 days of event logs) Default 10.days :freq_threshold - Number of occurrences to check for. Default = 2 :source, :event_id, :level, :name - Options filter values
ext_management_system	Returns Management System
files	Returns list of files for the object
guest_applications	Returns Guest Applications
hardware	Returns hardware
lans	Returns LANs
operating_system	Returns Operating System
storages	Returns datastores
switches	Returns network switches
vms	Returns VMs.

Method	Use
credentials(type = :remote)	Returns credentials for a Host for the specified type as an array for username/pwd. (Default type is :remote if no type is specified.) Supports 4 different types of credentials: :default = Default :remote = Remote Login (think SSH for ESX) :ws = Web Services :ipmi = IPMI Example 1: cred = host.credentials cred ⇒ ["user", "pwd"] Example 2: user_str, pwd_str = host.credentials(:ipmi) user_str ⇒ "user" pwd_str ⇒ "pwd"
ems_custom_keys	Returns Management Systems custom keys
ems_custom_get(key)	Gets Value for specified Management Systems custom key
ems_custom_set(attribute, value)	Sets value for specified custom key of the Management System
custom_keys	Lists Infrastructure Automation Server custom keys
custom_get(key)	Gets value for specified Infrastructure Automation Server custom key
custom_set(key, value)	Sets value for specified Infrastructure Automation Server custom key
ssh_exec(script)	Runs the specified script on the host
get_realtime_metric(metric, range, function)	Returns specified realtime metric
current_memory_usage	Returns current memory usage
current_cpu_usage	Returns current cpu usage
current_memory_headroom	Returns current memory headroom
to_s	Converts object to string
scan	Performs SmartState Analysis on the object

The following table lists the metric types available for the **`get_realtime_metric(metric, range, function)`** method for hosts.

Metric	Description
v_derived_storage_used	Capacity - Used space in bytes
v_pct_cpu_ready_delta_summation	CPU - Percentage ready
v_pct_cpu_wait_delta_summation	CPU - Percentage wait
v_pct_cpu_used_delta_summation	CPU - Percentage used
v_derived_host_count	State - Number of hosts (Hourly Count / Daily Average)
v_derived_cpu_reserved_pct	CPU - Percentage available
v_derived_memory_reserved_pct	Memory - Percentage available

The following Ruby snippet demonstrates using the **`get_realtime_metric(metric, range, function)`** method using the **`v_pct_cpu_ready_delta_summation`** metric.

```
host = $evm.root['host']
cpu_rdy = host.get_realtime_metric(:v_pct_cpu_ready_delta_summation, [15.minutes.ago.utc,5.minutes.ago.utc], :avg)

#########################
  #
  # Method: dumpHost
  # Inputs: $evm.root['host']
  # Description: Dump Host information
  #
  #########################
  def dumpHost(host)
    host = $evm.object['host'] || $evm.root['host']
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Attributes")
    host.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - Host:<#{host.name}> Attributes - #{k}: #
{v.inspect}")}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Attributes")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Associations")
    host.associations.sort.each { |assc| $evm.log("info", "#{@log_prefix} - Host:<#{host.name}> Associations - #{assc}")}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Associations")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Hardware")
    host.hardware.attributes.each { |k,v| $evm.log("info", "#{@log_prefix} - Host:<#{host.name}> Hardware - #{k}: #
```

```
{v.inspect}")}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Hardware")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Lans")
    host.lans.each { |lan| lan.attributes.sort.each { |k,v| $evm.log("info", "#{@log_prefix} - Host:<#{host.name}> Lan:<#
{lan.name}> - #{k}: #{v.inspect}")}}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Lans")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Switches")
    host.switches.each { |switch| switch.attributes.sort.each { |k,v| $evm.log("info", "#{@log_prefix} - Host:<#
{host.name}> Swtiche:<#{switch.name}> - #{k}: #{v.inspect}")}}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Switches")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Operating System")
    host.operating_system.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - Host:<#{host.name}> Operating
System - #{k}: #{v.inspect}")}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Operating System")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Guest Applications")
    host.guest_applications.each { |guest_app| guest_app.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} -
Host:<#{host.name}> Guest Application:<#{guest_app.name}> - #{k}: #{v.inspect}")}}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Guest Applications")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> Begin Virtual Columns")
    host.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@log_prefix} - Host:<#{host.name}> Virtual Columns - #
{vcn}: #{host.send(vcn).inspect}")}
    $evm.log("info","#{@log_prefix} - Host:<#{host.name}> End Virtual Columns")
    $evm.log("info", "")
  end
```

Hosts: Vmware ESX (host_vmware_esx)

Method	Use
disable_vmotion(device = nil)	Disable vMotion
enable_vmotion(device = nil)	Enable vMotion
enter_maintenance_mode(timeout = 0, evacuate = false)	Put Host in Maintenance Mode
exit_maintenance_mode(timeout = 0)	Leave Maintenance Mode
in_maintenance_mode?	Check to see if the host is in Maintenance Mode
power_down_to_standby(timeout = 0, evacuate = false)	Put Host in standby
power_up_from_standby(timeout = 0)	Take Host out of standby
reboot(force = false)	Reboot Host
shutdown(force = false)	Shutdown Host
vmotion_enabled?(device = nil)	Check to see if vMotion is enabled

LAN (lan)

Method	Use
switch	Returns objects switch
guest_devices	Returns objects guest devices
vms	Returns objects Virtual Machines
templates	Returns objects templates
hosts	Returns objects Hosts

Groups (miq_group)

Method	Use
users	Returns users in the current miq_group
vms	Returns Virtual Machines that this group owns
custom_keys	Returns all custom keys for the group
custom_get(key)	Returns the value of the specified custom key for the group
custom_set(key, value)	Sets the value for the specified key

```
  ##########################
  #
  # Method: dumpGroup
  # Inputs: $evm.root['user'].miq_group
  # Description: Dump User's Group information
  #
  ##########################
  def dumpGroup
    user = $evm.root['user']
    unless user.nil?
      miq_group = user.miq_group
      unless miq_group.nil?
        $evm.log("info","#{@method} - Group:<#{miq_group.description}> Begin Attributes [miq_group.attributes]")
        miq_group.attributes.sort.each { |k, v| $evm.log("info", "#{@method} - Group:<#{miq_group.description}> Attributes
```

```
- #{k}: #{v.inspect}")} unless $evm.root['user'].miq_group.nil?
        $evm.log("info","#{@method} - Group:<#{miq_group.description}> End Attributes [miq_group.attributes]")
        $evm.log("info", "")

        $evm.log("info","#{@method} - Group:<#{miq_group.description}> Begin Associations [miq_group.associations]")
        miq_group.associations.sort.each { |assc| $evm.log("info", "#{@method} - Group:<#{miq_group.description}>
Associations - #{assc}")}
        $evm.log("info","#{@method} - Group:<#{miq_group.description}> End Associations [miq_group.associations]")
        $evm.log("info","")

        $evm.log("info","#{@method} - Group:<#{miq_group.description}> Begin Virtual Columns
[miq_group.virtual_column_names]")
        miq_group.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@method} - Group:<#{miq_group.description}>
Virtual Columns - #{vcn}: #{miq_group.send(vcn).inspect}")}
        $evm.log("info","#{@method} - Group:<#{miq_group.description}> End Virtual Columns
[miq_group.virtual_column_names]")
        $evm.log("info","")
      end
    end
  end
```

Request (miq_request)

Request objects are submitted to Infrastructure Automation Server for processing. After the request phase, the request becomes a task object. The table below shows the relationship between a request object and a task object.

Request Object	Task Object
automation_request	automation_task
miq_host_provision_request	miq_host_provision
miq_provision_request	miq_provision
vm_reconfigure_request	vm_reconfigure_task
service_template_provision_request	service_template_provision_task
vm_migrate_request	vm_migrate_task

If you set something on the request object, it will be inherited by the task instance that does the work. This may be useful if you are provisioning multiple virtual machines at a time and need to modify the same setting for all. Otherwise, the item can be modified on the individual task.

Method	Use
add_tag(category, tag_name)	Add a tag to a provision instance by specifying the category and tag name Example: miq_provision.add_tag(:location, "CHI")
approve(approver,reason for approval)	Approves the miq_request. Example: $evm.root["miq_request"].approve("admin", "Auto-Approved")
approvers	Returns request approvers.
authorized?	Returns true if authorized, false if not.
clear_tag(category=nil, tag_name=nil)	Without any parameters, the clear_tag method will clear all tags from the provision request. Providing a category will clear all tags selected in that category. Clear a specific category/tag by providing it. Example: miq_provision.clear_tag(:location, "CHI")
deny	Denies the miq_request Example: # Deny the request $evm.log('info',"Request denied because of Quota") $evm.root["miq_request"].deny("admin", "Quota Exceeded")
get_classification(category)	Works the same as get_tag(category) but the returned data is a hash with :name and :description Example: request.get_classification(:department) Returns: [{:name⇒"accounting", :description⇒"Accounting"}, {:name⇒"engineering", :description⇒"Engineering"}]
get_classifications	Works the same as get_tag but the returned tag data is a hash with :name and :description Example: request.get_classifications Returns: {:cc⇒{:name⇒"001", :description⇒"Cost Center 001"}, :department⇒[{:name⇒"accounting", :description⇒"Accounting"}, {:name⇒"engineering", :description⇒"Engineering"}]}
get_option(key)	Returns a value from the options hash based on the name of the key name passed in. Internally many of the values are stored as an array of items. (For example, a target host would be stored as the index to the host object in the db and the display name.) Calling this method will return the first item if it is an array. For non-array values the item is returned unmodified. Example: miq_provision_request.get_option(:number_of_cpus)

Method	Use
get_tag(category)	Returns the tags selected for the specified tag category. Example: request.get_tag(:department) Returns: ["accounting", "engineering"]
get_tags	Get all selected tags stored in a hash by category. If more than one tag is selected in a category, the hash will contain an array of tag names. Otherwise it will contain the tag name as a string. Example: request.get_tags Returns: {:cc⇒"001", :department⇒["accounting", "engineering"]}
miq_request	(Legacy support) Internal Note: The miq_request instance use to be a separate instance from the specific request instance (like miq_provision_request). When the classes were refactored into 1 this method was added to allow existing code and automate methods to continue to run unchanged.)
miq_request_tasks	Returns the requests tasks
options	Returns a hash containing all the options set for the current provision object
pending	Puts the object in a pending state for approval Example: # Raise automation event: request_pending $evm.root["miq_request"].pending
reason	Returns reason for approval or denial of request
requester	Returns the requester
resource	Returns the resource for the request
set_message(value)	Sets the message for the request
set_option(key, value)	Sets the specified key/value pair for the object

Automation Request (automation_request)

Method	Use
automation_tasks	Returns objects automate tasks

Host Provision Request (miq_host_provision_request)

Method	Use
miq_host_provisions	Returns the miq_host_provisions objects
ci_type	Returns the cloud infrastructure type: host

VM Provision Request (miq_provision_request)

Method	Use
check_quota(quota_type, options={})	Returns the quota information for the specified type
ci_type	Returns the cloud infrastructure type: vm
eligible_clusters	Returns an array of available Cluster objects filtered by previously selected resources
eligible_customization_templates	Returns an array of available Customization Templates filtered by previously selected resources
eligible_folders	Returns an array of available Folder objects filtered by previously selected resources
eligible_hosts	Returns an array of available Host objects filtered by previously selected resources
eligible_iso_images	Returns an array of available ISO image objects filtered by previously selected resources
eligible_pxe_images	Returns an array of available PXE Image objects filtered by previously selected resources
eligible_pxe_servers	Returns an array of available PXE Server objects filtered by previously selected resources
eligible_resource_pools	Returns an array of available Resource Pool objects filtered by previously selected resources
eligible_resources(rsc_type)	Returns eligible resources given the type specified
eligible_storages	Returns an array of available Storage (Datastore) objects filtered by previously selected resources
eligible_windows_images	Returns an array of available Windows Image objects filtered by previously selected resources

Method	Use
get_folder_paths	Returns a hash where the key is an index and the value is the fully-qualified path name of the folder. (Sample: {7 ⇒ Dev/Dept1/QA, 8 ⇒ Test/Dept2/QA}) This format is useful when a fully-qualified path is required to match the folder name. For example, if you had multiple QA folders under different departments in the sample above. To find the proper QA folder you need to evaluate the entire folder path.
get_retirement_days	Returns the number of dates until retirement
miq_provision	Returns the task.
miq_request	Returns the miq_provision_requests miq_request object
set_cluster(rsc)	Set the cluster to use based on object returned from eligible_clusters
set_customization_template(rsc)	Set the customization_template to use based on object returned from eligible_customization_templates
set_folder(folder_path)	Set the folder to use based on object returned from eligible_folders. In addition, set_folder accepts the following folder types: Folder Paths - separated by forward slashes. (Must include Data-center name.) Example: Prod/Discovered virtual machine where Prod is the Data-center name and Discovered virtual machine is the folder name. Object returned from the get_folder_paths method
set_host(rsc)	Set the host to use based on object returned from eligible_hosts
set_network_adapter(idx, nic_hash, value=nil)	Modifies the network card attached to the VM container Available settings: :is_dvs true / false (Default: false) :network (Network Name) :mac_address :devicetype (Default: VirtualPCNet32) Defined by VMware: http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/vim.vm.device.VirtualEthernetCard.html :connectable ⇒ {:allowguestcontrol ⇒ true / false} (Default: true) :connectable ⇒ {:startconnected ⇒ true / false} (Default: true) :connectable ⇒ {:connected ⇒ true / false} (Default: true) Example: prov.set_network_adapter(1, {:network ⇒ dvs_net1, :is_dvs ⇒ true})
set_network_address_mode(mode)	Sets IP address type. Available modes are dhcp and static
set_nic_settings(idx, nic_hash, value=nil)	Modifies the network interface settings at the operating system level Available settings: :addr_mode "dhcp" / "static" (Default: static) :ip_addr :subnet_mask :gateway :dns_domain :dns_servers (Windows Only) Comma separated values :sysprep_netbios_mode (Windows Only) Defined by VMware: http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/vim.vm.customization.IPSettings.NetBIOSMode.html :wins_servers Passed as a string specifying the Primary and Secondary WINS servers separated by a comma. "<PrimaryWINS>, <SecondaryWINS>" Example: prov.set_nic_settings(1, {:ip_addr⇒10.226.133.55, :subnet_mask⇒'255.255.255.192', :gateway⇒'10.226.133.5', :addr_mode⇒["static", "Static"] })
set_iso_image(rsc)	Set the iso_image to use based on object returned from eligible_iso_images
set_pxe_image(rsc)	Set the pxe_image to use based on object returned from eligible_pxe_images
set_pxe_server(rsc)	Set the pxe_server to use based on object returned from eligible_pxe_servers
set_resource_pool(rsc)	Set the resource_pool to use based on object returned from eligible_resource_pools
set_resource(rsc)	Sets the resource for the request. (Helper method, should not be called directly)
set_retirement_days	Set the number of days until retirement.

Method	Use
set_storage(rsc)	Set the Datastore (storage object) to use based on object returned from eligible_storages
set_vm_notes(note)	Sets text for the VM notes (aka annotation) field
set_windows_image(rsc)	Set the windows_image to use based on object returned from eligible_windows_images
source_type	Returns the provision source type. (values are 'vm' or 'template')
src_vm_id	Returns ID of the template being cloned
target_type	Returns the provision target type. (values are 'vm' or 'template')
vm_template	Returns the requests template

Request Task (miq_request_task)

Method	Use
add_tag(category, tag_name)	Add a tag to a provision instance by specifying the category and tag name. Example: miq_provision.add_tag(:location, "CHI")
clear_tag(category=nil, tag_name=nil)	Without any parameters, the clear_tag method will clear all tags from the provision request. Providing a category will clear all tags selected in that category. Clear a specific category/tag by providing it. Example: miq_provision.clear_tag(:location, "CHI")
destination	Returns the destination object. (The resultant object from running the task. In the case of provisioning, this would be the newly created VM.)
execute	Executes or processes the request.
finished(msg)	Sets the task to finished with the supplied message.
get_classification(category)	Works the same as get_tag(category) but the returned data is a hash with :name and :description. Example: request.get_classification(:department) Returns: [{:name⇒"accounting", :description⇒"Accounting"}, {:name⇒"engineering", :description⇒"Engineering"}]
get_classifications	Works the same as get_tag but the returned tag data is a hash with :name and :description. Example: request.get_classifications Returns: {:cc⇒{:name⇒"001", :description⇒"Cost Center 001"}, :department⇒[{:name⇒"accounting", :description⇒"Accounting"}, {:name⇒"engineering", :description⇒"Engineering"}]}
get_option_last(key)	This method is the same as get_option, except that it returns the last array value.
get_option(key)	Returns a value from the options hash based on the name of the key name passed in. Internally many of the values are stored as an array of items. (For example, a target host would be stored as the index to the host object in the db and the display name.) Calling this method will return the first item if it is an array. For non-array values the item is returned unmodified. Example: miq_provision_request.get_option(:number_of_cpus)
get_tag(category)	Returns the tags selected for the specified tag category. Example: request.get_tag(:department) Returns: ["accounting", "engineering"]
get_tags	Get all selected tags stored in a hash by category. If more than one tag is selected in a category, the hash will contain an array of tag names. Otherwise it will contain the tag name as a string. Example: request.get_tag Returns: {:cc⇒"001", :department⇒["accounting", "engineering"]}
message=(msg)	Sets the message for the request task.
miq_request	Returns the miq_request for the task.
miq_request_task	Returns the parent miq_request task.
miq_request.user_message=(msg)	Allows automate method to override the default request status message. As a result, you can set custom messages for the progression of the state machine and its success or failure. This message will appear as the last message for the service request.
miq_request_tasks	Returns the children miq_request tasks.
options	Returns a hash containing all the options set for the current object.
set_option(key, value)	Updates a key/value pair in the options hash for the provision object. Often the value is required to be an array.
source	Returns the source object. (The source, or input, object that the task runs against. In the case of provisioning, this would be the VM or template selected to be provisioned.)

Automation Task (automation_task)

Method	Use

Method	Use
automation_request	Returns associated automation_request object
status	Returns status of the task

Host Provision Task (miq_host_provision)

Method	Use
host	Returns objects host
miq_host_provision_request	Returns the request that created the task
status	Returns status of host provision

VM Provision Task (miq_provision)

Method	Use
check_quota(quota_type, options={})	Returns the quota information for the specified type
eligible_clusters	Returns an array of available Cluster objects filtered by previously selected resources
eligible_customization_templates	Returns an array of available Customization Templates
eligible_folders	Returns an array of available Folder objects filtered by previously selected resources
eligible_hosts	Returns an array of available Host objects filtered by previously selected resources
eligible_iso_images	Returns an array of available ISO Image objects filtered by previously selected resources
eligible_pxe_images	Returns an array of available PXE Image objects filtered by previously selected resources
eligible_pxe_servers	Returns an array of available PXE Servers filtered by previously selected resources
eligible_resource_pools	Returns an array of available Resource Pool objects filtered by previously selected resources
eligible_resources(rsc_type)	Returns the eligible resources for the resource type specified
eligible_storages	Returns an array of available Storage (Datastore) objects filtered by previously selected resources
eligible_windows_images	Returns an array of available Windows Image objects filtered by previously selected resources
get_domain_details	Returns domain information
get_domain_name	Returns domain name
get_folder_paths	Returns a hash where the key is an index and the value is the fully-qualified path name of the folder. (Sample: {7 ⇒ Dev/Dept1/QA, 8 ⇒ Test/Dept2/QA}) This format is useful when a fully-qualified path is required to match the folder name. For example, if you had multiple QA folders under different departments in the sample above. To find the proper QA folder you need to evaluate the entire folder path.
get_network_details	Returns network information
get_network_scope	Returns network scope
miq_provision_request	Returns the provision request object
set_cluster(rsc)	Set the cluster to use based on object returned from eligible_clusters
set_customization_spec(name=nil, override=false)	Sets the name of the custom spec to use as defined by its name in Virtual Center (Note: This is not supported when being passed by VMware.)
set_customization_template(rsc)	Set the customization_template to use based on object returned from eligible_customization_templates
set_dvs(portgroup, switch = portgroup)	Set the name of the Distributed Virtual Switch (portgroup). An options <switch> name can also be passed Example: miq_provision.set_dvs('default')
set_folder(folder_path)	Set the folder to use based on object returned from eligible_folders. In addition, set_folder accepts the following folder types: Folder Paths - separated by forward slashes. (Must include Data-center name.) Example: Prod/Discovered virtual machine where Prod is the Data-center name and Discovered virtual machine is the folder name. Object returned from the get_folder_paths method
set_host(rsc)	Set the host to use based on object returned from eligible_hosts

