IBM Cloud Orchestrator
Version 2.5.0.10

User's Guide

Note

Before you use this information and the product it supports, read the information in "Notices" on page 397.

This edition applies to version 2, release 5, fix pack 10 of IBM® Cloud Orchestrator (program number 5725-H28) and to all subsequent releases and modifications until otherwise indicated in new editions.

The material in this document is an excerpt from the IBM Cloud Orchestrator knowledge center and is provided for convenience. This document should be used in conjunction with the knowledge center.

Contents

Preface

This publication documents how to use IBM Cloud Orchestrator.

Who should read this information

This information is intended for cloud administrators who install and configure IBM Cloud Orchestrator, and for users who work with this product.

Chapter 1. Overview

With IBM Cloud Orchestrator, you can manage your cloud infrastructure.

IBM Cloud Orchestrator helps you with end-to-end service deployment across infrastructure and platform layers. It also provides integrated IT workflow capabilities for process automation and IT governance, resource monitoring, and cost management. The product offers you an extensible approach to integration with existing environments such as network management tools. It facilitates integration with customer-specific service management processes, such as those defined in the IT infrastructure library (ITIL).

Using IBM Cloud Orchestrator, you have a consistent, flexible, and automated way of integrating the cloud with customer data center policies, processes, and infrastructures across various IT domains. Use the intuitive, graphical tool in IBM Cloud Orchestrator to define, and implement business rules and IT policies. You can connect the aspects of different domains into a consistent orchestration of automated and manual tasks to achieve your business goals.

You choose between two editions: IBM Cloud Orchestrator and IBM Cloud Orchestrator Enterprise Edition which also includes Monitoring and Cost Management.

What is new

The following enhancements were introduced in the current release.

Effective Oct 14, 2019 IBM has announced the following changes in the entitlement of BigFix, which HCL acquired in June 2019:

To continue using BigFix for Patching, you must purchase BigFix from HCL or an authorized BigFix reseller. A replay of HCL's BigFix roadmap can be found here: https://register.gotowebinar.com/register/5699647677344055565. Please contact Mark Phinick at HCL for BigFix licensing information and pricing. Mark's email is mark.phinick@hcl.com. This entitlement change applies to the supported IBM Cloud Orchestrator versions 2.4 and 2.5, and is also applicable to the Standard Edition and Enterprise Edition variants of IBM Cloud Orchestrator.

Integration with Ansible Tower
For more information about the Ansible toolkit, see enablement/ansible/ansible_introduction.dita.

Delete requests history
You can remove requests from the history of IBM Cloud Orchestrator. For more information, see "Deleting request history" on page 211.

Currency support for IBM Cloud Orchestrator:

Table 1.

IBM Cloud Orchestrator components	Versions included
IBM Cloud Manager with OpenStack	4.3 FP 14
Red Hat Enterprise Linux	7.6
IBM HTTP Server	8.5.5.16
vCenter	6.7 U1b
NSX	6.4.5
IBM Java JDK	8.0.5.40
WebSphere Application Server	8.5.5.16
PowerVC	1.4.3

Note: The currency support that is listed is specific to new inclusions in V2.5.