Method	Use
set_network_adapter(idx, nic_hash, value=nil)	Modifies the network card attached to the VM container Available settings: :is_dvs true / false (Default: false) :network (Network Name) :mac_address :devicetype (Default: VirtualPCNet32) Defined by VMware: http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/vim.vm.device.VirtualEthernetCard.html :connectable ⇒ {:allowguestcontrol ⇒ true / false} (Default: true) :connectable ⇒ {:startconnected ⇒ true / false} (Default: true) :connectable ⇒ {:connected ⇒ true / false} (Default: true) Example: prov.set_network_adapter(1, {:network ⇒ dvs_net1, :is_dvs ⇒ true})
set_network_address_mode(mode)	Available modes are dhcp and static
set_nic_settings(idx, nic_hash, value=nil)	Modifies the network interface settings at the operating system level Available settings: :addr_mode "dhcp" / "static" (Default: Statis) :ip_addr :subnet_mask :gateway :dns_domain :dns_servers (Windows Only) Comma separated values :sysprep_netbios_mode (Windows Only) Defined by VMware: http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/vim.vm.customization.IPSettings.NetBIOSMode.html :wins_servers Passed as a string specifying the Primary and Secondary WINS servers separated by a comma. "<PrimaryWINS>, <SecondaryWINS>" Example: prov.set_nic_settings(1, {:ip_addr⇒10.226.133.55, :subnet_mask⇒'255.255.255.192', :gateway⇒'10.226.133.5', :addr_mode⇒["static", "Static"] })
set_iso_image(rsc)	Set the iso_image to use based on object returned from eligible_iso_images
set_pxe_image(rsc)	Set the pxe_image to use based on object returned from eligible_pxe_images
set_pxe_server(rsc)	Set the pxe_server to use based on object returned from eligible_pxe_servers
set_resource_pool(rsc)	Set the resource_pool to use based on object returned from eligible_resource_pools
set_storage(rsc)	Set the Datastore (storage object) to use based on object returned from eligible_storages
set_vlan(vlan)	Sets the name of the VLan to use Example: miq_provision.set_vlan('default')
set_vm_notes(note)	Sets text for the VM notes (aka annotation) field
set_vm_notes(notes)	Sets text for the VM notes (aka annotation) field
set_windows_image(rsc)	Set the windows_image to use based on object returned from eligible_windows_images
source_type	Returns the provision source type. (values are 'vm' or 'template')
status	Returns provision status
target_type	Returns the provision target type. (values are 'vm' or 'template')
vdi_farm	Returns VDI Farm information
vm	The newly created vm
vm_template	Returns the template selected to be provisioned

```
begin
  miq_provision = $evm.root["miq_provision"] || $evm.root['miq_provision']
  prov = $evm.root["miq_provision"]
  user = prov.miq_request.requester
```

```
  raise "User not specified" if user.nil?


  ###################################
  # Process Change Request Number and set VM Annotation
  ###################################
  intake = prov.get_option(:vm_description)
  intake = "Change Request#: #{intake}"
  prov.set_option(:vm_description,intake)


  ###################################
  # Set the customization spec based on the environment tag chosen in the dialog
  ###################################
  tags = prov.get_tags
  $evm.log("info","Tags: #{tags.inspect}")
  env = tags[:environment]
  $evm.log("info", "Mapping custom spec based on Category Environment <#{env}> chosen in the dialog")
  if env.eql? "dev"
    customization_spec = "Dev-Specification"
    miq_provision.set_customization_spec(customization_spec)
  end
  if env.eql? "stg"
    customization_spec = "Stg-Specification"
    miq_provision.set_customization_spec(customization_spec)
  end


  ###################################
  # Set the VM Notes as follows:
  ###################################
  vm_notes = "#{intake}"
  vm_notes +=  "\nOwner: #{miq_provision.get_option(:owner_first_name)} #{miq_provision.get_option(:owner_last_name)}"
  vm_notes += "\nEmail: #{miq_provision.get_option(:owner_email)}"
  vm_notes += "\nSource Template: #{miq_provision.vm_template.name}"
  miq_provision.set_vm_notes(vm_notes)


  ###################################
  # Drop the VM in the targeted folder
  # In VC a folder must exist that matches the LDAP Group
  # VM will be placed in the Folder
  ###################################
  if prov.get_option(:placement_folder_name).nil?
    ###################################
    # If you want to use a Default folder, set folder = below to the default
    ###################################
    #    folder = "22F DC/LAB FARM/GSE/Intel/Infrastructure/ManageIQ/SelfServiceVMs"
    folder = "DC1/Infrastructure/ManageIQ/SelfService"
    $evm.log("info", "Placing VM in VC folder: <#{folder}")
    $evm.log("info", "Set_folder called with [#{folder.inspect}]")

    miq_provision.set_folder(folder)
  end


  ###################################################
  # Set the IP Address based on the :mac_address entered in the dialog
  #
  ###################################################
  ipaddr = prov.get_option(:mac_address)

  if ! ipaddr.nil?
    # Set provisioning options to override options
    prov.set_option(:sysprep_spec_override, [true, 1])
    prov.set_option(:addr_mode, ["static", "Static"])
    prov.set_option(:ip_addr, ipaddr)
    # Reset :mac_address to nil
    prov.set_option(:mac_address, nil)
  end

  $evm.log("info", "Provision Options: #{prov.options.inspect}")

  exit MIQ_OK

rescue => err
  $evm.log("info", "Set_folder err [#{err}]\n#{err.backtrace.join("\n")}")
end
```

Adding Disks to OpenStack Instance Provisioning

Using the clone_options, construct the keys and values required to define new disks as well as override or set other keys if needed.

Include the following settings with prov.set_options to set one or multiple volumes to attach. Note that each hash requires the :volume_id and :device_name keys.

```
clone_options = {
  :block_device_mapping => [
    {:volume_id => "d4xxxx92-5xx0-4xx9-bxx4-f1xxxxxxxx19", :device_name => "/dev/sdb"},
    {:volume_id => "e7xxxxf5-dxxd-4xx0-8xx3-a2xxxxxxxx12", :device_name => "/dev/sdc"},
```

```
  ]
}

prov = $evm.root["miq_provision"]
prov.set_option(:clone_options, clone_options)
```

Booting OpenStack Instances from Volume

You can boot instances created in Red Hat Enterprise Linux OpenStack Platform from a specified volume from an existing selection of block volumes. You can enable automate to pass the required parameters to OpenStack to use volumes as booting devices on an instance.

Include the following settings in the prov.set_option for provisioning instances created in OpenStack.

```
prov.set_option(
  :clone_options, {
    :image_ref => nil,
    :block_device_mapping_v2 => [{
      :boot_index => 0,
      :uuid => "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
      :device_name => "vda",
      :source_type => "volume",
      :destination_type => "volume",
      :delete_on_termination => false
    }]
  }
)
```

You can allow passing “block_device_mapping_v2” in the clone options and remove keys from clone_options by setting value to nil.

VM Provisioning for Clouds Task (miq_provision_cloud)

Method	Use
availability_zones	Returns object’s availability zones
instance_types	Returns object’s instance types
security_groups	Returns object’s security groups
floating_ip_addresses	Returns object’s floating IP addresses
cloud_networks	Returns object’s cloud network
cloud_subnets	Returns object’s cloud subnet
guest_access_key_pairs	Returns object’s guest key pair
cloud_tenants	Returns object’s cloud tenant

Service Template Provision Task (service_template_provision_task)

Method	Use
dialog_options	Returns objects dialog options hash
dialog_parser	Provides consolidated and configurable dialog parsing
get_dialog_option(key)	Returns objects dialog value for the specified key
service_resource	Returns the service resource for the task
set_dialog_option(key, value)	Sets a dialog option Example: set_dialog_option('memory',memory_size)
status	Returns the tasks status

Service Reconfiguration Task (service_reconfigure_task)

Method	Use
dialog_options	Show all dialog options for object
get_dialog_option(key)	Show a dialog option based stored in key
set_dialog_option(key, value)	Set value as a dialog option in key
status	Returns status of the task
finished(msg)	Sets the task to finished with the supplied message

VM Migrate Task (vm_migrate_task)

Method	Use
status	Returns status of the migration task

Proxies (miq_proxy)

Method	Use
host	Returns object’s hosts
powershell(script, returns = ‘string’)	Submits a powershell script

Servers (miq_server)

These methods are available to the Infrastructure Automation Server.

Method	Use
zone	Returns Infrastructure Automation Servers Zone
region_number	Returns Infrastructure Automation Servers Region Number
region_name	Returns Infrastructure Automation Servers Region Name

```
  ##########################
  #
  # Method: dumpServer
  # Inputs: $evm.root['miq_server']
  # Description: Dump MIQ Server information
  #
  ##########################
  def dumpServer
    $evm.log("info","#{@method} - Server:<#{$evm.root['miq_server'].name}> Begin Attributes")
    $evm.root['miq_server'].attributes.sort.each { |k, v| $evm.log("info", "#{@method} - Server:<#
{$evm.root['miq_server'].name}> Attributes - #{k}: #{v.inspect}")}
    $evm.log("info","#{@method} - Server:<#{$evm.root['miq_server'].name}> End Attributes")
    $evm.log("info", "")
  end
```

Network (network)

Method	Use
hardware	Returns objects hardware
guest_device	Returns objects guest devices

PXE Image (pxe_image)

Method	Use
customization_templates	Returns objects customization templates
pxe_server	Returns objects pxe server

PXE Server (pxe_server)

Method	Use
advertised_images	Returns objects advertised images
advertised_pxe_images	Returns objects advertised pxe images
default_pxe_image_for_windows	Returns objects default pxe image for windows
discovered_images	Returns objects discovered images
discovered_pxe_images	Returns objects discovered pxe images
images	Returns objects images
pxe_images	Returns objects pxe_images
windows_images	Returns objects windows images

Security Groups (security_group)

Method	Use
ext_management_system	Returns object's Management System
cloud_network	Returns object's cloud network
cloud_tenant	Returns object's cloud tenant
firewall_rules	Returns object's firewall rules
vms	Returns object's VMs

Service (service)

Method	Use
custom_keys	Returns custom keys
custom_get(key)	Gets value for specified custom key
custom_set(attribute, value)	Sets value for specified custom key
display=(display)	Set display option
group=(group)	Sets group that owns the service
name=(new_name)	Sets name of service
owner=(owner)	Sets owner of the service
request_retire	Retire Service immediately
retirement_warn=(seconds)	Sets when to send retirement warning
retires_on=(date)	Sets retirement date
shutdown_guest	Shuts downs guest operating system of the Service
start	Start the Service
stop	Stop the Service
suspend	Suspend the Service
vms	Show all virtual machines associated with this service
direct_vms	Show virtual machines directly associated with this service

Method	Use
indirect_vms	Show virtual machines associated with lower level services in the hierarchy
root_service	Show the top level service in the hierarchy for the target service
all_service_children	Show all lower level services to the target service in the hierarchy
direct_service_children	Show direct services associated with the target service
indirect_service_children	Show services associated with lower level services of the target service
parent_service	Show the parent service for the target service
description=(new_description)	Sets the service description
remove_from_vmdb	Delete the service from the database
dialog_options	Returns all dialog options
get_dialog_option(key)	Returns a specific dialog option specified by key
set_dialog_option(key, value)	Sets value of a dialog option specified by key

Service Resource (service_resource)

Method	Use
service	Returns the associated service
service_template	Returns the associated service template
resource	Returns the resource for the request
source	Returns the source object

Service Template (service_template)

Method	Use
group=(group)	Sets group for the service template
owner=(owner)	Sets owner for the service template

Snapshot (snapshot)

These methods can be used on Snapshots

Method	Use
vm	Returns Snapshots VM
current?	Checks to see if this is the current snapshot
get_current_snapshot	Returns the current snapshot id
revert_to	Reverts to specified snapshot
remove	Removes specified snapshot

Datastores (storage)

Method	Use
ext_management_systems,	Returns objects Management System
hosts	Returns objects Hosts
vms	Returns objects Virtual Machines
unregistered_vms	Returns objects unregistered Virtual Machines
to_s	Converts object to string
scan	Performs SmartState Analysis on the object

```
 ##########################
  #
  # Method: dumpStorage
  # Inputs: $evm.root['storage']
  # Description: Dump Storage information
  #
  ##########################
  def dumpStorage(storage)
    $evm.log("info","#{@log_prefix} - Storage:<#{storage.name}> Begin Attributes")
    storage.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - Storage:<#{storage.name}> Attributes - #{k}:
#{v.inspect}")}
    $evm.log("info","#{@log_prefix} - Storage:<#{storage.name}> End Attributes")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Storage:<#{storage.name}> Begin Associations")
    storage.associations.sort.each { |assc| $evm.log("info", "#{@log_prefix} - Storage:<#{storage.name}> Associations - #
{assc}")}
    $evm.log("info","#{@log_prefix} - Storage:<#{storage.name}> End Associations")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - Storage:<#{storage.name}> Begin Virtual Columns")
    storage.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@log_prefix} - Storage:<#{storage.name}> Virtual
Columns - #{vcn}: #{storage.send(vcn)}")}
    $evm.log("info","#{@log_prefix} - Storage:<#{storage.name}> End Virtual Columns")
    $evm.log("info","")
  end
```

Switch (switch)

These methods can be used on Switches.

Method	Use
host	Returns switch's Host
guest_devices	Returns switch's guest devices
lans	Returns switch's lans

User (user)

These methods can be used on the currently logged on user.

Method	Use
current_group	Returns user's assigned internal group
custom_get(key)	Returns the custom key value specified by "key"
custom_keys	Returns an array of custom keys
custom_set(key,value)	Sets custom value for "key" to "value"
email	Returns user's email address
get_ladap_attribute(name)	Returns the value of the specified LDAP attribute
get_ldap_atttribute_names	Returns user's LDAP attribute names
ldap_group	Returns user's assigned LDAP group
miq_requests	Returns user's requests
name	Returns user's name
userid	Returns user's userid
vms	Returns Virtual Machines that this user owns

```
  ##########################
  #
  # Method: dumpUser
  # Inputs: $evm.root['user']
  # Description: Dump User information
  #
  ##########################
  def dumpUser
    user = $evm.root['user']
    unless user.nil?
      $evm.log("info","#{@method} - User:<#{user.name}> Begin Attributes [user.attributes]")
      user.attributes.sort.each { |k, v| $evm.log("info", "#{@method} - User:<#{user.name}> Attributes - #{k}: #
{v.inspect}")}
      $evm.log("info","#{@method} - User:<#{user.name}> End Attributes [user.attributes]")
      $evm.log("info", "")

      $evm.log("info","#{@method} - User:<#{user.name}> Begin Associations [user.associations]")
      user.associations.sort.each { |assc| $evm.log("info", "#{@method} - User:<#{user.name}> Associations - #{assc}")}
      $evm.log("info","#{@method} - User:<#{user.name}> End Associations [user.associations]")
      $evm.log("info","")

      $evm.log("info","#{@method} - User:<#{user.name}> Begin Virtual Columns [user.virtual_column_names]")
      user.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@method} - User:<#{user.name}> Virtual Columns - #
{vcn}: #{user.send(vcn).inspect}")}
      $evm.log("info","#{@method} - User:<#{user.name}> End Virtual Columns [user.virtual_column_names]")
      $evm.log("info","")
    end
  end
```

Virtual Machines and Templates (vm_or_template)

The following methods can be used on a virtual machine or template object.

Method	Use
add_disk [1]	Add storage to an existing VM. This method accepts as parameters: disk_name, disk_size_mb, and options to pass values for :thinProvisioned, :dependent, and :persistent. Example: `vm.add_disk("[#{vm.storage_name}]", size_in_mb, thin_provisioned: true)`
changed_vm_value?	Checks the 2 most recent drift state captures , and answers whether the specified value changed between them
collect_running_processes	Collects the running processes of the object
create_snapshot(name, desc = nil)	Create a snapshot of the object. The **name** parameter is not used for Red Hat Virtualization virtual machines when creating a snapshot.
custom_get(key)	Get the value of specified Infrastructure Automation Server key from the object
custom_keys	List all Infrastructure Automation Server custom keys for the object
custom_set(key, value)	Set a custom Infrastructure Automation Server key value
datacenter	Returns objects Datacenter
direct_service	Show the direct service relationship of the virtual machine
directories	Returns number of directories on the object
ems_blue_folder (this will be reworked to be more VMware-specific)	Returns objects blue folder from VMware. These are the folders showing in VM and Templates view in VMware

Method	Use
ems_cluster	Returns objects cluster
ems_custom_get(key)	Gets specified key of custom Management System Attribute
ems_custom_keys	List the custom keys defined by the Management System for the object
ems_custom_set(attribute, value)	Sets specified key and value of custom Management System Attribute
ems_folder	Returns objects folder on Management System
ems_ref_string	Returns unique identifier the Management System uses to identify this resource. For example, in VMware a VM would return a value like: "vm-26622"
event_log_threshold? (options)	Searches event log records to determine if an event has occurred x number of times within a defined time frame. Returns true if the number of matching records found are greater or equal to the specified freq_threshold, otherwise it returns false Options values: :message_filter_type - Must be one of "STARTS WITH", "ENDS WITH", "INCLUDES", "REGULAR EXPRESSION" :message_filter_value - <string value to search for> :time_threshold - Options time interval to search. Example: 2.days (Search the past 2 days of event logs) Default 10.days :freq_threshold - Number of occurrences to check for. Default = 2 :source, :event_id, :level, :name - Options filter values
event_threshold?(options)	Checks if an event (or multiple events) have occurred X number of times in N seconds. The values below are used if no data is passed event_threshold?(options = {:time_threshold ⇒ 30.minutes, :event_types ⇒ ["MigrateVM_Task_Complete"], :freq_threshold ⇒ 2})
ext_management_system	Returns objects Management System
files	Returns number of files on the object
get_realtime_metric(metric, range, function)	Returns specified realtime metric
group=(group)	Sets objects group
guest_applications	Returns objects Guest Application list
hardware	Returns objects Hardware
host	Returns objects Host
migrate(host, pool = nil, priority = "defaultPriority", state = nil)	Migrates the object to another host. The only required parameter is host
miq_provision	If VM was created using Infrastructure Automation Server provisioning, this is the miq_provision task instance that created the VM
move_into_folder(folder) [2]	Use this method to move VMware virtual machines into folders. Example: vm = Vm.find_by_name('<vm_name>') folder = EmsFolder.find_by(:name ⇒ '<folder_name>') vm.move_into_folder(folder) unless folder.nil?
operating_system	Returns objects Operating System
owner	Return objects owner
owner=(owner)	Sets objects owner
performances_maintains_value_for_duration?	Based on options given, checks to see if a performance threshold is maintained for a time period Example: vm.performances_maintains_value_for_duration?(:column ⇒ "cpu_usage_rate_average", :operator ⇒ "=", :value ⇒ 3.51, :duration ⇒ 20.minutes)
reboot_guest	Reboots the guest operating system
reconfigured_hardware_value?	Checks if hardware value has been reconfigured
refresh	Refresh power states and relationships of the object
registered?	Is the object registered?
remove_all_snapshots	Remove all of the objects snapshots
remove_from_disk	Removes the object from disk
remove_from_vmdb	Removes the object from the VMDB
remove_snapshot(snapshot_id)	Remove a specific snapshot based on the snapshot_id
resource_pool	Returns objects Resource Pool
retire_now	Retire the object immediately
retirement_warn=(seconds)	Send a retirement warning
retires_on=(date)	Retire the object on date specified
revert_to_snapshot(snapshot_id)	Revert to a snapshot based on the snapshot_id
scan(scan_categories = nil)	Perform SmartState Analysis on the object. Scan_categories is optional
service	Show the top-level service for a virtual machine in a service hierarchy. For the immediate parent service relationship of a virtual machine, use direct_service
shutdown_guest	Shuts down the guest operating system of the object

Method	Use
snapshots	Returns list of snapshots for the object
standby_guest	Puts the operating system on standby
start	Starts the object See Samples/PowerOn_DHOB
stop	Stops the object
storage	Returns objects Datastore
suspend	Suspends the object
to_s	Converts object to string
unlink_storage	Removes the reference to the VM's Datastore
unregister	Unregisters the object from the Management System

```
  ##########################
  #
  # Method: dumpVM
  # Inputs: $evm.root['vm']
  # Description: Dump VM information
  #
  ##########################
  def dumpVM(vm)
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Attributes [vm.attributes]")
    vm.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Attributes - #{k}: #{v.inspect}")}
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Attributes [vm.attributes]")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Associations [vm.associations]")
    vm.associations.sort.each { |assc| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Associations - #{assc}")}
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Associations [vm.associations]")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Hardware Attributes [vm.hardware]")
    vm.hardware.attributes.each { |k,v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Hardware - #{k}: #
{v.inspect}")}
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Hardware Attributes [vm.hardware]")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Hardware Associations [vm.hardware.associations]")
    vm.hardware.associations.sort.each { |assc| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> hardware Associations -
#{assc}")}
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End hardware Associations [vm.hardware.associations]")
    $evm.log("info","")

    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Neworks [vm.hardware.nics]")
    vm.hardware.nics.each { |nic| nic.attributes.sort.each { |k,v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}>
VLAN:<#{nic.device_name}> - #{k}: #{v.inspect}")}}
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Networks [vm.hardware.nics]")
    $evm.log("info","")

    unless vm.ext_management_system.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin EMS [vm.ext_management_system]")
      vm.ext_management_system.attributes.sort.each { |ems_k, ems_v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}>
EMS:<#{vm.ext_management_system.name}> #{ems_k} - #{ems_v.inspect}")}
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End EMS [vm.ext_management_system]")
      $evm.log("info","")
    end

    unless vm.owner.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Owner [vm.owner]")
      vm.owner.attributes.each { |k,v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Owner - #{k}: #{v.inspect}")}
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Owner [vm.owner]")
      $evm.log("info","")
    end

    unless vm.operating_system.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Operating System [vm.operating_system]")
      vm.operating_system.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Operating
System - #{k}: #{v.inspect}")}
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Operating System [vm.operating_system]")
      $evm.log("info","")
    end

    unless vm.guest_applications.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Guest Applications")
      vm.guest_applications.each { |guest_app| guest_app.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} -
VM:<#{vm.name}> Guest Application:<#{guest_app.name}> - #{k}: #{v.inspect}")}} unless vm.guest_applications.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Guest Applications")
      $evm.log("info","")
    end

    unless vm.snapshots.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Snapshots")
      vm.snapshots.each { |ss| ss.attributes.sort.each { |k, v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}>
Snapshot:<#{ss.name}> - #{k}: #{v.inspect}")}} unless vm.snapshots.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Snapshots")
      $evm.log("info","")
    end
```

```
    unless vm.storage.nil?
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin VM Storage [vm.storage]")
      vm.storage.attributes.sort.each { |stor_k, stor_v| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Storage:<#
{vm.storage.name}> #{stor_k} - #{stor_v.inspect}")}
      $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End VM Storage [vm.storage]")
      $evm.log("info","")
    end

    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> Begin Virtual Columns -----")
    vm.virtual_column_names.sort.each { |vcn| $evm.log("info", "#{@log_prefix} - VM:<#{vm.name}> Virtual Columns - #{vcn}:
#{vm.send(vcn).inspect}")}
    $evm.log("info","#{@log_prefix} - VM:<#{vm.name}> End Virtual Columns -----")
    $evm.log("info","")
  end

  ####################
  #
  # Method: createSnapshot
  #
  ####################

  def createSnapshot(vm, snap_name, snap_desc=snap_name)
    $evm.log("info","#{@method} - VM:<#{vm.name}> Creating Snapshot:<#{snap_name}> Description:<#{snap_desc}>")
    vm.create_snapshot(snap_name, snap_desc)
  end

  ####################
  # Red Hat Virtualization Virtual Machines
  # Method: createSnapshot
  #
  ####################

    def createSnapshot(vm, snap_desc=snap_name)
        $evm.log("info","#{@method} - VM:<#{vm.name}> Creating Snapshot:<#{snap_name}> Description:<#{snap_desc}>")
        vm.create_snapshot(snap_desc)
    end
```

VMs (vm)

Method	Use
add_to_service(service)	Adds the VM to a service object
remove_from_service	Removes the VM from its parent service

The following table lists the metric types available for the **get_realtime_metric(metric, range, function)** method for virtual machines.

Metric	Description
v_derived_storage_used	Capacity - Used space in bytes
v_pct_cpu_ready_delta_summation	CPU - Percentage ready
v_pct_cpu_wait_delta_summation	CPU - Percentage wait
v_pct_cpu_used_delta_summation	CPU - Percentage used
v_derived_vm_count	State - Peak average virtual machines (Hourly Count / Daily Average)
v_derived_cpu_reserved_pct	CPU - Percentage available
v_derived_memory_reserved_pct	Memory - Percentage available

The following Ruby snippet demonstrates using the **get_realtime_metric(metric, range, function)** method using the **v_pct_cpu_ready_delta_summation** metric.

```
host = $evm.root['vm']
cpu_rdy = vm.get_realtime_metric(:v_pct_cpu_ready_delta_summation, [15.minutes.ago.utc,5.minutes.ago.utc], :avg)
```

VMs for Clouds (vm_cloud)

Method	Use
availability_zone	Returns object's availability zone
flavor	Returns object's favor
cloud_network	Returns object's cloud network
cloud_subnet	Returns object's cloud subnet
floating_ip	Returns object's floating IPs
security_groups	Returns object's security groups
key_pairs	Returns object's key pairs

Windows Image (windows_image)

Method	Use
customization_templates	Returns the images customization templates.
pxe_server	Returns the images pxe server.

Creating Categories and Tags

These methods are invoked using $evm.execute. See after the table for usage examples.

Method(parameter)	Use
category_exists?	Checks to see if a tag category exists. Usually used with unless or if to then create a category that does not exist. Example: $evm.execute('category_exists?', "department")
category_create	Creates a new category, and if only one tag value from this category can be assigned to a configuration item, sets that tag value using the **`single_value`** option. Example: $evm.execute('category_create', :name ⇒ "department", :single_value ⇒ false, :description ⇒ "Department") **NOTE:** The **`category_create`** method currently does not work as expected. The user interface shows the value provided for **`:description`** in the **Display Name** field and the value provided for **`:example_text`** is displayed in the **Description** field.
tag_exists?	Checks to see if a tag exists in a category. Example: $evm.execute('tag_exists?', "department", "finance")
tag_create	Creates a new tag in the specified category. Example: $evm.execute('tag_create', "department", :name ⇒ "finance", :description ⇒ "Finance")

Category and Tags Methods Example

In this example, the VMDB is checked to see if the **Department** category exists. If it does, then a message is logged. If not, the category is created and a message is logged. Values are then added to the category.

```
if $evm.execute('category_exists?', "department")
  $evm.log("info", "Classification department exists")
else
  $evm.log("info", "Classification department doesn't exist, creating category")
  $evm.execute('category_create', :name => "department", :single_value => false, :description => "Department")
  $evm.log("info", "Adding new tag in Department Category")
  $evm.execute('tag_create', "department", :name => "finance", :description => "Finance")
```

Quota

Method(parameter)	Use
vms_by_owner, vms_by_group, vms_by_owner_and_group	Collect stats about existing VMs by owner or in the same LDAP group as the owner in the current request. Sample return object: {:class_name⇒"Vm", :count⇒5, :ids⇒[22, 120, 121, 122, 117], :cpu⇒6, :memory⇒5120, :allocated_storage⇒29032972288, :used_storage⇒29032972288}
retired_vms_by_owner, retired_vms_by_group, retired_vms_by_owner_and_group	Collect stats about retired VMs, that have not been deleted from the host, by owner or in the same LDAP group as the owner in the current request. Sample return object: {:class_name⇒"Vm", :count⇒5, :ids⇒[22, 120, 121, 122, 117], :cpu⇒6, :memory⇒5120, :allocated_storage⇒29032972288, :used_storage⇒29032972288}
provisions_by_owner, provisions_by_group	Collect stats based on provisions running on the same day as the current request (based on scheduled date or immediate) by owner or in the same LDAP group as the owner. (Results do not include Provision Requests that have not been approved.) Sample return object: {:class_name⇒"MiqProvisionRequest", :count⇒6, :ids⇒[115, 116, 104, 102, 105, 112], :cpu⇒6, :memory⇒1536, , :storage⇒62914560}
requests_by_owner, requests_by_group	Collect stats based on provision requests made the same day as the request by the same owner or LDAP group as the owner in the current request, regardless of when the provision request is scheduled to run. (Results do not include Provision Requests that have been denied.) Sample return object: {:class_name⇒"MiqProvisionRequest", :count⇒6, :ids⇒[115, 116, 104, 102, 105, 112], :cpu⇒6, :memory⇒1536, , :storage⇒62914560}
active_provisions_by_owner, active_provisions_by_group, active_provisions	Collect stats based on currently active provision requests by the same owner or LDAP group as the owner in the current request. Sample return object: {:class_name⇒"MiqProvisionRequest", :count⇒6, :ids⇒[115, 116, 104, 102, 105, 112], :cpu⇒6, :memory⇒1536, , :storage⇒62914560}

1. Supported for Red Hat Virtualization and VMware virtual machines
2. Supported for VMware virtual machines only

Terminology

- Account Role A designation assigned to a user allowing or restricting a user to parts and functions of the Infrastructure Automation console.
- Action An execution that is performed after a condition is evaluated.
- Alert Infrastructure Automation alerts notify administrators and monitoring systems of critical configuration changes and threshold limits in the virtual environment. The notification can take the form of either an email or an SNMP trap.
- Analysis Profile A customized scan of hosts, virtual machines, or instances. You can collect information from categories, files, event logs, and registry entries.
- Cloud A pool of on-demand and highly available computing resources. The usage of these resources are scaled depending on the user requirements and metered for cost.
- Infrastructure Automation appliance A virtual machine on which the virtual management database (VMDB) and Infrastructure Automation server reside.
- Infrastructure Automation Console A web-based interface into the Infrastructure Automation appliance.
- Infrastructure Automation Role A designation assigned to a Infrastructure Automation server that defines what a Infrastructure Automation server can do.
- Infrastructure Automation Server The application that runs on the Infrastructure Automation appliance and communicates with the SmartProxy and the VMDB.
- Cluster Hosts that are grouped together to provide high availability and load balancing.
- Condition A test of criteria triggered by an event.
- Discovery Process run by the Infrastructure Automation server which finds virtual machine and cloud providers.
- Drift The comparison of a virtual machine, instance, host, cluster to itself at different points in time.
- Event A trigger to check a condition.
- Event Monitor Software on the Infrastructure Automation appliance which monitors external providers for events and sends them to the Infrastructure Automation server.
- Host A computer on which virtual machine monitor software is loaded.
- Instance/Cloud Instance A on-demand virtual machine based upon a predefined image and uses a scalable set of hardware resources such as CPU, memory, networking interfaces.
- Managed/Registered VM A virtual machine that is connected to a host and exists in the VMDB. Also, a template that is connected to a provider and exists in the VMDB. Note that templates cannot be connected to a host.
- Managed/Unregistered VM A virtual machine or template that resides on a repository or is no longer connected to a provider or host and exists in the VMDB. A virtual machine that was previously considered registered may become unregistered if the virtual machine was removed from provider inventory.
- Provider A computer on which software is loaded which manages multiple virtual machines that reside on multiple hosts.
- Policy A combination of an event, a condition, and an action used to manage a virtual machine.
- Policy Profile A set of policies.
- Refresh A process run by the Infrastructure Automation server which checks for relationships of the provider or host to other resources, such as storage locations, repositories, virtual machines, or instances. It also checks the power states of those resources.
- Regions Regions are used to create a central database for reporting and charting. Regions are used primarily to consolidate multiple VMDBs into one master VMDB for reporting.
- Resource A host, provider, instance, virtual machine, repository, or datastore.
- Resource Pool A group of virtual machines across which CPU and memory resources are allocated.
- Repository A place on a datastore resource which contains virtual machines.
- SmartProxy
    - The SmartProxy is a software agent that acts on behalf of the Infrastructure Automation appliance to perform actions on hosts, providers, storage and virtual machines.
    - The SmartProxy can be configured to reside on the Infrastructure Automation appliance or on an ESX server version.
    - The SmartProxy can be deployed from the Infrastructure Automation appliance, and provides visibility to the VMFS storage. Each storage location must have a SmartProxy with visibility to it. The SmartProxy acts on behalf of the Infrastructure Automation appliance.
    - If the SmartProxy is not embedded in the Infrastructure Automation server, it communicates with the Infrastructure Automation appliance over HTTPS on standard port 443.
- SmartState Analysis Process run by the SmartProxy which collects the details of a virtual machine or instance. Such details include accounts, drivers, network information, hardware, and security patches. This process is also run by the Infrastructure Automation server on hosts and clusters. The data is stored in the VMDB.
- SmartTags Descriptors that allow you to create a customized, searchable index for the resources in your clouds and infrastructure.
- Storage Location A device, such as a VMware datastore, where digital information resides that is connected to a resource.
- Tags Descriptive terms defined by a Infrastructure Automation user or the system used to categorize a resource.

- Template A template is a copy of a preconfigured virtual machine, designed to capture installed software and software configurations, as well as the hardware configuration, of the original virtual machine.
- Unmanaged Virtual Machine Files discovered on a datastore that do not have a virtual machine associated with them in the VMDB. These files may be registered to a provider that the Infrastructure Automation server does not have configuration information on. Possible causes may be that the provider has not been discovered or that the provider has been discovered, but no security credentials have been provided.
- Virtual Machine A software implementation of a system that functions similar to a physical machine. Virtual machines utilize the hardware infrastructure of a physical host, or a set of physical hosts, to provide a scalable and on-demand method of system provisioning.
- Virtual Management Database (VMDB) Database used by the Infrastructure Automation appliance to store information about your resources, users, and anything else required to manage your virtual enterprise.
- Virtual Thumbnail An icon divided into smaller areas that summarize the properties of a resource.
- Zones Infrastructure Automation Infrastructure can be organized into zones to configure failover and to isolate traffic. Zones can be created based on your environment. Zones can be based on geographic location, network location, or function. When first started, new servers are put into the default zone.

Managing Authentication for Infrastructure Management stand-alone deployment

You can find details about how to manage authentication of Infrastructure Management in stand-alone mode.

Important: These instructions do not apply to Infrastructure Management when installed with IBM Cloud Pak for AIOps. These instructions are only to be used when you are installing and running Infrastructure Management as a stand-alone instance.

- Active Directory
- 2-Factor-Authentication with IPA
- IPA/AD Trust Authentication
- LDAP
- SAML
- OpenID-Connect

- Active Directory
- Enabling Network Manager
- Discovering AD Domains
- Joining AD Domain
- Allowing AD Users to Login
- Configure SSSD
- Configure Apache
- Configure Administrative UI

Active Directory

In this guide we will cover how to manually configure a 7.1 Appliance's external authentication to work against Active Directory. This provides AD users access to the Appliance UI as well as the REST API.

1. Enabling Network Manager
2. Discovering AD Domain
3. Joining AD Domain
4. Allowing AD Users to Login to appliance
5. Configure SSSD
6. Configure Apache
7. Configure Administrative UI

In these examples, the AD Domain shown will be EXAMPLE.COM

Enabling Network Manager

By default, the appliance is configured with Network Manager enabled. Having Network Manager enabled allows the **realm** command to discover the available Active Directory domains that can be joined. If not enabled, one can still join an AD domain if known by the domain name.

If the appliance has Network Manager disabled, it can be enabled as follows:

```
# systemctl enable NetworkManager
# systemctl start NetworkManager
```

Discovering AD Domains

This only works if Network Manager is enabled.

```
# realm discover
example.com
  type: kerberos
  realm-name: EXAMPLE.COM
  domain-name: example.com
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common
  login-formats: %U@example.com
  login-policy: allow-realm-logins
```

Joining AD Domain

When joining an AD domain, specify a user that has enough permission to be able to browse the directory.

```
# realm join example.com -U user
Password for user: xxxxxxxx
#
```

Allowing AD Users to Login

```
# realm permit --all
```

Configure SSSD

Update the **/etc/sssd/sssd.conf** file as follows:

Note: Starting with SSSD version 1.15.2, which will be available in CentOS version 7.4, SSSD will provide the domain name as a user attribute. The below examples show how to set *ldap_user_extra_attrs* and *user_attributes* to take advantage of this new feature. If running an appliance built with CentOS version prior to CentOS 7.4 do not include *domainname* for these attributes.

```
    [domain/example.com]
    ad_domain = example.com
    krb5_realm = EXAMPLE.COM
    realmd_tags = manages-system joined-with-samba
    cache_credentials = True
    id_provider = ad
    krb5_store_password_if_offline = True
    default_shell = /bin/bash
    ldap_id_mapping = True
    use_fully_qualified_names = True
    fallback_homedir = /home/%d/%u
    access_provider = ad
=>  ad_gpo_map_permit = +httpd-auth
=>  ldap_user_extra_attrs = mail, givenname, sn, displayname, domainname

=>  [sssd]
=>  domains = example.com
=>  config_file_version = 2
=>  services = nss, pam, ifp
=>  default_domain_suffix = example.com

=>  [nss]
=>  homedir_substring = /home

=>  [pam]
=>  default_domain_suffix = example.com

=>  [ifp]
=>  default_domain_suffix = example.com
=>  allowed_uids = apache, root, manageiq
=>  user_attributes = +mail, +givenname, +sn, +displayname, +domainname
```

Configure Apache

Make sure the Kerberos keytab created by **realm join** above is readable by Apache.

```
# chgrp apache /etc/krb5.keytab
# chmod 640    /etc/krb5.keytab
```

Create the Apache configuration files

```
# TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"
# cp ${TEMPLATE_DIR}/etc/pam.d/httpd-auth                          \
                    /etc/pam.d/httpd-auth
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user.conf       \
                    /etc/httpd/conf.d/
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth.conf.erb \
                    /etc/httpd/conf.d/manageiq-external-auth.conf
```

Update the Apache configuration file **/etc/httpd/conf.d/manageiq-external-auth.conf** as follows to reference the Kerberos keytab appropriately.

```
...
    <Location /dashboard/kerberos_authenticate>
      AuthType            GSSAPI
      AuthName            "GSSAPI Single Sign On Login"
=>    GssapiCredStore     keytab:/etc/krb5.keytab
      GssapiLocalName     on
      Require             pam-account httpd-auth

      ErrorDocument 401 /proxy_pages/invalid_sso_credentials.js
    </Location>

...
```

Set appropriate SELinux permissions:

```
# setsebool -P allow_httpd_mod_auth_pam on
# setsebool -P httpd_dbus_sssd          on
```

Restart Services

```
# systemctl restart sssd
# systemctl restart httpd
```

Configure Administrative UI

Login as admin, then in *Configure→Configuration→Authentication*

- Set mode to External (httpd)
- Check: *Get User Groups from External Authentication (httpd)*
- Check: *Enable Single Signon* if you want to allow Kerberos SSO to AD.
- Click Save.

The above steps need to be done on each UI and WebService enabled appliance.

in *Configure→Configuration→Access Control*

- Make sure the user's AD group for the appliance are created and appropriate roles assigned to those groups.

- 2-Factor Authentication
- Requirements
- Configure Administrative UI
- Configure External Authentication
- Enabling 2-Factor Authentication
- Using 2-Factor Authentication

2-Factor Authentication

Enabling 2-Factor Authentication on the Appliance is currently supported with External Authentication to IPA.

In this guide we will cover how to manually configure a 7.1 Appliance's external authentication to work with 2-Factor Authentication with IPA. This provides IPA Users access to the Appliance Administrative UI and the REST API using their IPA Password followed by a One-Time-Password.

1. Requirements
2. Configuring the Administrative UI
3. Configuring External Authentication
4. Enabling 2-Factor Authentication
5. Using 2-Factor Authentication

Requirements

The following is needed in order to enable 2-Factor-Authentication to the Appliance:

- A CentOS/RHEL 7.1 based Appliance
- IPA Server based on FreeIPA 4.1.0 or later

Configure Administrative UI

Login as admin, then in *Configure→Configuration→Authentication*

- Set mode to External (httpd)

- Check: *Get User Groups from External Authentication (httpd)*
- Do Not Check: *Enable Single Signon*
  - **Note**: Kerberos SSO with OTP is not supported in the current release of FreeIPA 4.1.0 on CentOS/RHEL 7.1
- Click Save.

The above steps need to be done on each UI and WebService enabled appliance.

in *Configure→Configuration→Access Control*

- Make sure the user's groups are created on the Appliance and appropriate roles assigned to those groups.

Configure External Authentication

Using the Appliance Console, simply enable external authentication as you currently would to IPA.

1. Login to console as *root*
2. Run *appliance_console*
3. Summary screen should show **External Auth** as *not configured*, Press any key
4. From the Advanced Setting menu, select the menu item **Configure External Authentication (httpd)**
5. Enter the FQDN of the IPA Server, i.e. *ipaserver.test.company.com*
6. Enter the IPA Server domain, i.e. *test.company.com*
7. Enter the IPA Server realm, i.e. *TEST.COMPANY.COM*
8. Press enter to select the default IPA Server Principal, i.e. *admin*
9. Enter the Password of the IPA Server Principal
10. Review details, and Enter **y** to proceed.

Enabling 2-Factor Authentication

Enabling 2-Factor Authentication is done using the IPA administrative UI.

- The IPA administrator needs to change the User authentication types for the user from Password or Radius to *Two factor authentication (password + OTP)*
- The previous can be done either per user or for all users from the IPA Server global Configuration under User Options.
- The user logs into IPA, then from the OTP Tokens tab, add an OTP token - Time-based (TOTP) or Counter-based (HOTP).
- Specifying a Description for the OTP, then clicking add, brings up a QR Code which can then be scanned into an app like **FreeOTP** for example.
- Then from the main IPA login screen, the user can Synchronize the OTP Token by specifying their Username, password and then two OTP's generated from the **FreeOTP** app.
- After which, the user can login to FreeIPA using their Password+OTP.

Using 2-Factor Authentication

Once the above is done, the user now needs to specify their IPA Username and their Password followed by their One-Time-Password as generated by the **FreeOTP** app, when:

- Logging into the Appliance Administrative UI
- Accessing the REST API

- IPA/AD Trust Authentication
- Requirements
- Set up and configure an IPA server for AD Trust Authentication
- Enable the MiQ Appliance to use the configued IPA server
- Configure the MiQ appliance to use external authentication
- Create groups on the MiQ appliance
- Use AD Trust Authentication

IPA/AD Trust Authentication

Active Directory (AD) Trust Authentication on the Appliance is supported with External Authentication to IPA.

In this guide we will cover how to manually configure an Appliance's external authentication to work with AD Trust Authentication using IPA. This provides IPA Users access to the Appliance Administrative UI and the REST API using their AD credentials.

1. Requirements

2. Set up and configure an IPA server for AD Trust Authentication
3. Enable the MiQ Appliance to use the configued IPA server
4. Configure the MiQ appliance to use external authentication
5. Create groups on the MiQ appliance
6. Use AD Trust Authentication

Requirements

The following is needed in order to enable AD Trust Authentication to the Appliance:

- A CentOS/RHEL 7.2 based MiQ Appliance
- Windows Server 2008 R2 or later with configured AD DC and DNS installed locally on the DC

Set up and configure an IPA server for AD Trust Authentication

- Configure an IPA Server based on FreeIPA 3.3.3 or later

Instructions for setting up and configuring cross-realm trust between an IPA domain and an AD (Active Directory) domain can be found at freeipa.org Active Directory Trust Setup

- Add necessary user attributes to the SSSD configuration on the IPA server

The SSSD configuration file on the IPA Server must be updated to list needed user attributes.

Add the following entry to the SSSD configuration file /etc/sssd/sssd.conf

Note: Starting with SSSD version 1.15.2, which will be available in CentOS version 7.4, SSSD will provide the domain name as a user attribute. The below examples show how to set *ldap_user_extra_attrs* and *user_attributes* to take advantage of this new feature. If running an appliance built with CentOS version prior to CentOS 7.4 do not include *domainname* for these attributes.

```
[ifp]
user_attributes = +mail, +givenname, +sn, +displayname, +domainname
```

and update *ldap_user_extra_attrs* to include *domainname* where appropriate.

```
[domain/example.com]
ldap_user_extra_attrs = mail, givenname, sn, displayname, domainname
```

- DNS Configuration Significance

Special care should be made when configuring DNS as improper DNS configurations can result in poor performance and improper functionality. For more details refer to the following documents:

- freeipa.org Active Directory Trust Setup/DNS configuration
- freeipa.org DNS
- freeipa.org DNSSEC

Enable the MiQ Appliance to use the configued IPA server

Use the Appliance Console to enable external authentication to the IPA Server.

1. Log in to console as *root*
2. Run *appliance_console*
3. Summary screen should show **External Auth** as *not configured*, Press any key
4. From the Advanced Setting menu, select the menu item **Configure External Authentication (httpd)**
5. Enter the FQDN of the IPA Server, i.e. *ipaserver.test.company.com*
6. Enter the IPA Server domain, i.e. *test.company.com*
7. Enter the IPA Server realm, i.e. *TEST.COMPANY.COM*
8. Press enter to select the default IPA Server Principal, i.e. *admin*
9. Enter the Password of the IPA Server Principal
10. Review details, and Enter **y** to proceed.

Configure the MiQ appliance to use external authentication

Log in to the MiQ appliance as admin, then in *Settings→Configuration→Server→Authentication*

- Set mode to External (httpd)
- Check: *Get User Groups from External Authentication (httpd)*
- *Optionally* Check: *Enable Single Signon*
- Click Save.

The above steps need to be done on each UI and WebService enabled appliance.

in *Settings→Configuration→Access Control*

- Make sure the user's groups are created on the Appliance and appropriate roles assigned to those groups.

Create groups on the MiQ appliance

The below steps need to be done on each UI and WebService enabled appliance.

Log in to the MiQ appliance as admin, then in *Settings→Configuration→Access Control→Groups→Configuration→Add a new Group*

- Check: *(Look Up LDAP Groups)*
- Enter the AD user as the *User to Look Up* i.e. *ipauser@ipaserver.test.company.com*
- Click Retrieve.
- Choose a group from the *LDAP Groups for User* dropdown.
- Assign the appropriate roles to the group
- Click Add.

Use AD Trust Authentication

Once the above is done, the user simply needs to specify their AD Username and Password when:

- Logging into the Appliance Administrative UI
- Accessing the REST API
- Using the Self Service UI
- Using the Single Sign On (SSO) to access the MiQ appliance after generating a Kerberos ticket by using kinit with AD credentials.

- LDAP
- Testing OpenLDAP
  - With SSL
  - With User Bind DN and Password
- Configure SSSD
  - Configure sssd.conf with authselect (CentOS 8 and newer)
  - Configure sssd.conf with authconfig (CentOS 7 and older)
  - Update sssd.conf
    - Testing SSSD Updates
- Configure Apache
- Configure SELinux
- Configure Administrative UI

LDAP

This guide covers how to manually configure an appliance's external authentication to work against LDAP. This provides LDAP users access to the Appliance UI as well as the REST API.

1. Testing OpenLDAP
2. Configure SSSD
3. Configure Apache
4. Configure SELinux
5. Configure Administrative UI

In these examples, the LDAP domain used will be *example.com*

Note: The SSSD and OpenLDAP configurations shown below are simply examples. Depending on the Ldap environment, Ldap directory server used, the configurations can widely differ. Many other options are available to the *authconfig* command as well as the *sssd.conf* configuration file.

The following reference are available for customizing SSSD for any LDAP directory:

```
# man authconfig
# man ldap.conf
# man sssd-ldap
```

The document @ https://onemoretech.wordpress.com/2014/02/23/sssd-for-ldap-auth-on-linux/ is also helpful.

Testing OpenLDAP

With SSL

For Configuring external authentication to Ldap, the sssd-ldap communication must be secured (SSL or Start TLS). So an SSL Certificate needs to be created.

The *apacheds-cert.pem* certificate file is used in the examples below.

Update the LDAP Directory Server to point to the keystore for the certificate. This is used for both SSL and Start TLS.

For SSL, update the **/etc/openldap/ldap.conf** accordingly to test:

```
SASL_NOCANON    on

URI             ldaps://ldap-example.com:10636
BASE            dc=example,dc=com
TLS_REQCERT     demand
TLS_CACERTDIR   /etc/openldap/cacerts
TLS_CACERT      /etc/openldap/cacerts/apacheds-cert.pem
```

Then try out a search:

```
# ldapsearch -x -H ldaps://ldap-example:10636 -LLL \
       -b "ou=people,dc=example,dc=com" -s sub     \
       "(objectclass=organizationalPerson)"
```

With User Bind DN and Password

You can also test the ldap directory by specifying a user Bind DN and password on the Ldap search command line as follows:

```
# ldapsearch -x -H ldap://ldap-example:389 -LLL                 \
       -b "ou=people,dc=example,dc=com" -s sub                  \
       -D "ui=:userid,ou=People,dc=example,dc=com" -w :password \
       "(objectclass=organizationalPerson)"
```

Configure SSSD

There are two different ways that SSSD can be configured.

For *CentOS 7* and earlier **authconfig** is used to configure SSSD. Starting with *CentOS 8* a new command, **authselect** is used.

Perform the steps in only one of these two sections below:

- Configure sssd.conf with authselect (CentOS 8 and newer)
- Configure sssd.conf with authconfig (CentOS 7 and older)

Note: SSSD must be configured to communicate with LDAP over SSL or Start TLS. Otherwise an authentication error will be returned to the Appliance login screen.

Configure sssd.conf with authselect (CentOS 8 and newer)

```
  # authselect select sssd --force
```

Seed the sssd.conf configuration file

authselect does not automatically generate the */etc/sssd/sssd.conf* file, however the template, *sssd.conf.erb*, is now delivered with ManageIQ which can be used to seed the */etc/sssd/sssd.conf*.

```
# TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"
# mkdir -p /etc/sssd
# cp ${TEMPLATE_DIR}/etc/sssd/sssd.conf.erb /etc/sssd/sssd.conf
# chmod 600 /etc/sssd/sssd.conf
```

Update the SSSD configuration file /etc/sssd/sssd.conf to specify the correct *ldap_search_base* and *ldap_uri*:

```
...
ldap_search_base = dc=example,dc=com
id_provider = ldap
auth_provider = ldap
chpass_provider = ldap
ldap_uri = ldaps://ldap-example.com:10636
ldap_id_use_start_tls = False
cache_credentials = True
...
```

Configure sssd.conf with authconfig (CentOS 7 and older)

```
  # authconfig \
    --enablesssd                                \
    --enablesssdauth                            \
```

```
    --enablelocauthorize                        \
    --enableldap                                \
    --enableldapauth                            \
    --ldapserver=ldaps://ldap-example.com:10636 \
    --disableldaptls                            \
    --ldapbasedn=dc=example,dc=com              \
    --enablerfc2307bis                          \
    --enablecachecreds                          \
    --update
```

Update sssd.conf

Edit the different sections in **/etc/sssd/sssd.conf** for the Appliance as in the following example, customizing the main **[domain/example.com]** section for the particular Ldap installation.

Note: Starting with SSSD version 1.15.2, which will be available in CentOS version 7.4, SSSD will provide the domain name as a user attribute. The below examples show how to set *ldap_user_extra_attrs* and *user_attributes* to take advantage of this new feature. If running an appliance built with CentOS version prior to CentOS 7.4 do not include *domainname* for these attributes.

```
=>  [domain/example.com]
    autofs_provider = ldap
    id_provider = ldap
    auth_provider = ldap
    chpass_provider = ldap
    ldap_schema = rfc2307bis

    ldap_uri = ldaps://ldap-example.com:10636
    ldap_id_use_start_tls = False
=>  ldap_tls_cacertdir = /etc/openldap/cacerts
=>  ldap_tls_cacert = /etc/openldap/cacerts/apacheds-cert.pem

=>  ldap_pwd_policy = none

    ldap_search_base = dc=example,dc=com
=>  ldap_network_timeout = 3

=>  ldap_user_search_base = ou=people,dc=example,dc=com
=>  ldap_user_object_class = posixAccount
=>  ldap_user_name = uid
=>  ldap_user_uid_number = uidNumber

=>  ldap_group_object_class = groupOfNames
=>  ldap_group_search_base = ou=user_groups,dc=example,dc=com
=>  ldap_group_name = cn
=>  ldap_group_member = member

    cache_credentials = True
=>  entry_cache_timeout = 600

=>  ldap_user_extra_attrs = mail, givenname, sn, displayname, domainname

    [sssd]
=>  domains = example.com
    config_file_version = 2
=>  services = nss, pam, ifp
=>  sbus_timeout = 30
=>  default_domain_suffix = example.com

    [nss]
    homedir_substring = /home

    [pam]
=>  default_domain_suffix = example.com

=>  [ifp]
=>  default_domain_suffix = example.com
=>  allowed_uids = apache, root, manageiq
=>  user_attributes = +mail, +givenname, +sn, +displayname, +domainname
```

Testing SSSD Updates

The best way to test manual SSSD updates is to query a user via the message bus system (D-Bus) API. When user attributes and groups are properly returned, that is a good indication that the External Authentication would succeed.

For the *dbus-send* command to succeed when run from command line, SELinux must be in permissive mode.

```
# setenforce 0
```

After each update to **sssd.cond**, restart the sssd service before testing the changes.

```
systemctl restart sssd
```

Example query of user attributes for user evmuser. This primarily validates the *ldap_user_* attributes of **sssd.conf**.

```
# dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe /org/freedesktop/sssd/infopipe
org.freedesktop.sssd.infopipe.GetUserAttr string:evmuser array:string:mail,givenname,sn,displayname,domainname
```

Query groups of user evmuser. This primarily validates the *ldap_group_* attributes of **sssd.conf**.

```
# dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe /org/freedesktop/sssd/infopipe
org.freedesktop.sssd.infopipe.GetUserGroups string:evmuser
```

With logging for each of the **sssd.conf** section in the appropriate /var/log/sssd/ log file. The log files of primary interest are:

Log File	Type of debugging
/var/log/sssd/sssd.log	SSSD communication with processes
/var/log/sssd/sssd_example.com.log	sssd-ldap communication to the LDAP server
/var/log/sssd/sssd_ifp.log	Gathering user and group information from LDAP server

For each of the log file here, an debug level attribute (maximum value of 9) can be added to the appropriate **sssd.conf** section as follows:

```
    [ifp]
=>  debug_level = 9
    default_domain_suffix = example.com
    ...
```

When done testing, re-enable SELinux enforcing mode.

```
# setenforce 1
```

Note: When testing authentication on the Appliance, if updates are done in LDAP and not immediately seen when authenticating, then clean the SSSD cache as follows then retry:

```
# sss_cache -E
```

Configure Apache

Create the Apache configuration files

```
# TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"
# cp ${TEMPLATE_DIR}/etc/pam.d/httpd-auth                          \
                    /etc/pam.d/httpd-auth
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user.conf       \
                    /etc/httpd/conf.d/
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth.conf.erb \
                    /etc/httpd/conf.d/manageiq-external-auth.conf
```

Configure SELinux

For SSSD to Ldap, assure that non-standard Ldap ports (other than 389 or 636) are allowed:

```
# semanage port -a -t ldap_port_t -p tcp 10389
# semanage port -a -t ldap_port_t -p tcp 10636
```

Set appropriate SELinux permissions:

```
# setsebool -P allow_httpd_mod_auth_pam on
# setsebool -P httpd_dbus_sssd          on
```

Restart Services

```
# systemctl restart sssd
# systemctl restart httpd
```

Configure Administrative UI

Log in as admin, navigate to *Settings > Application Settings > Settings*. Select the server, then select the *Authentication tab*

- Set mode to External (httpd)
- Check: *Get User Groups from External Authentication (httpd)*
- Do not check: *Enable Single Signon* since Kerberos is not configured against LDAP.
- Click Save.

The above steps need to be completed on each UI and WebService enabled appliance.

Navigate to *Settings > Application Settings > Access Control*

- Make sure the user's LDAP group for the appliance are created and appropriate roles assigned to those groups.

- SAML
- Requirements
- Configure SAML
  - Apache Configuration
  - SAML Configuration
    - SAML Assertions
- Configure Administrative UI
- Re-Enabling Local Login
  - Administrative UI:
  - Appliance Console Interface:

    - Appliance Console CLI:
- Configuring SAML using the CLI
    - Configure SAML
    - Unconfigure SAML

SAML

External Authentication on the Appliance has been enhanced to support SAML. The SAML implementation has been tested with KeyCloak 1.8.

In this guide we will cover how to manually configure an Appliance's external authentication to work with SAML. The SAML implementation has been tested with KeyCloak but is implemented generically using Apache's mod_auth_mellon module and should work with other SAML Identity Providers.

The current implementation only secures the Appliance's Web administrative UI with SAML.

Note: The REST API and Self-Service UI do not currently support SAML.

1. Requirements
2. Configuring SAML
3. Configuring the Administrative UI
4. Re-Enabling Local Login
5. Configuring SAML using the CLI

Requirements

The following is needed in order to enable SAML Authentication to the Appliance:

- A CentOS/RHEL 7.2 based Appliance
- A SAML Identity Provider, (e.g. KeyCloak 1.8 or Later)

Configure SAML

All SAML related certificates and keys are accessed from /etc/httpd/saml2/

First ssh to the appliance as root, then create that directory:

```
# mkdir -p /etc/httpd/saml2
```

Apache Configuration

Copy the Apache remote user and SAML template configuration files:

```
# TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user.conf        \
    /etc/httpd/conf.d/
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth-saml.conf \
    /etc/httpd/conf.d/
```

SAML Configuration

Notable defaults in the *manageiq-external-auth-saml.conf* file:

Description	Default
Identity Provider Files (i.e. KeyCloak)	
Metadata File	/etc/httpd/saml2/idp-metadata.xml
Service Provider Files (i.e. mod_auth_mellon)	
Private Key File	/etc/httpd/saml2/miqsp-key.key
Certificate File	/etc/httpd/saml2/miqsp-cert.cert
Metadata File	/etc/httpd/saml2/miqsp-metadata.xml

Other mod_auth_mellon parameters must not be modified as the Appliance expects them to be defined as such, i.e. End points, Protected URL, etc.

For the Service Provider files on the Appliance, these can be generated using the mod_auth_mellon command *mellon_create_metadata.sh* as follows:

```
# cd /etc/httpd/saml2
# /usr/libexec/mod_auth_mellon/mellon_create_metadata.sh https://<miq-appliance> https://<miq-appliance>/saml2
```

The mellon_create_metadata.sh script creates file names based on the appliance URL but can be renamed to match the expected file names from the *manageiq-external-auth-saml.conf* file:

```
# mv https_<miq-appliance>.key  miqsp-key.key
# mv https_<miq-appliance>.cert miqsp-cert.cert
# mv https_<miq-appliance>.xml  miqsp-metadata.xml
```

With the service provider metadata.xml file generated, the Service Provider definition can now be defined in the Identity Provider.

For KeyCloak, a Realm can be created for one or more Appliances with individual Clients defined one per Appliance where the Client ID is essentially the URL of the appliance.

Adding a Client in the KeyCloak ManageIQ Realm:

- Select and Import the miqsp-metadata.xml file created for mod_auth_mellon.
- Set Client ID as https://<miq-appliance>
- Set Client protocol as **saml**.

The Client definition for the appliance can then be updated with the following:

Setting	Value
Name ID Format	username
Valid Redirect URIs	https://<miq-appliance/saml2/paosResponse>
	https://<miq-appliance/saml2/postResponse>
Base URL	https://<miq-appliance/>

Fine Grain SAML Endpoint Configuration	
Setting	**Value**
Assertion Consumer Service POST Binding URL	https://<miq-appliance/saml2/postResponse>
Assertion Consumer Service Redirect Binding URL	https://<miq-appliance/saml2/postResponse>
Logout Service Redirect Binding URL	https://<miq-appliance/saml2/logout>

The Identity Provider's Metadata file idp-metadata.xml can then be obtained as follows:

```
# cd /etc/httpd/saml2
# curl -s -o idp-metadata.xml \
    http://<keycloak-server>:8080/auth/realms/<miq-realm>/protocol/saml/descriptor
```

Finally, restart Apache on the appliance as follows:

```
# systemctl restart httpd
```

SAML Assertions

For authenticating to the appliance, the following remote user parameters are looked at by the appliance upon a successful login and redirect from the Identity Provider.

HTTP Environment	SAML Assertion
REMOTE_USER	username
REMOTE_USER_EMAIL	email
REMOTE_USER_FIRSTNAME	firstname
REMOTE_USER_LASTNAME	lastname
REMOTE_USER_FULLNAME	fullname
REMOTE_USER_GROUPS	groups

For KeyCloak, the above SAML Assertions can be created for the Appliance Client in KeyCloak as Mappers.

Name	Mapper Type	Property	Friendly Name	SAML Attribute Name	SAML Attribute Name Format
username	User Property	username	username	username	Basic
email	User Property	email	email	email	Basic
firstname	User Property	firstName	firstname	firstname	Basic
lastname	User Property	lastName	lastname	lastname	Basic
Name	**Mapper Type**	**User Attribute**	**Friendly Name**	**SAML Attribute Name**	**SAML Attribute Name Format**
fullname	User Attribute	fullName	fullname	fullname	Basic

Name	Mapper Type	Group attribute name	Friendly Name	SAML Attribute Name	SAML Attribute Name Format
groups	Group List	groups	groups	groups	Basic

Note: The fullName attribute was not available in the default database as of this writing and must be added to each user as a user attribute.

Configure Administrative UI

After having configured Apache for SAML, the next step is to update the Appliance Administrative UI to be SAML aware and function accordingly.

Login as admin, then in *Configure→Configuration→Authentication*

- Set mode to External (httpd)
- Check: *Enable Single Signon* - With this option enabled, initial access to the Appliance Administrative UI will redirect to the SAML Identity Provider authentication screen. Note that logouts from the Appliance will return the user to the Appliance login screen allowing them to login as admin unless *Disable Local Login* is checked below.
- Check: *provider Type: _Enable SAML* - This enables the SAML login button on the login screen, the redirects to the SAML protected page for authentication, and supports the SAML logout process.
- Optional: Check: *Disable Local Login* - Do this **only** if you need to disable *admin* login to appliance and only allow SAML based authentication. Note that if there are issues with the Identity Provider or you need admin access to the appliance you won't be able to login until you re-enable the Local Login as described

below.

- Check: *Get User Groups from External Authentication (httpd)*
- Click Save.

The above steps need to be done on each UI enabled appliance.

in *Configure→Configuration→Access Control*

- Make sure the user's groups are created on the Appliance and appropriate roles assigned to those groups.

Re-Enabling Local Login

If the Local Login has been disabled in the Administrative UI and there is a need to be able to login as *admin*, the Local Login can be re-enabled as follows:

Administrative UI:

This option is available if the Identity Provider is available and one can login using a user with enough administrative privileges to update it:

- Login as administrative user,
- In then in *Configure→Configuration→Authentication* uncheck *Disable Local Login* and save.

Appliance Console Interface:

- ssh to the appliance as root
- Run *appliance_console*
- Select menu entry *Update External Authentication Options*
- Select *Enable Local Login*
- then *Apply updates*

Appliance Console CLI:

- ssh to the appliance as root
- Run *appliance_console_cli –extauth-opts* **local_login_disabled=false**

Configuring SAML using the CLI

Another way to configure SAML on the appliance is to use the appliance console CLI instead of the above steps. The **`--saml-config`** subcommand of the appliance console CLI allows one to configure the appliance for SAML Authentication by updating both Apache configurations as well as the necessary Administrative UI settings. The **`--saml-unconfig`** subcommand is also provided for unconfiguring the appliance SAML Authentication and reverting the appliance to Database Authentication.

Usage of the new Appliance Console CLI subcommands are as follows:

Configure SAML

To configure the appliance for SAML Authentication:

```
# appliance_console_cli --saml-config [--saml-client-host=miq_appliance.fqdn] \
                                       --saml-idp-metadata=file|url           \
                                      [--saml-enable-sso]
```

When –saml-client-host is not specified, the configured appliance host is used for creating the SP metadata. This host fqdn must be reachable from the SAML IDP.

Configuring SAML requires the IDP metadata file. The –saml-idp-metadata option allows the user to specify the file path of a copy that was downloaded to the appliance or by a URL, in which case the CLI downloads the copy from the IDP.

Examples:

```
--saml-idp-metadata=/tmp/downloaded_idp_metadata.xml
```

```
--saml-idp-metadata=http://<idp-host:port>/auth/realms/<miq-realm>/protocol/saml/descriptor
```

In both cases, the IDP metadata file is copied to **`/etc/httpd/saml2/idp-metadata.xml`**

By default, SSO is not enabled, so from the Appliance login page, the user clicks on Log In to Corporate System to get redirected to the SAML login page. With this option enabled, the Appliance redirects the user to the SAML login page for logging in.

After SAML is configured, the SP metadata file can be fetched from **`/etc/httpd/saml2/miqsp-metadata.xml`** to create the related Client on the SAML IDP.

Unconfigure SAML

To unconfigure the appliance from SAML Authentication and revert to Database authentication:

```
# appliance_console_cli --saml-unconfig
```

This will remove the Apache external authentication SAML configuration files and revert the appliance's authentication settings to Database mode.

- OpenID-Connect
- Requirements
- Configuring Keycloak
  - OIDC Assertions
  - Apache Configuration
  - OIDC Configuration
- Configuring the Administrative UI
- Re-Enabling Local Login
  - Administrative UI:
  - Appliance Console Interface:
  - Appliance Console CLI:
- Configuring OIDC using the CLI
  - Configure OpenID-Connect
  - Unconfigure OpenID-Connect

OpenID-Connect

External Authentication on the Appliance has been enhanced to support OpenID-Connect, which will also be referred to as *OIDC*.

The OIDC implementation has been tested with KeyCloak 3.2.1.

In this guide we will cover how to manually configure an Appliance's external authentication to work with OIDC. The OIDC implementation has been tested with KeyCloak but is implemented generically using Apache's mod_auth_openidc module and should work with other OIDC Identity Providers.

The current implementation secures the Appliance's Web administrative and Self-Service UI with OIDC.

1. Requirements
2. Configuring Keycloak
3. OIDC Assertions
4. Apache Configuration
5. OIDC Configuration
6. Configuring the Administrative UI
7. Re-Enabling Local Login
8. Configuring OIDC using the CLI

Requirements

The following is needed in order to enable OIDC Authentication to the Appliance:

- A CentOS/RHEL 7.5 based Appliance
- A OIDC Identity Provider, (e.g. KeyCloak 3.2.1 or Later)

Configuring Keycloak

For KeyCloak, a Realm can be created for one or more Appliances with individual Clients defined one per Appliance where the Client ID identifies the individual appliance, e.g. the appliance hostname.

Adding a Client in the KeyCloak ManageIQ Realm:

- Set Client ID as <miq-appliance>
- Set Client protocol as **openid-connect**.

The Client definition for the appliance can then be updated with the following:

Setting	Value
Client Protocol	openid-connect
Access Type	confidential
Valid Redirect URIs	https://<miq-appliance>
	https://<miq-appliance/ui/service/>
	https://<miq-appliance/oidc_login/redirect_uri>
User Info Signed	unsigned
Request Object Signature Algorithm	any

Note: The trailing / for the /ui/service/ entry is required.

The value used for OIDCClientSecret (*oidc_client_secret*) in the *manageiq-external-auth-oidc.conf* file is taken from the Keycloak Client's *Credentials / Secret* value. To generate a new client secret, First set the *Credentials / Client Authenticator* to *Client Id and Secret* . Then *Regenerate Secret* button.

OIDC Assertions

For authenticating to the appliance, the following remote user parameters are looked at by the appliance upon a successful login and redirect from the Identity Provider.

HTTP Environment	OIDC Assertion
REMOTE_USER	username
REMOTE_USER_EMAIL	email
REMOTE_USER_FIRSTNAME	firstname
REMOTE_USER_LASTNAME	lastname
REMOTE_USER_FULLNAME	fullname
REMOTE_USER_GROUPS	groups

For KeyCloak, the above OIDC Assertions can be created for the Appliance Client in KeyCloak as Mappers.

The following mappers are automatically generated when the OIDC client is created:

Name	Category	Mapper Type
email	Token mapper	User Property
family name	Token mapper	User Property
given name	Token mapper	User Property
full name	Token mapper	User's full name
username	Token mapper	User Property

The following *Group Membership* mapper must be manually created:

Name	Consent Required	Mapper Type	Token Claim Name	Full group path	Add to ID token	Add to access token	Add to userinfo	Add to token introspection
groups	OFF	Group Membership	groups	OFF	ON	OFF	OFF	ON

The following *User Session Note* mappers must be manually created:

Name	Consent Required	Mapper Type	User Session Note	Token Claim Name	Claim JSON Type	Add to ID token	Add to access token
Client Host	OFF	User Session Note	clientHost	clientHost	String	ON	ON
Client ID	OFF	User Session Note	clientId	clientId	String	ON	ON
Client IP Address	OFF	User Session Note	clientAddress	clientAddress	String	ON	ON

Apache Configuration

Copy the Apache OIDC template configuration files:

```
# TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user-openidc.conf \
    /etc/httpd/conf.d/
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth-openidc.conf.erb \
    /etc/httpd/conf.d/manageiq-external-auth-openidc.conf
```

OIDC Configuration

The Apache */etc/httpd/conf.d/manageiq-external-auth-openidc.conf* configuration files must be updated with installation specific values:

The defaults in the *manageiq-external-auth-oidc.conf* file are:

Name	Value
LogLevel	warn
OIDCCryptoPassphrase	sp-cookie
OIDCOAuthRemoteUserClaim	username

Installation specific values must be specified in the *manageiq-external-auth-oidc.conf* file for the following parameters, where the *parameter* including the <%= ... %> anchors need to be replaced with the appropriate values:

Name	Parameter	Value
ServerName and OIDCRedirectURI	miq_appliance	FQDN of the appliance
OIDCProviderMetadataURL	oidc_provider_metadata_url	The OpenID-Connect Provider URL, i.e. http://<keycloak_server>:8080/auth/realms/miq/.well-known/openid-configuration
OIDCClientID and OIDCOAuthClientID	oidc_client_id	The OpenID-Connect Client ID
OIDCClientSecret end OIDCOAuthClientSecret	oidc_client_secret	The OpenID-Connect Client Secret

Name	Parameter	Value
OIDCOAuthIntrospectionEndpoint	oidc_introspection_endpoint	The OpenID-Connect Introspection Endpoint

Optional parameters can be appended to the end fo the configuration file to avoid SSL verification. These can be used when a valid SSL server certificate is not available for communicating with the OpenID Provider.

NOTE: Using these parameters is insecure and should only be used in a development environment.

Name	Value
OIDCSSLValidateServer	Off
OIDCOAuthSSLValidateServer	Off

Finally, restart Apache on the appliance as follows:

```
# systemctl restart httpd
```

Configuring the Administrative UI

After having configured Apache for OIDC, the next step is to update the Appliance Administrative UI to be OIDC aware and function accordingly.

Login as admin, then in *Configure→Configuration→Authentication*

- Set mode to External (httpd)
- Check: *Provider Type: Enable Enable OpenID-Connect* - This enables the OIDC login button on the login screen, that redirects to the OIDC protected page for authentication, and supports the OIDC logout process.
- Optional: Check: *Enable Single Signon* - With this option enabled, initial access to the Appliance Administrative UI will redirect to the OIDC Identity Provider authentication screen. Note that logouts from the Appliance will return the user to the Appliance login screen allowing them to login as admin unless *Disable Local Login* is checked below.
- Optional: Check: *Disable Local Login* - Do this **only** if you need to disable *admin* login to appliance and only allow OIDC based authentication. Note that if there are issues with the Identity Provider or you need admin access to the appliance you won't be able to login until you re-enable the Local Login as described below.
- Check: *Get User Groups from External Authentication (httpd)*
- Click Save.

The above steps need to be done on each UI enabled appliance.

In Configure→Configuration→Access Control

- Make sure the user's groups are created on the Appliance and appropriate roles assigned to those groups.

Re-Enabling Local Login

If the Local Login has been disabled in the Administrative UI and there is a need to be able to login as *admin*, the Local Login can be re-enabled using either of the following methods:

Administrative UI:

This option is available if the Identity Provider is available and one can login using a user with enough administrative privileges to update it:

- Login as administrative user,
- In then in *Configure→Configuration→Authentication* uncheck *Disable Local Login* and save.

Appliance Console Interface:

- ssh to the appliance as root
- Run *appliance_console*
- Select menu entry *Update External Authentication Options*
- Select *Enable Local Login*
- then *Apply updates*

Appliance Console CLI:

- ssh to the appliance as root
- Run *appliance_console_cli –extauth-opts* **local_login_disabled=false**

Configuring OIDC using the CLI

Another way to configure OpenID-Connect on the appliance is to use the appliance console CLI instead of the above steps. The `--oidc-config` subcommand of the appliance console CLI allows one to configure the appliance for OpenID-Connect Authentication by updating both Apache configurations as well as the

necessary Administrative UI settings. The `--oidc-unconfig` subcommand is also provided for unconfiguring the appliance OpenID-Connect Authentication and reverting the appliance to Database Authentication.

Usage of the new Appliance Console CLI subcommands are as follows:

Configure OpenID-Connect

To configure the appliance for OpenID-Connect Authentication:

```
# appliance_console_cli --oidc-config
                        [--oidc-client-host]
                        --oidc-url
                        --oidc-client-id
                        --oidc-client-secret
                        [--oidc-introspection-endpoint]
                        [--oidc-enable-sso]
                        [--oidc-insecure]
```

When the `--oidc-introspection-endpoint` is not specified, an attempt will be made to fetch the introspection endpoint by querying the OpenID-Connect Provider URL.

When the `--oidc-client-host` is not specified, the configured appliance host is used for the ServerName and OpenID-Connect Redirect URI in the OpenID-Connect Apache configuration file. The `--oidc-client-host` is usually the externally reachable FQDN of the appliance.

When `--oidc-insecure` is specified no SSL verification will be done when attempting to fetch the introspection endpoint and SSL verification will be disabled using parameters `OIDCSSLValidateServer` and `OIDCOAuthSSLValidateServer` as described above.

Configuring OpenID-Connect requires the following parameters:

Parameter	Description
--oidc-url	This is the OpenID-Connect Provider URL
--oidc-client-id	The OpenID-Connect Provider Cliend ID
--oidc-client-secret	The OpenID-Connect Provider Client Secret

The `--oidc-url` varies for different providers. An example provider URL for keycloak:

```
--oidc-url=http://keycloak-server:8080/auth/realms/miq-realm/.well-known/openid-configuration
```

With **miq-realm** being the Keycloak Realm this OpenID-Connect appliance client is being configured for.

The `--oidc-client-id` is the Client ID specified for the appliance in the OpenID-Connect provider, usually just `<miq-appliance-fqdn>`

By default, SSO is not enabled, so from the Appliance login page, the user clicks on Log In to Corporate System to get redirected to the OpenID-Connect login page. With the `--oidc-enable-sso` option specified, the Appliance redirects the user to the OpenID-Connect login page for logging in.

Unconfigure OpenID-Connect

To unconfigure the appliance from OpenID-Connect Authentication and revert to Database authentication:

```
# appliance_console_cli --oidc-unconfig
```

This will remove the Apache external authentication OpenID-Connect configuration files and revert the appliance's authentication settings to Database mode.

- Integration with OpenShift Container Platform
  - Overview
  - Prerequisites
  - Retrieving the OpenShift Container Platform Management Token
  - Configuring Infrastructure Automation
  - Configuring Infrastructure Automation Capacity and Utilization
    - Enabling SmartState Analysis
  - Adding OpenShift Container Platform as a Container Provider
  - Container Image Scanning
  - Configuring Image Scanning
  - Scheduling A Recurring Scan
  - Working with Images
    - Viewing Results
    - Manual Scanning
    - Evaluating Compliance
    - Generating a Report on Images
  - OpenSCAP Policy Profile
    - Customizing the Scanning Policy Profile
  - Controlling OpenShift Pod Execution
  - Reference

Integration with OpenShift Container Platform

Overview

This guide walks you through adding a Red Hat OpenShift Container Platform cluster to a Infrastructure Automation container provider catalog. This deployment focuses on enabling the OpenShift Container Platform cluster metrics plug-in, so that Infrastructure Automation can collect information from an OpenShift Container Platform cluster upon integration.

Each procedure in this guide is covered in greater detail in the Infrastructure Automation and OpenShift Container Platform product documentation. However, links to the corresponding sections are provided for more detail.

The following sections will describe the required configuration for both products prior to integration.

Prerequisites

This guide assumes that you have:

- Already deployed Infrastructure Automation on your chosen platform
- Already deployed OpenShift Container Platform

When enabling metrics on OpenShift Container Platform, you can store your metrics data on *persistent* or *non-persistent* storage. To use persistent storage, you need to provision a persistent volume specifically for this purpose before configuring the metrics components. See Persistent Volumes in the OpenShift Container Platform *Architecture* documentation for more information.

Retrieving the OpenShift Container Platform Management Token

After enabling cluster metrics on your OpenShift Container Platform deployment, retrieve the *management token* while you are still logged in to the OpenShift Container Platform host. This will be required later in Adding OpenShift Container Platform as a Container Provider.

Run the following to obtain the token needed to add a Red Hat OpenShift provider:

1. Obtain the **`management`** service account token name:

```
# oc describe sa -n $project_name $service_account_name
...
Tokens:  management-admin-token-0f3fh
         management-admin-token-q7a87
```

2. Select and describe one of the tokens to retrieve the full token output, replacing **`management-admin-token-0f3fh`** with the name of your token:

```
# oc describe secret -n $project_name management-admin-token-0f3fh
...
Data
====
token:  eyJhbGciOiJSUzI1NiI...
```

Configuring Infrastructure Automation

Configuring Infrastructure Automation involves two steps:

1. Configuring Infrastructure Automation Capacity and Utilization, and
2. Enabling SmartState Analysis

These steps are required to allow Infrastructure Automation to collect metrics from OpenShift Container Platform Enabling OpenShift Container Platform Metrics and use them to perform a SmartState analysis. You can choose different servers to perform either function; the following sections assume that you will.

Configuring Infrastructure Automation Capacity and Utilization

For metrics collection to work properly, you also need to configure Infrastructure Automation to allow for all three **Capacity & Utilization** server roles, which are available from the settings menu under **Configuration > Server > Server Control**.

To enable these server roles:

1. Click **Configuration**, then select the server to configure from menu: **Settings > Zone** in the accordion menu on the left.
2. Browse to the **Server Roles** list in the menu: **Server > Server Control** section. From there, set the required capacity and utilization roles to **ON**, namely:
    1. **Capacity & Utilization Coordinator**
    2. **Capacity & Utilization Data Collector**
    3. **Capacity & Utilization Data Processor**
3. Click **Save**.

Data collection is enabled immediately. However, the first collection begins 5 minutes after the server is started, and every 10 minutes after that. Therefore, the longest the collection takes after enabling the Capacity & Utilization Collector role is 10 minutes. The first collection from a particular provider may take a few minutes since Infrastructure Automation is gathering data points going one month back in time.

For more information, see Capacity and Utilization Collection in the *Deployment Planning Guide*.

Enabling SmartState Analysis

After enabling the required server roles, enable SmartState analysis. See Smart State Analysis Support from the Support Matrix.

Enabling SmartState analysis is similar to Assigning the Capacity and Utilization Server Roles, in that the procedure also involves enabling server roles on a specific server. To do so:

1. Click **Configuration**.
2. Select the server to configure from menu: **Settings**[Zone] in the left pane of the appliance.
3. Browse to the **Server Roles** list in the menu: **Server > Server Control** section. From there, set the appropriate SmartState roles to **ON**. Namely:
    1. **SmartProxy**
    2. **SmartState Analysis**
4. Click **Save**.

Adding OpenShift Container Platform as a Container Provider

At this point, you should now be ready to add your OpenShift cluster to Infrastructure Automation as a container provider. To do so, prepare the token you retrieved in Retrieving the OpenShift Container Platform Management Token and follow the procedure below:

1. Browse to menu: **Compute > Containers > Providers**.
2. Click **Configuration**, then click (**Add a New Containers Provider**).
3. Enter a **Name** for the provider.
4. From the **Type** list, select **OpenShift Container Platform**.
5. Enter the appropriate **Zone** for the provider. If you do not specify a zone, it is set to **`default`**.
6. In the **Default** tab, configure the following for the OpenShift provider:
    1. Select a **Security Protocol** method to specify how to authenticate the provider:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.

          Note:

          You can obtain your OpenShift Container Platform provider's CA certificate for all endpoints (default, metrics, alerts) from **`/etc/origin/master/ca.crt`**. Paste the output (a block of text starting with **`-----BEGIN CERTIFICATE-----`**) into the **Trusted CA Certificates** field.
        - **SSL without validation**: Authenticate the provider insecurely (not recommended).
    2. Enter the **Hostname** (or IPv4 or IPv6 address) of the provider.

       Important:

       The **Hostname** must use a unique fully qualified domain name.
    3. Enter the **API Port** of the provider. The default port is **`6443`**.
    4. Enter a token for your provider in the **Token** box.

       Note:

       To obtain a token for your provider, run the **`oc get secret`** command on your provider; see Obtaining an OpenShift Container Platform Management Token.

       For example:

       # oc get secret –namespace management-infra management-admin-token-8ixxs –template='' | base64 –decode
    5. Click **Validate** to confirm that Infrastructure Automation can connect to the OpenShift Container Platform provider.
7. In the **Metrics** endpoint tab optionally configure the metrics service details:
    1. Select a service type (e.g.: Prometheus) or leave disabled
    2. Select a **Security Protocol** method to specify how to authenticate the service:
        - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
        - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
        - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)

3. Enter the **Hostname** (or IPv4 or IPv6 address) of the metrics service.

   In order to find the hostname you can use the `oc get route` command.

   You can retrieve the Prometheus route by running:

   ```
   oc get route prometheus-k8s -n openshift-monitoring
   ```

   Example:

   ```
   oc get route prometheus-k8s -n openshift-monitoring
   NAME              HOST/PORT                                                PATH              SERVICES   PORT
   TERMINATION          WILDCARD
   prometheus-k8s    prometheus-k8s-openshift-monitoring.ocp.example.com  prometheus-k8s    web
   reencrypt/Redirect   None
   ```

   In this example you would use `prometheus-k8s-openshift-monitoring.ocp.example.com` as the metrics hostname.

4. Enter the **API Port** of your metrics endpoint uses a non-standard port for access. The default port is `443`.

5. Click **Validate** to confirm that Infrastructure Automation can connect to the metrics endpoint.

8. In the **Virtualization** endpoint tab optionally configure the virtualization service details:

   1. Select a service type (OpenShift Virtualization) or leave disabled

   2. Select a **Security Protocol** method to specify how to authenticate the service:

      - **SSL**: Authenticate the provider securely using a trusted Certificate Authority. Select this option if the provider has a valid SSL certificate and it is signed by a trusted Certificate Authority. No further configuration is required for this option.
      - **SSL trusting custom CA**: Authenticate the provider with a self-signed certificate. For this option, copy your provider's CA certificate to the **Trusted CA Certificates** box in PEM format.
      - **SSL without validation**: Authenticate the provider insecurely using SSL. (Not recommended)

   3. Enter the **Hostname** (or IPv4 or IPv6 address) of the virtualization service.

   4. Enter the **API Port** of your virtualization service. The default port is `443`.

   5. Enter a token for your provider in the **Token** box.

9. Click the **Advanced** tab to add image inspector settings for scanning container images on your provider using OpenSCAP.

   Note:

   - These settings control downloading the image inspector container image from the registry and obtaining the Common Vulnerabilities and Exposures (CVE) information (for effective scanning) via a proxy.
   - CVE URL that Infrastructure Automation requires to be open for OpenSCAP scanning: https://www.redhat.com/security/data/metrics/ds/. This information is based on the source code of OpenSCAP.

   1. Enter the proxy information for the provider in either **HTTP**, **HTTPS**, or **NO Proxy** depending on your environment.

   2. Enter the **Image-Inspector Repository** information. For example, `openshift3/image-inspector`.

   3. Enter the **Image-Inspector Registry** information. For example, `registry.access.redhat.com`.

   4. Enter the **Image-Inspector Tag** value. A tag is a mark used to differentiate images in a repository, typically by the application version stored in the image.

   5. Enter `https://www.redhat.com/security/data/metrics/ds/` in **CVE location**.

10. Click **Add**.

Note:

You can also set global default image-inspector settings for all OpenShift providers in the advanced settings menu by editing the values under `ems_kubernetes`, instead of setting this for each provider.

For example:

```
:image_inspector_registry: registry.access.redhat.com
:image_inspector_repository: openshift3/image-inspector
```

Container Image Scanning

Configuring Image Scanning

Infrastructure Automation manages vulnerability scanning of container images. When an OpenShift provider is added, OpenShift images from the internal registry are discovered. To enable image scanning, perform the following configuration steps:

1. Browse to menu: **Compute > Containers > Providers**.

2. Select the checkboxes of the OpenShift providers on which to enable scanning.
3. From the **Policy** pull-down menu, click **Manage Policies**.
4. Select the **OpenSCAP profile** checkbox.
5. Click **Save**.

This action will trigger a SmartState analysis, or scan, of all images referenced by the OpenShift provider. The initial scan may take several hours to complete, depending on the number and size of images. The scan occurs in the OpenShift provider, which Infrastructure Automation receives and records in the database. OpenShift limits the number of scanning pods; only three images can be scanned simultaneously.

Scheduling A Recurring Scan

Software vulnerability databases are updated frequently. To apply these updates, a rescan is required. To schedule a recurring scan of container images:

 :schedule openscap scan

1. Click **Configuration**.
2. From **Settings > Zones** in the left pane of the appliance, select **Schedules**.
3. From the drop-down menu, click **Configuration > Add a new Schedule**.
4. Type an arbitrary **Name**.
5. Type an arbitrary **Description**.
6. Ensure the **Active** checkbox is selected.
7. In **Action**, select **Container Image Analysis**.
8. In **Filter**, select **All Container Images for Containers Provider**, **OpenShift**.
9. In **Run**, set the schedule as desired.
10. Set the **Time Zone**, **Starting Date**, and **Starting Time**.
11. Click **Add**.

Working with Images

Viewing Results

Image scanning results are displayed in each image summary page.

1. Select menu: **Compute > Containers > Container Images**.
2. Click the desired image.

For an OpenSCAP HTML report, locate the **Configuration** section and select **OpenSCAP HTML**.

 :container configuration

For compliance and scanning history information, locate the **Compliance** section and note the **Status** field or select **Available** from the **History** field.

 :container scan history

Manual Scanning

SmartState analysis scanning may be initiated manually for images. From an image summary page, select **Configuration > Perform SmartState Analysis**. Refreshing the image page will reflect the latest scan results and compliance history.

Evaluating Compliance

If the image scan policy has been updated since the last scan, compliance conditions may be re-evaluated. From an image summary page, select **Policy > Check Compliance of Last Known Configuration**. Refreshing the image page will reflect the latest compliance history.

Generating a Report on Images

You can output the results of an OpenSCAP scan of images to a report for an overview of the security risk level of images. The **Images by Failed OpenSCAP Rule Results are included with Infrastructure Automation and shows whether the image passed or failed OpenSCAP policy criteria, and the security risk.

Note: You can also create a copy of this report and edit it to contain additional information, such as the project name where the image is used to produce more useful results. For more information, see Editing a Report and Reportable Fields in Infrastructure Automation in *Monitoring, Alerts, and Reporting* for instructions on customizing reports.

Steps to create a report to show image compliance:

1. Browse to menu: **Overview > Reports**.
2. Click the menu: **Reports > All Reports** accordion.

3. Browse to menu: **Configuration Management > Containers > Images by Failed OpenSCAP Rule Results** to run a report that shows which images failed the OpenSCAP compliance.
4. Click **Queue**.
5. The report generation is placed in the queue and its status shows in the reports page.

		Queued At	Run At	Source	Username	Group	Status
☐		03/12/18 04:31:57 UTC		Requested by user	ocpadmin	EvmGroup-container_administrator_updated	Queued
☐		03/07/18 08:43:25 UTC	03/07/18 08:44:00 UTC	Requested by user	ocpadmin	EvmGroup-container_administrator_updated	Complete

6. Click **(Refresh this page)** to update the status.
7. Browse to the **Saved Reports** accordion, and click the report when it completes.
8. Click the report download buttons for the type of export you want. The report is automatically named with the type of report and date.
   - Click **Download this report in text format** to download as text.
   - Click **Download this report in CSV format** to download as a comma-separated file.
   - Click **Download this report in PDF format** to download as PDF.

OpenSCAP Policy Profile

Infrastructure Automation is pre-configured with a default scanning policy profile. This includes conditions to scan and identify compliance, as well as annotate compliance failure. SmartState analysis is performed when new images are added to OpenShift.

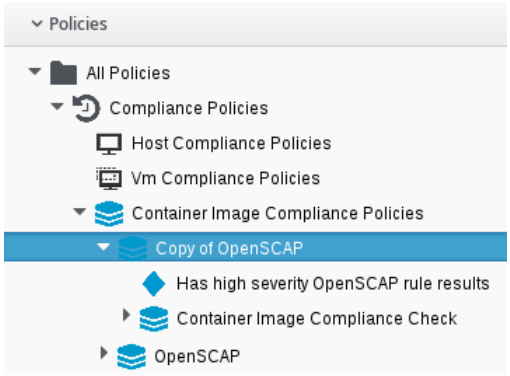
Customizing the Scanning Policy Profile

The built-in OpenSCAP policy profile cannot be edited. However, you can assign *edited* copies of these policies to a new policy profile. This provides you the ability to create a customized version of the built-in OpenSCAP policy profile.

To do so, you must first copy the policy that you want to customize:

1. Browse to menu: **Control > Explorer**.
2. Click the **Policies** accordion, select **Container Image Compliance Policies**, and click **OpenSCAP**.
3. Click **Configuration**, and an option to copy the policy appears; for example, **Copy this Container Image Policy**.
4. Click **OK** to confirm.

The new policy is created with a prefix of **Copy of** in its description, and it can be viewed in the Policies accordion.



You can now edit the copied policy. After editing copied policies, you can add them to a new policy profile. For instructions on how to edit policies, create a new policy profile, and add policies to it, see the Policies and Profiles guide. Once you have a customized policy profile, you can assign it to a containers provider.

Controlling OpenShift Pod Execution

Through the default policy profile, non-compliant images receive the control policy action **Mark as Non-Compliant**. This action annotates the **image** object (not to be confused with the **imagestream** object) with *images.openshift.io/deny-execution=true*. This annotation may be used to prevent nodes from running non-compliant images.

Reference

More information about OpenSCAP, see visit the OpenSCAP web site.

- Self Service User Interface Guide
- Infrastructure Automation Self Service
  - Accessing the Self Service User Interface
  - Navigating the Self Service User Interface
  - Dashboard Tab
  - My Services Tab
    - Setting a Retirement Date for a Service
    - Retiring a Service
    - Setting Ownership of a Service
    - Adding and Editing Tags
    - Removing a Service
    - Editing a Service
    - Controlling Virtual Machine Power States
    - Creating Virtual Machine Snapshots
    - Viewing Virtual Machine Snapshots
    - Accessing the Virtual Machine Console
    - Web Console (Cockpit) Integration
      - Accessing the Web Console
  - My Orders Tab
  - Service Catalog Tab

Self Service User Interface Guide

Infrastructure Automation Self Service

Infrastructure Automation Self Service is a web-based graphical user interface for ordering and managing IT service requests. You can enable self-service tenant end users, who can easily access their services, track requests, and manage their accounts using the Self Service user interface (SSUI), which has widgets, dashboard controls and feedback. The Self Service user interface supports role-based access control (RBAC) of menus and features, similar to in the full administrative user interface.

Note:

All catalog items, dialogs, buttons, icons among other items must be configured in the full administrative user interface before they are available to users in the Self Service user interface.

Accessing the Self Service User Interface

The Self Service user interface is running by default and does not need to be started or configured. To access the Self Service user interface, you must know the host name (or IP address) of the appliance, and the login credentials.

Self Service URL:

```
https://HOSTNAME/ui/service/
```



You can choose your preferred language for the Self Service user interface from the options in the **Language** list available on the login screen. The following languages are supported:

- English
- Español (Spanish)
- Français (French)
- 日本語 (Japanese)
- 한국어 (Korean)
- Nederlands (Dutch)
- Português (Portuguese)
- Português do Brasil (Brazilian Portuguese)
- 简体中文 (Simplified Chinese)
- Traditional Chinese

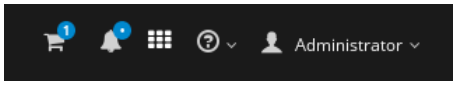
Additionally, the *User Default* and *Browser Default* options default to the language already set by the user or browser respectively.

Navigating the Self Service User Interface

The Self Service user interface has a slideout navigation menu on the left that can be minimized by clicking the menu icon (≡), like in the full administrative user interface. The slideout navigation menu comprises of the following tabs:

- **Dashboard**
- **My Services**
- **My Orders**
- **Service Catalogs**

The menu bar that runs along the top of the user interface contains several options including a shopping cart, notifications feature, full administrative user interface option, and a help option. Besides that, you can see the current logged in LDAP user, for example, **Administrator**.



- Shopping Cart Click the shopping cart icon to view services that are available to order, manage your cart as required, and order all services at one time. You can read more about the shopping cart model adopted in the Self Service user interface under ???.
- Notifications The Self Service user interface has introduced a new notifications feature that displays messages generated by the system for various events, such adding services to the shopping cart, ordering services, provisioning, service power operations, and so on. Click the bell icon to open or close the **Notifications** drawer, which shows various notification events. You can click the double arrows («) located on the upper-left of the **Notifications** drawer to expand or contract the size of the **Notifications** screen as required.

- Administration User Interface Link Clicking the menu icon ( ) shows the **Administration UI** link, which when clicked, takes you to the full administrative user interface login page in a new window or tab of your web browser.
- Help Click the help icon to access the options to open **Documentation**, **Red Hat Customer Portal**, and **About**. Click **About** to see information about your Self Service environment, such as the logged in user name and role, the browser details, and importantly, the Self Service user interface version number, which can be useful for support purposes.

Note:

The resources linked to in the help icon will be, by default, the ones listed above. System administrators can change them to other resources as required for the logged in user.

- Role-Based Access Control The Self Service user interface supports role-based access control of menus and features. If you are a member of multiple user groups, you can change the acting group by navigating to ***Administrator***

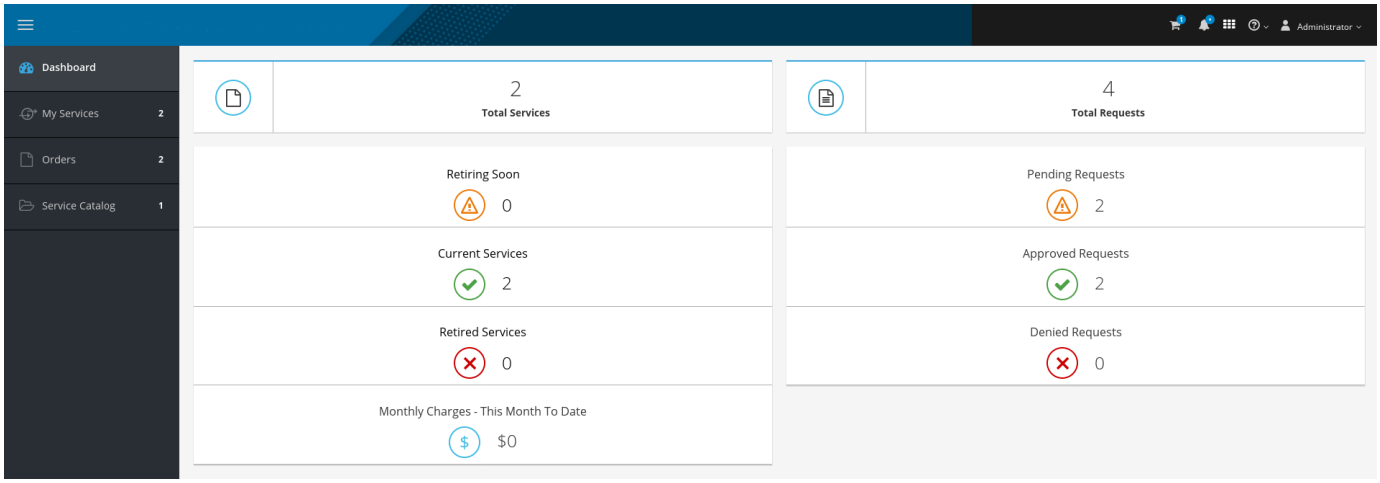
  Change Group, then select the new group from the available options in the list.
- Switch Language Although you can set your preferred language for the user interface from the login screen itself, you can switch to a different language than what is currently set also by navigating to ***Administrator***

  Switch Language within the Self Service user interface.

Dashboard Tab

Once you log in to the Self Service user interface, you will see the dashboard screen by default. The **Dashboard** tab provides a quick summary of the services available, requests made, and their statuses as below:

- Total Services
- Current Services
- Services Retiring Soon
- Retired Services
- Total Requests
- Pending Requests
- Approved Requests
- Denied Requests



- Monthly Charges - This Month To Date The Self Service user interface has added basic support for displaying monthly chargeback data of the computing and storage cost for the use of resources, in the currencies supported by the full administrative user interface. This provides a quick summary of where you are from cost standpoint to that point in the month. For more information about configuring chargeback rates in the full administrative user interface and how Infrastructure Automation calculates chargeback costs, see Chargeback in the *Monitoring, Alerts, and Reporting* guide.

Note:

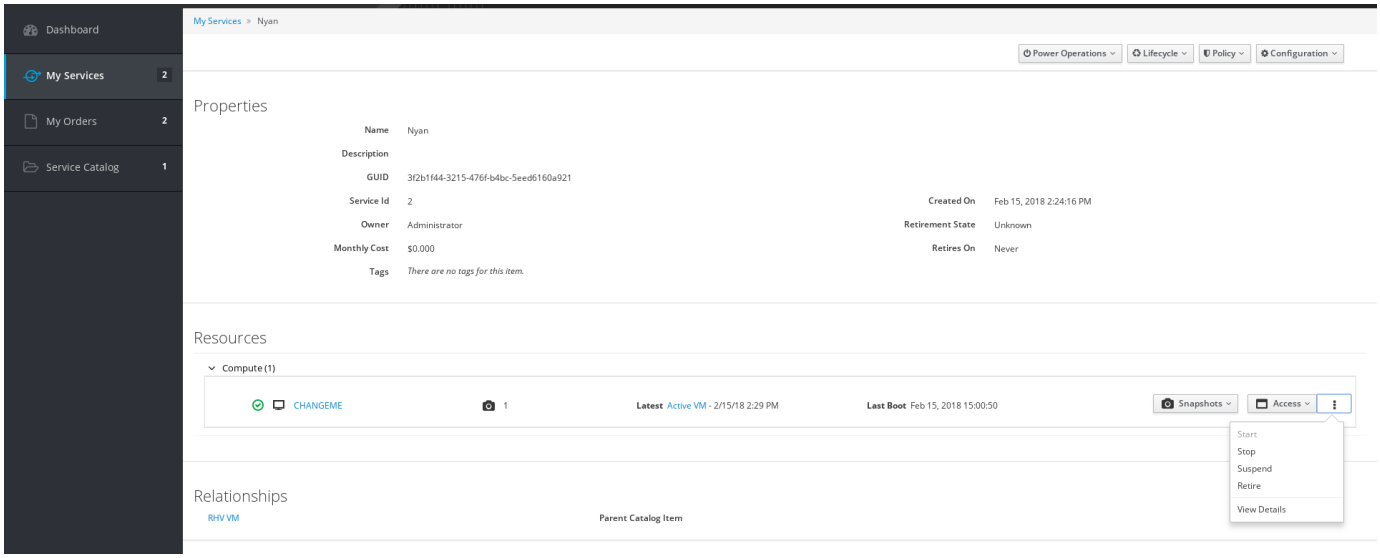
If you are on the first day of the month, there can be no data yet therefore resulting in zero dollar amount. Also, if you have a lot of data, it might not have rolled into the new month yet, resulting in no data and dollar amount for month to date.

My Services Tab

The **My Services** tab lists all service requests individually with a summary of information associated with each service, including the **Service Name**, its current **Power State** represented by an icon, **Number of VMs**, **Owner**, date **Created**, and **Retires** date. Additionally, you can see the chargeback cost per service for a virtual machine's month-to-date usage. You can filter or sort services by those entities, and by **Relative Cost**.

On the same screen, you have the option to edit the service name and description, edit tags, set ownership, retire the service now or at a later date, or remove the service by clicking more options *(represented by three dots)* next to the service. You also have options to perform power operations, that is start, stop, or suspend, at the service level. You can choose to click on a service where you can perform all of these actions at the service level using the designated buttons located at the top of the screen, as well as access the virtual machine dashboard, manage snapshots, and open the virtual machine and web console from under the **Resources** section. You can also control power operations and life cycle of virtual machine resources via more options *(represented by three dots)*.





Virtual Machine Dashboard:.

From the **My Services** screen, click on a service and then on the virtual machine name or **View Details** under more options *(represented by three dots)* to access the virtual machine dashboard. The dashboard displays the following information about the virtual machine resource:

- Compliance
- Retirement
- Snapshots
- Timeline of power events
- Utilization of CPU, Memory, and Storage
- Smart State analysis
- Tags
- Custom attributes
- Disk usage

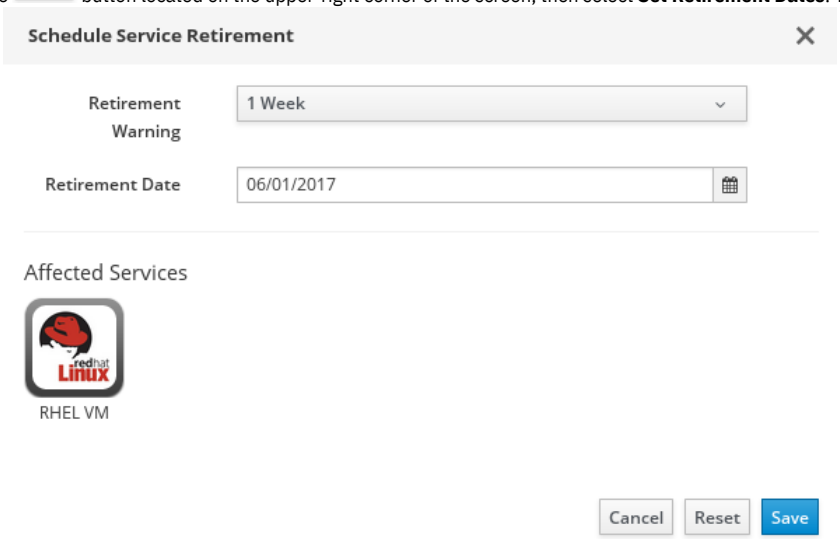
Additionally, you can also control the virtual machine power operations, and manage snapshots from the dashboard. A snapshot is a copy of the virtual machine's disk file (VMDK) that captures the entire state of the virtual machine at a point in time.

Setting a Retirement Date for a Service

Follow this procedure to schedule retirement for a service at a later date. For retiring a service immediately, see Retiring a Service.

1. From the **Services** screen, click on the required service you want to schedule retirement for.

2. Click the button located on the upper-right corner of the screen, then select **Set Retirement Dates**. The **Schedule Service Retirement** window will



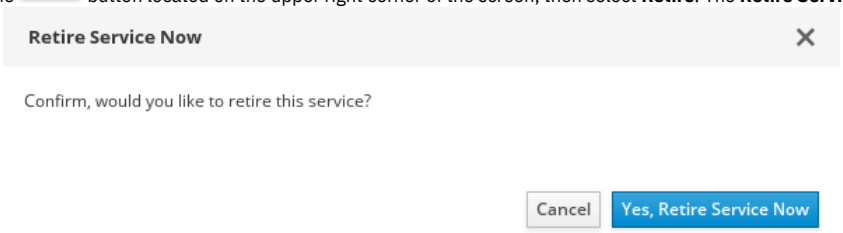
appear.

3. Set the **Retirement Date**.
4. Select the **Retirement Warning** from the list.
5. Click **Save**.

Retiring a Service

Follow this procedure to retire a service immediately. For retiring a service at a later date, see Setting a Retirement Date for a Service.

1. From the **Services** screen, click on the service you want to retire now.
2. Click the button located on the upper right corner of the screen, then select **Retire**. The **Retire Service Now** window will appear to confirm your



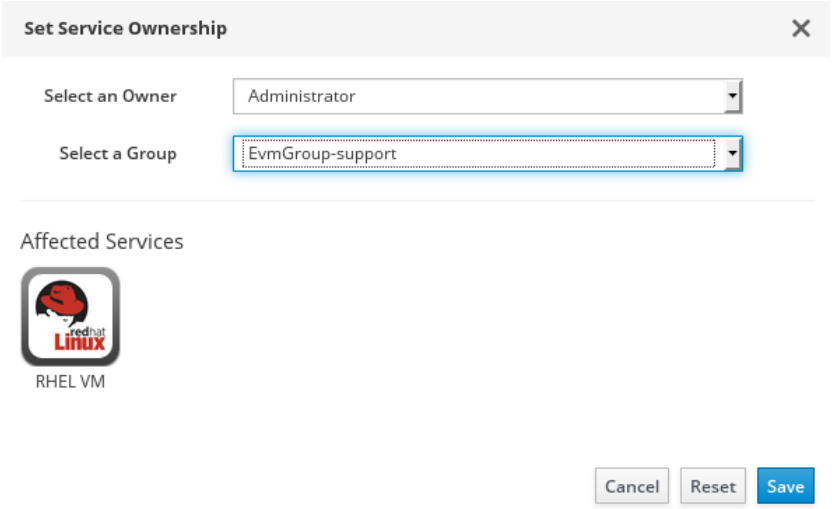
action.

3. Click **Yes, Retire Service Now**.

Setting Ownership of a Service

Follow this procedure to set ownership of a service.

1. From the **Services** screen, click on the more options button next to the service you want set ownership for, then click **Set Ownership**. Alternatively, click on the service. Click the button located on the upper-right corner of the screen, then select **Set Ownership**.

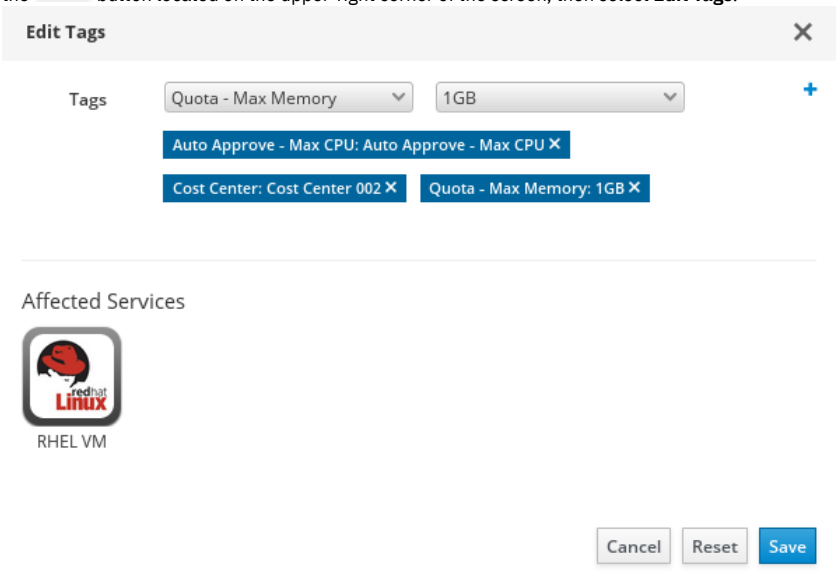


2. Select an owner from the list.
3. Select a group from the list.
4. Click **Save**.

Adding and Editing Tags

Follow this procedure to add or edit tags for a service.

1. From the **Services** screen, click on more options next to the service you want add or edit tags for, then click **Edit Tags**. Alternatively, click on the service. Click the button located on the upper-right corner of the screen, then select **Edit Tags**.



2. To add a tag, select the tag category and value from the lists, then click the blue plus (+) sign as shown in the screen capture.
3. To edit tags, simply change the value assigned to a category, or click on the cross sign in the tag to delete an existing tag.
4. Click **Save**.

Removing a Service

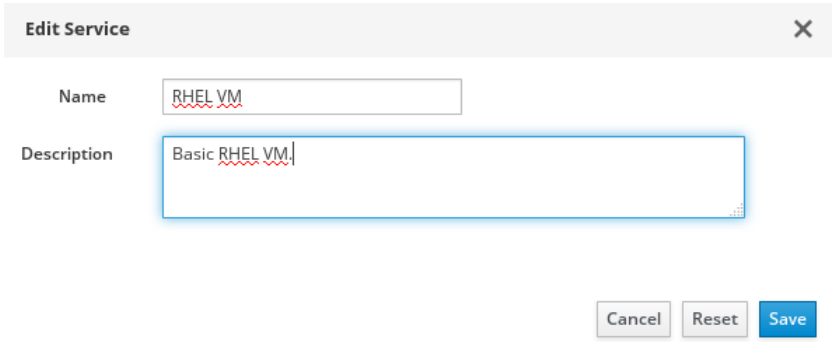
Follow this procedure to remove a service.

1. From the **Services** screen, click on more options next to the service you want to remove, then click **Remove**. Alternatively, click on the service. Click the button located on the upper-right corner of the screen, then select **Remove**. A confirmation window appears.
2. Click **Yes, Remove Service**.

Editing a Service

Follow this procedure to edit the name and description of a service.

1. From the **My Services** screen, click on more options next to the service you want to edit, then click **Edit**. Alternatively, click on the service. Click the button located on the upper-right corner of the screen, then select **Edit**. An **Edit Service** screen appears.



2. Edit the name and description of the service as required.
3. Click **Save**.

Controlling Virtual Machine Power States

Follow this procedure to perform power operations on virtual machines, such as start, stop, or suspend, depending on the current state of the virtual machine.

1. From the **My Services** screen, click on the required service.
2. Under **Resources**, click more options *(represented by three dots)* where you have the option to start, stop, or suspend the virtual machine. Alternatively, click on the virtual machine name to see more details about the resource, including the **Power Operations** button.

Creating Virtual Machine Snapshots

To create a snapshot from the Services screen:

1. From the **My Services** screen, click on the service associated with the virtual machine you want to create a snapshot for.
2. In **Resources**, click **Snapshots** next to the virtual machine you want to create a snapshot for, then select **Create** to create a snapshot. The **Create Snapshot** window appears.
3. Enter a name for the snapshot.
4. Optional: Select **Memory** if you want to capture the state of the virtual machine's memory.
5. Optional: Enter a **Description**.
6. Click **Create**.

To create a snapshot from the Resources screen:

1. From the **My Services** screen, click on the service associated with the virtual machine you want to create a snapshot for.
2. In **Resources**, click on the virtual machine you want to create a snapshot for.
3. On the virtual machine dashboard, click the **Snapshots** button, then select **Create**. The **Create Snapshot** window appears.
4. Enter a name for the snapshot.
5. Optional: Select **Memory** if you want to capture the state of the virtual machine's memory.
6. Optional: Enter a **Description**.
7. Click **Create**.

Viewing Virtual Machine Snapshots

1. From the **My Services** screen, click on the service associated with the virtual machine you want to view snapshots for.
2. In **Resources**, click **Snapshots** next to the virtual machine and select **View** to view available snapshots. Alternatively, click on the virtual machine, then click **Snapshots** to view available snapshots.

Note:

You can revert or delete a snapshot only from the snapshot summary screen. To do so, navigate to **Snapshots > View**, click more options *(represented by three dots)* next to the snapshot you want to revert or delete, and then click **Revert** or **Delete** as required. Revert returns a virtual machine to a previous state when you took the snapshot.

Accessing the Virtual Machine Console

Follow this procedure to open a console window to the virtual machine. The console is the remote control system that enables you to interact directly with your virtual machines.

1. From the **Services** screen, click on the required service.

2. In **Resources**, click the Access button next to the virtual machine and select **VM Console**.

The virtual machine console will open in a new window of your web browser. Note that you may have to enable the pop-up settings in your browser to access the console.

Web Console (Cockpit) Integration

The Web console (Cockpit) is an interactive Linux server administrative interface. You can perform simple administration tasks such as starting containers, storage administration, network configuration, and inspecting logs. While Cockpit allows you to monitor and administer several servers at the same time, your browser connects to a primary server that runs the Cockpit web service through which connections to additional servers can be established.

The Infrastructure Automation full administrative user interface and the Self Service user interface have enabled for users to access the Web console (Cockpit interface) for a given virtual machine.

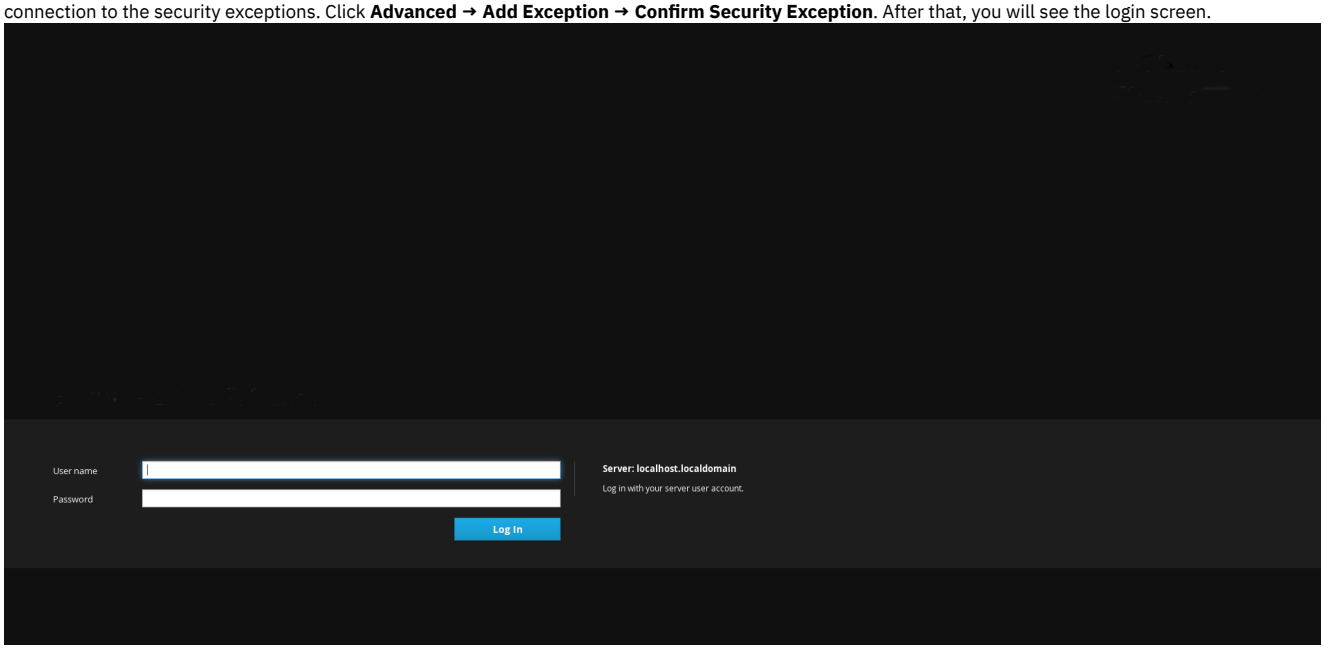
Accessing the Web Console

Note:

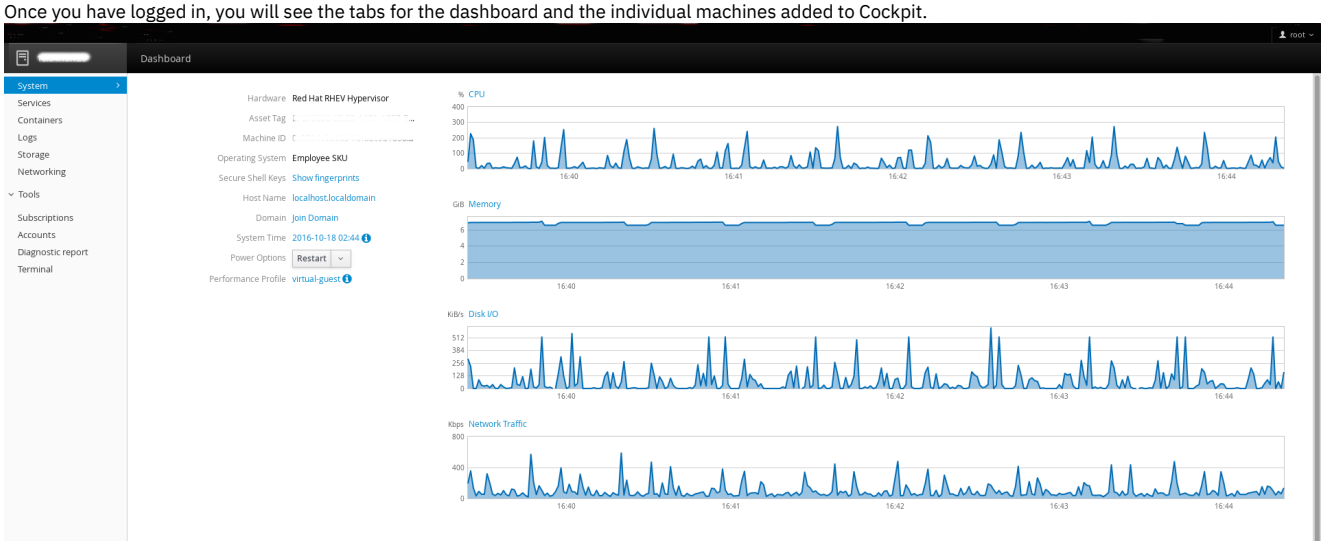
Launching of Cockpit requires as IP address for the virtual machine. It must be pre-configured and running on the virtual machine.

You can open the Web console (Cockpit) for a virtual machine from the Access button.

1. From the **My Services** screen, click on the service associated with the virtual machine you want to open the Web console for.
2. In **Resources**, click **Access** next to the virtual machine and select **Web Console**. A notification will appear to show the Cockpit interface is loading.
3. The Cockpit interface for the virtual machine will open in a new browser window. If you get a security warning by the browser, you will need to add this connection to the security exceptions. Click **Advanced → Add Exception → Confirm Security Exception**. After that, you will see the login screen.



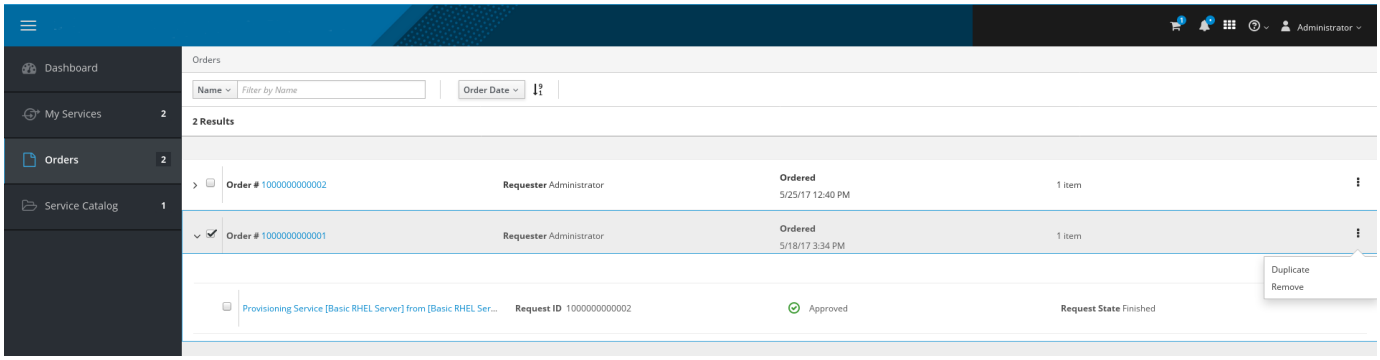
4. Once you have logged in, you will see the tabs for the dashboard and the individual machines added to Cockpit.



My Orders Tab

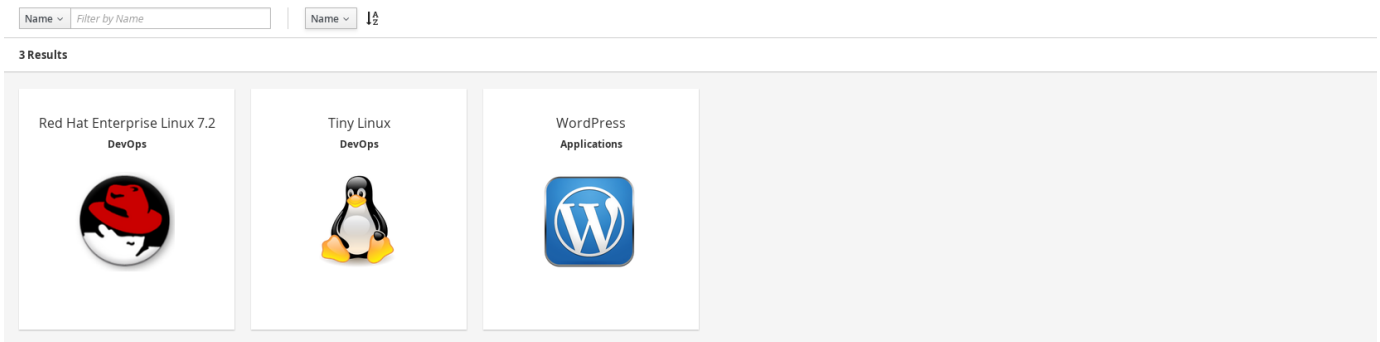
The **My Orders** tab lists all shopping cart orders which contain service requests added from the **Service Catalog**.

Click on the tab to see a summary of each order, including the order number, requester, order date, and the number of items. Click the arrow next to an order number to see more details such as the requests associated with the order, request ID, whether it is approved or denied, and the current state of the request. You can reorder a request by clicking the **Reorder** button, then enter the options associated with the order, and add to shopping cart. You can click on more options *(represented by three dots)* next to an order to **Remove** the order.



Service Catalog Tab

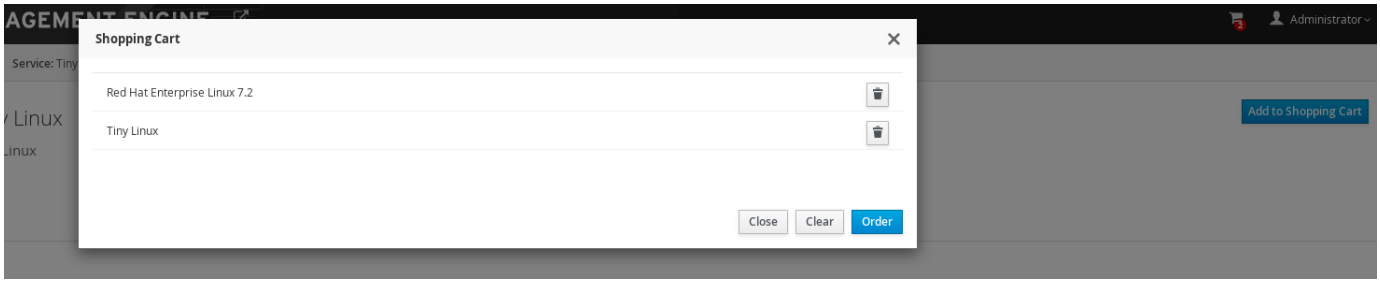
The **Service Catalog** tab lists all service catalog items, displayed as tiles. These items must be configured on the administrative user interface before they will be accessible to the user here. You can filter or sort the service catalog items by **Name**, **Description** and **Catalog Name**.



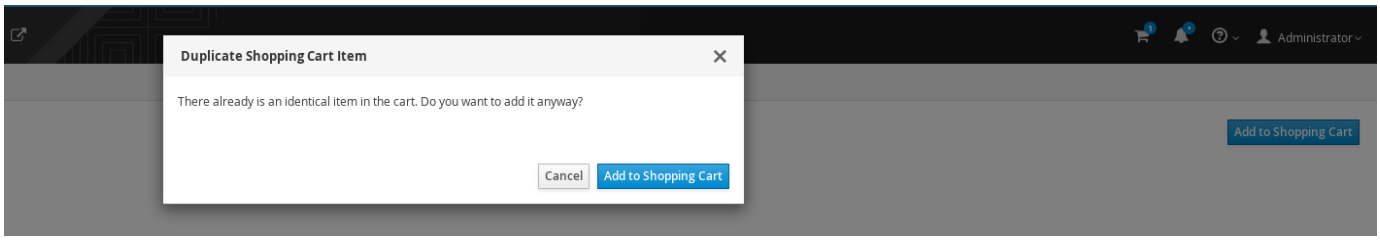
Click on a service to go to the page where you can enter the requested information in the dialog fields under **Service and VM Name** and **VM Characteristics** associated with the service. Note that there is now an ability to add dynamic dialog elements within the Self Service user interface, meaning you can include a particular dialog field where Infrastructure Automation will communicate to the instance and method, obtain the required information, and enter the value in the field dynamically. After you have entered all information in the dialog fields, click **Add to Shopping Cart**.

Shopping Cart:

The Self Service user interface has adopted a shopping cart model for ordering services, instead of ordering them individually. Click on the shopping cart icon at the top to view all services added to the shopping cart that are available to order. You can delete a particular service from the cart if required, and order the services in the cart all at one time. Or, you can even choose to clear all the services in the cart and exit.



If you click **Add to Shopping Cart** for the same item, you will see a pop up window alerting you that there is an identical item in the cart but if you want to add it anyway.



Troubleshooting for Infrastructure Management

Learn how to isolate and resolve problems with Infrastructure Management.

To get support, see Support in the product documentation.

- After upgrading to IBM Cloud Pak for AIOps users cannot access Infrastructure Management
- Error when using multiple tabs in the same browser
- Non-LDAP users cannot access Infrastructure Management
- The vmware-infra-* pods in CrashLoopBackOff error states
- Infrastructure Management vm operator encounters an error creating CRD
- Infrastructure Management vm operator encounters an error when upgrading on OpenShift
- Error when deleting cloud volumes in Infrastructure Management
- Error when logging out of Infrastructure Management
- No new data coming in to IBM Cloud Pak® for AIOps from Infrastructure Management
- **infrastructure-management-zookeeper-0** pod is not Ready on the Red Hat OpenShift on IBM Cloud cluster

After upgrading users cannot access Infrastructure Management

After upgrading, an OIDC provider error can be seen when you access Infrastructure Management.

This error can result in the following error message:

```
OpenID Connect Provider error: Error in handling response type.
```

Solution: To resolve this issue, complete the following steps:

1. Log in to your Red Hat OpenShift cluster:

   ```
   oc login
   ```

2. Change to the Infrastructure Management project:

   ```
   oc project management-infrastructure-management
   ```

3. Save the secret encoded string to a file and use the **base64** command to decode the data and direct the output to another text file.

   ```
   oc get secret -o jsonpath='{.data.oidc\.conf}' ibm-infra-management-application-connection | base64 -d | sed -e
   's#oidc/endpoint/OP/jwk#idprovider/v1/auth/jwk#g'> oidc_new.conf
   ```

   ```
   cat oidc_new.conf | base64 | tr -d '\n'
   ```

4. Replace the value above in the secret **ibm-infra-management-application-connection** in the **oidc.conf**

   ```
   oc edit secret ibm-infra-management-application-connection
   ```

   - Replace oidc.conf with the new value in step 3.
   - Save the secret.

5. Restart the **httpd** pod in the **management-infrastructure-management** namespace to pick up the changes by deleting the **<httpd_pod_instance_name>**. For example,

   ```
   oc delete pod `oc get pod | grep httpd | cut -f1 -d" "`
   ```

Error with multiple tabs in the same browser by using Infrastructure Management

When you are using Infrastructure Management, you can encounter an error starting multiple tabs in the same browser.

You can encounter the following error when you attempt to access Infrastructure Management using multiple tabs in the same browser.

```
Edit aborted! Infrastructure Management does not support the browser's back button or access from multiple tabs or windows
of the same browser. Close any duplicate sessions before proceeding.
```

This error occurs when Infrastructure Management does not support the browser's back button or access from multiple tabs or windows of the same browser.

Solution: Closing the other Infrastructure Management browser tab can resolve the issue.

Non-LDAP users cannot access Infrastructure Management but can see the data in IBM Cloud Pak for AIOps

While you are logged in to IBM Cloud Pak® for AIOps with a non-LDAP user, attempting to start Infrastructure Management can fail with an error.

When this error occurs, an error message can resemble the following example:

```
OpenID Connect Provider error: Error in handling response type.
```

When you attempt to access the Infrastructure Management. console through the side panel, you are sent to **inframgmtinstall.apps.<cluster_name>.<customer_domain>.com** as expected. This link brings you to a login page.

If you log in with an LDAP user, the **initialAdminGroupName** must exist as an LDAP group. Non-LDAP users cannot be used with Single Sign On for Infrastructure Management.

Solution: The initialAdminGroupName that was provided in the IMConfig was not a valid group in the enterprise LDAP directory, instead it was a group manually setup in the Identity and Access Management (IAM) panel. The user that attempted to access Infrastructure Management. using this group was manually added to the group, giving the illusion of a successful integration with the LDAP directory.

The initialAdminGroupName must be a valid group within the enterprise LDAP directory and the designated user must be added to this group. Once the user logs into the IBM Cloud Pak for AIOps console, they will be automatically added to the Identity and Access Management panel.

The vmware-infra-* pods in CrashLoopBackOff error states

You can encounter an error where several pods are crashing, and restarting constantly. These pods can appear to be failing with an authentication error.

When this error occurs, your output when checking the pod status can resemble the following example:

```
NAMESPACE                                                NAME
READY   STATUS              RESTARTS   AGE
management-infrastructure-management                     1-vmware-infra-event-catcher-30-65d7bfd967-tcsrg
0/1     CrashLoopBackOff    399        44h
management-infrastructure-management                     1-vmware-infra-operations-30-6dc9f87b75-mqfpz
0/1     CrashLoopBackOff    402        44h
management-infrastructure-management                     1-vmware-infra-refresh-30-84fcd6cbd4-fc68w
0/1     Error               412        44h
```

When you check the logs, these pods appear to be failing with an authentication error:

```
{"@timestamp":"2020-08-02T21:13:00.050303 ","hostname":"1-vmware-infra-refresh-30-84fcd6cbd4-
fc68w","pid":7,"tid":"2ae9ee5f997c","level":"err","message":"MIQ(ManageIQ::Providers::Vmware::InfraManager::RefreshWorker:
:Runner) ID [2241] PID [7] GUID [403658a2-96a2-4823-b996-94cd6472dffd] EMS id [30] failed authentication check. Worker
exiting."}
```

This error can happen when VMWare is unable to be accessed, which can be due to an authentication issue, network issue, or some other reason. When this connection issue occurs, the pods enter this state.

Solution: To resolve this error, complete the following steps:

1. Fix the credentials or network issue, allowing the pods to recover.
2. Delete the provider to stop the CrashLoopBackOff.

Infrastructure Management vm operator encounters an error when creating CRD

After you install Infrastructure Management, you can encounter an error where no **deployers** are in the **management-infrastructure-management** namespace.

```
oc get deployers.core.hybridapp.io --all-namespaces
```

Output:

```
No resources found
```

This error can occur when the operator attempts to create the CR before the CRD is installed. To check whether this issue is occurring, run the following command:

```
oc logs -n management-infrastructure-management -f infra-management-vm-operator-764ff5fff9-zm81f
```

Output:

```
I0729 23:17:27.105548       1 deployer.go:81] Create Deployer: error creating CRD, the server could not find the requested
resource
```

Solution: Restart the **infra-management-vm-operator** pod. When the pod is restarted run the following command to view the result:

```
oc get deployers.core.hybridapp.io --all-namespaces
```

Output:

```
NAMESPACE                                 NAME       AGE
management-infrastructure-management   ibminfra   12s
```

Infrastructure Management vm operator encounters an error when upgrading on OpenShift

When you are upgrading Infrastructure Management, you can encounter an error where the operator fails to install correctly.

Checking the OpenShift console InstalledOperators for project: **management-infrastructure-management** shows the operator in "Failed" state. Checking **oc describe** shows a message "conflicting CRD owner in namespace". For example,

```
oc describe operator ibm-management-infra-vm.management-infrastructure-management
```

```
Name:         ibm-management-infra-vm.management-infrastructure-management
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  operators.coreos.com/v1
Kind:         Operator
Metadata:
  ...
  ...
      Namespace:               management-infrastructure-management
      API Version:             operators.coreos.com/v1alpha1
      Conditions:
        Last Transition Time:  2020-11-23T23:19:46Z
        Last Update Time:      2020-11-23T23:19:46Z
        Message:               installing: missing deployment with name=infra-management-vm-operator
        Reason:                NeedsReinstall
        Status:                True
        Type:                  Pending
      Kind:                    ClusterServiceVersion
      Name:                    ibm-management-infra-vm.v2.2.3-202011230408
      Namespace:               management-infrastructure-management
      API Version:             operators.coreos.com/v1alpha1
      Conditions:
        Last Transition Time:  2020-11-23T23:19:44Z
        Last Update Time:      2020-11-23T23:19:44Z
        Message:               conflicting CRD owner in namespace
        Reason:                OwnerConflict
        Status:                True
        Type:                  Failed
      Kind:                    ClusterServiceVersion
      Name:                    ibm-management-infra-vm.v2.2.3-202011231858
      Namespace:               management-infrastructure-management
Events:                        <none>
```

This error CAN be seen when conflicting CSVs exist:

```
oc get csv -n management-infrastructure-management
```

Output:

```
NAME                                                DISPLAY                                                VERSION
REPLACES                                            PHASE
ibm-management-infra-vm.v2.2.3-202011231109         IBM Management Infrastructure for virtual machines   2.2.3-
202011231109   ibm-management-infra-vm.v2.2.3-202011230859          Replacing
ibm-management-infra-vm.v2.2.5-202012021409         IBM Management Infrastructure for virtual machines   2.2.5-
202012021409   ibm-management-infra-vm.v2.2.3-202011231109          Succeeded
```

Solution: Delete any older CSVs and this will allow the operator to install.

```
oc delete csv -n management-infrastructure-management ibm-management-infra-vm.v2.2.3-202011231109
```

```
oc delete csv -n management-infrastructure-management ibm-management-infra-vm.v2.2.3-202011230859
```

Error when deleting cloud volumes in Infrastructure Management

After you create a cloud volume, you might be unable to detach the volume from an instance. When this issue occurs, you can view an error if you then try to delete the volume. The error can resemble the following message:

```
Unable to detach Cloud Volume "volume_name" from "instance_name" Instance "instance_ID" is not attached to volume
"volume_ID".
```

This error can occur when the refresh (EmsRefresh) has not run or completed. to resolve this issue, manually run a refresh by browsing to **Configuration** > **Refresh Relationships and Power States**. Then, wait for the refresh to complete. When the refresh is completed, delete or detach the volume again.

Errors logging out of or re-logging in to Infrastructure Management

While logged in to IBM Cloud Pak® for AIOps and the IBM Cloud Pak for AIOps console you log out of Infrastructure Management instance and you receive an error. The error states:

```
This page isn't working right now.  HTTP ERROR 400
```

Logging out of Infrastructure Management takes you back to the login for IBM Cloud Pak for AIOps. When attempting to start the IBM Cloud Pak for AIOps console it fails to log you back in through SSO and you are prompted to enter login credentials.

You can log out of IBM Cloud Pak for AIOps, then log back in to IBM Cloud Pak for AIOps and start the IBM Cloud Pak for AIOps console.

No new data coming in to IBM Cloud Pak® for AIOps from Infrastructure Management

You can encounter an issue where providers are added within Infrastructure Management, but are not showing IBM Cloud Pak for AIOps.

This issue can occur when the access token that provides authorization into Infrastructure Management from IBM Cloud Pak for AIOps is expired.

To confirm this cause, check the BypassAuth annotation in the **`imconnection connections.infra.management.ibm.com`** CR in the **`management-infrastructure-management`** project or namespace.

```
metadata:
  annotations:
    BypassAuth: 'false'
```

If the BypassAuth annotation is not defined or is set to "false" then:

- check if spec.cfHost is defined with the correct Infrastructure Management url.
- check for the definition of spec.secrets.accessToken.

```
spec:
  cfHost: 'inframgmtinstall.apps.cp4mcm-23.company.com'
  secrets:
    accessToken:
      secretKeyRef:
        key: accesstoken
        name: ibmconnectionsecret
```

If accessToken is defined then check the log of the VM operator and look for entries with "TokenRefresh Token has expired" similar to the following:

```
I0401 13:59:35.553292       1 update.go:34] TokenRefresh Starting update of token.
I0401 13:59:35.555424       1 secret.go:166] TokenRefresh Getting secrets from management-infrastructure-management using
name imconnectionsecret
I0401 13:59:35.571513       1 secret.go:173] TokenRefresh got token secret 131462371
I0401 13:59:35.571540       1 update.go:94] TokenRefresh checking if expired.
W0401 13:59:35.571553       1 secret.go:266] TokenRefresh Token has expired. Current time: 2021-04-01 13:59:35.571549714
+0000 UTC m=+29212.072337265, End time: 2021-03-26 18:52:49 +0000 UTC
E0401 13:59:35.571570       1 update.go:65] refresh token has elapsed not attempting refresh
E0401 13:59:35.571615       1 update.go:27] error getting token using refresh token
I0401 13:59:35.571622       1 configs.go:92] Sleeping for 300.000000 seconds.
```

Solution

If the accessToken has expired then update the secret referenced in **`accessToken.secretKeyRef.name`**.

Note: For version 2.3 upgraded from 2.2, the default secret name is **`imconnectionsecret`**. For a new install of version 2.3 the default secret name is **`ibm-infra-management-application-connection`**.

To fix the problem the oauth username and password can be added to the **`<secret_name>`** in order for the tokenrefresh process to request a new accesstoken from IAM. Once the accesstoken is obtained the oauth username and password are removed from the secret.

Edit the secret using the command:

```
oc edit secret <secret_name> -n management-infrastructure-management -o YAML
```

Add after the line **`type: Opaque`**

```
type: Opaque
stringData:
  oauth_username: <user_name>
  oauth_password: <password>
```

Where **`<secret_name>`** is **`ibm-infra-management-application-connection`**

If the BypassAuth annotation is set to "true" then check to make sure spec.cfHost in the **`imconnection`** CR is set to the following:

```
spec:
  cfHost: 'web-service.management-infrastructure-management.svc.cluster.local:3000'
```

Verifying the solution

Check the log for the **sre-inventory-inventory-cfcollector** pod in the **kube-system** project for entries similar to the following:

```
I0425 00:13:35.835415       7 main.go:41] ============Do synchronization from CloudForm http://web-service.management-
infrastructure-management.svc.cluster.local:3000
I0425 00:13:39.922796       7 query_vms.go:147] Totally 333 vms found in cloudform. 333 VMs should be returned and
actually 333 returned in this query.
I0425 00:13:40.029993       7 query_providers.go:78] Totally 4 providers found in cloudform. 4 providers should be
returned and actually 4 returned in this query.
I0425 00:13:40.114010       7 query_availabilityzone.go:77] Totally 0 availability_zones found in cloudform. 0
availability_zones should be returned and actually 0 returned in this query.
I0425 00:13:40.309270       7 query_hosts.go:112] Totally 31 hosts found in cloudform. 31 hosts should be returned and
actually 31 returned in this query.
I0425 00:13:40.387620       7 query_services.go:84] Totally 0 services found in cloudform. 0 services should be returned
and actually 0 returned in this query.
I0425 00:13:40.470200       7 query_cloud_volumes.go:104] Totally 0 cloud volumes found in cloudform. 0 cloud volumes
should be returned and actually 0 returned in this query.
I0425 00:13:40.627044       7 query_data_stores.go:105] Totally 43 data stores found in cloudform. 43 data stores should
be returned and actually 43 returned in this query.
I0425 00:13:41.366165       7 query_cloud_networks.go:101] Totally 112 cloud networks found in cloudform. 112 cloud
networks should be returned and actually 112 returned in this query.
I0425 00:13:42.101278       7 query_cloud_subnets.go:97] Totally 342 cloud subnets found in cloudform. 342 cloud subnets
should be returned and actually 342 returned in this query.
I0425 00:13:42.176657       7 query_network_routers.go:92] Totally 0 network router found in cloudform. 0 network routers
should be returned and actually 0 returned in this query.
I0425 00:13:42.451049       7 query_security_groups.go:95] Totally 112 security groups found in cloudform. 112 security
groups should be returned and actually 112 returned in this query.
I0425 00:13:42.959106       7 query_floating_ips.go:96] Totally 287 floating ips found in cloudform. 287 floating ips
should be returned and actually 287 returned in this query.
I0425 00:13:42.961653       7 sender.go:206] Sending Resources { request: 99999, add: 853, update: 147, delete: 0, edgedd:
0, edgeDelete:0}
I0425 00:13:43.427390       7 sender.go:206] Sending Resources { request: 99999, add: 0, update: 200, delete: 0, edgedd:
800, edgeDelete:0}
I0425 00:13:46.473678       7 sender.go:206] Sending Resources { request: 99999, add: 0, update: 0, delete: 0, edgedd: 53,
edgeDelete:0}
I0425 00:13:46.683429       7 main.go:43] ============Done synchronization this time
```

infrastructure-management-zookeeper-0 pod is not Ready on the Red Hat OpenShift on IBM Cloud cluster

You might notice that the **infrastructure-management-zookeeper** pod is not in *Ready* status on Red Hat OpenShift on IBM Cloud cluster.

The issue might happen because **infrastructure-management-zookeeper** pod requires more cpu and memory to start on a Red Hat OpenShift on IBM Cloud cluster.

Solution: Use the following steps to fix the failed pod issue:

1. Change to the Infrastructure Automation namespace.

   ```
   oc project <IANamespace>
   ```

   Where **IANamespace**is the namespace where Infrastructure Automation is installed.

   Note: If IBM Cloud Pak for AIOps is also installed, then use the IBM Cloud Pak for AIOps namespace in the command.

2. Get the name of iminstall custom resource:

   ```
   oc get iminstall --no-headers | awk '{print $2}'
   ```

3. Make a copy of the existing iminstall custom resource by using the command:

   ```
   oc get iminstall <iminstall CR name> -o yaml > <iminstall CR name>.orig
   ```

   Where **<iminstall CR name>** is the name of iminstall custom resource.

4. Delete the iminstall custom resource:

   ```
   oc delete iminstall <iminstall CR name>
   ```

   Where **<iminstall CR name>** is the name of iminstall custom resource.

5. Edit the iminstall custom resource file by using the command:

   ```
   vi <iminstall CR name>.orig
   ```

   Where **<iminstall CR name>** is the name of iminstall custom resource.

6. Remove the following lines from iminstall custom resource:

   ```
   finalizers:
      - infra.management.install.finalizer.infra.management.ibm.com
   resourceVersion: "3023054"
   uid: c790e781-07a3-46d7-bf20-5e1acc1ba2ad
   ```

Note: You might have different values for the mentioned properties.

7. Add the following variables to the iminstall custom resource:

```
spec:
  zookeeperCpuRequest: 400m
  zookeeperCpulimit: 500m
  zookeeperMemoryRequest: 400Mi
```

Note:if you are using IAConfig to create the iminstall custom resource then, you can add the following environment variables to the IAConfig custom resource.

```
name: ibm-management-im-install
spec:
  iminstall:
    applicationDomain: inframgmtinstall.<your_env_domain>
    imagePullPolicy: Always
    imagePullSecret: ibm-entitlement-key
    initialAdminGroupName: <your_admin_group>
    zookeeperCpuRequest: 400m
    zookeeperCpulimit: 500m
    zookeeperMemoryRequest: 400Mi
    license:
      accept: true
```

8. Apply the iminstall custom resource by using the command:

```
oc apply -f <iminstall CR name>.orig
```

Where **<iminstall CR name>** is the name of iminstall custom resource.

Managed services

By using the Managed services, you can automate provisioning of infrastructure and virtual machine applications across multiple cloud environments with optional workflow orchestration.

The Managed services was previously called as Terraform & Service Automation or IBM Cloud Automation Manager. For documentation for previous versions of IBM Cloud Automation Manager, see the IBM Cloud Automation Manager Documentation.

Managed services is a multi-cloud, self-service management platform running on IBM Cloud Pak® for AIOps that empowers developers and administrators to meet business demands.

Service Composer from the Managed services allows you to expose cloud services to any cloud through the Service library. It accelerates developer velocity and improves governance.

Managed services uses open source Terraform to manage and deliver cloud infrastructure as code. Cloud infrastructure delivered as code is reusable, able to be placed under version control, shared across distributed teams, and it can be used to easily replicate environments. Managed services brings a common approach to automate all workloads in all clouds to reduce configuration errors and improve operational efficiency. You can manage infrastructure as code using HashiCorp Terraform software.

The Managed services content library comes pre-populated with sample templates to help you get started quickly. It allows you to leverage IBM, community, and home-grown assets as building blocks and modernize legacy applications and build new cloud native applications. Use the sample templates as is or customize them as needed. A Chef runtime environment can also be deployed using Managed services for more advanced application configuration and deployment.

With Managed services, you can provision cloud infrastructure and accelerate application delivery into multiple cloud environments with a single user experience.

You can spend more time building applications and less time building environments when cloud infrastructure is delivered with automation. You are able to get started fast with pre-built infrastructure from the Managed services library.

To access the Managed services template and Service instances, log in to IBM Cloud Pak® for AIOps console and click **Automate Infrastructure** > **Instances**.

To access the Managed services template and Service library, log in to IBM Cloud Pak® for AIOps console and click **Automate Infrastructure** > **Library**.

To access the Managed services connection and configuration section, log in to IBM Cloud Pak® for AIOps console and click **Automate Infrastructure** > **Manage**.

To access the Service catalog, log in to IBM Cloud Pak® for AIOps console and click **Automate Infrastructure** > **Service catalog**.

To deploy infrastructure, you can orchestrate a flow with a wide variety of modules and integrations.

- Creating and managing services
- Creating and managing service instances
- Creating and managing custom actions
- Managed services Content

Uninstalling Infrastructure Automation

1. If you deployed Infrastructure Automation - Infrastructure Management, delete the deployed *IMInstall* instance.
    1. Log in to your Red Hat OpenShift Container Platform cluster's console.

2. From the main navigation menu, click **Operators** > **Installed Operators**.
3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
4. Select the **IBM Management Infrastructure install** tab. The *IMInstall* instance is displayed.
5. Click the **IMInstall** under provided APIs column
6. Select **Delete IMInstall** in the Node Options menu for the IMInstall instance.
7. Wait for the *IMInstall* instance to be deleted.

2. Uninstall ManageService instance from Infrastructure Automation - Managed services on the Red Hat OpenShift Container Platform console.

    1. Log in to your Red Hat OpenShift Container Platform cluster's console.
    2. Click **Operators > Installed Operators.**
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select the **IBM Management Infrastructure managed services** tab. The *ManageService* instance is displayed.
    5. Click the **ManageService** under provided APIs column
    6. Select **Delete ManageService** in the Node Options menu for the ManageService instance.
    7. Wait for the *ManageService* instance to be deleted.

3. Uninstall the *IAConfig* instance from the Infrastructure Automation operator.

    1. Log in to your Red Hat OpenShift Container Platform cluster's console.
    2. Click **Operators > Installed Operators.**
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select the **IBM Infrastructure Automation** tab. The **IBM Infrastructure Automation** instance is displayed. Click **IBM Infrastructure Automation** in the **provided APIs** column.
    5. Select **Delete IAConfig** in the Node Options menu for the IAConfig instance.
    6. Wait for the Managed services and Infrastructure Management non-operator pods to be removed. To view the remaining pods, run the following command:

        ```
        oc -n <namespace> get pods
        ```

        Where **<namespace>** is the project (namespace) where Infrastructure Automation is installed.

        - Managed services pods are prefixed with **cam**.
        - Infrastructure Management pods can be prefixed with any of the following prefixes: **1-**, **httpd**, **memcached**, **orchestrator**, **postgresql**, **ibm-infra-management**.

        Make sure that all the Managed services and Infrastructure Management pods are removed before you move on to the next step.

4. Uninstall the Infrastructure Automation operator.

    1. Log in to the Red Hat OpenShift Container Platform console for your environment.
    2. From the main navigation menu, click **Operators** > **Installed Operators**.
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select **Uninstall Operator** from the options menu for the **IBM Infrastructure Automation** operator.

5. If you installed only Infrastructure Automation and not IBM Cloud Pak® for AIOps in the Infrastructure Automation namespace, then uninstall the following operators in the installed namespace.

    1. Log in to the Red Hat OpenShift Container Platform console for your environment.
    2. From the main navigation menu, click **Operators** > **Installed Operators**.
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select **Uninstall Operator** from the options menu for the **IBM CloudPak functional services** operator.

6. Delete the namespace that Infrastructure Automation is deployed in.

    1. Run the following command to remove finalizers from **clients.oidc.security.ibm.com**.

        ```
        oc -n <namespace> edit clients.oidc.security.ibm.com
        ```

        Where **<namespace>** is the namespace that Infrastructure Automation is deployed in.

        Delete the following finalizer lines, and save.

        ```
        finalizers:
          - client.oidc.security.ibm.com
        ```

    2. Run the following command to remove finalizers from **operandrequests.operator.ibm.com**.

        ```
        oc -n <namespace> edit operandrequests.operator.ibm.com
        ```

        Where **<namespace>** is the namespace that Infrastructure Automation is deployed in.

        Delete all the finalizer lines, and save.

        ```
        finalizers:
           - finalizer.request.ibm.com
        ```

    3. Delete the namespace that Infrastructure Automation is deployed in.

        ```
        oc delete project <namespace>
        ```

        Where **<namespace>** is the namespace that Infrastructure Automation is deployed in.

7. (Optional) Delete the Cert Manager

If you installed the IBM Cloud Pak® foundational services Cert Manager, and no other IBM Cloud Paks® are using it, then you can remove it by following the instructions in Manual steps for uninstalling singleton services - Cert Manager in the IBM Cloud Pak foundational services documentation.

8. (Optional) Delete the License Service

If you installed the IBM Cloud Pak foundational services License Service, and no other IBM Cloud Paks® are using it, then you can remove it by following the instructions in Uninstalling License Service in the IBM Cloud Pak foundational services documentation.

9. Uninstall IBM Cloud Pak foundational services

If IBM Cloud Pak for AIOps is not using IBM Cloud Pak foundational services in the Infrastructure Automation namespace, then you can remove IBM Cloud Pak foundational services by following the instructions in Uninstalling IBM Cloud Pak foundational services in the IBM Cloud Pak foundational services documentation.

If the uninstallation fails, or is not complete and is not progressing, then see Troubleshooting installing and uninstalling Infrastructure Automation to help you identify any problems.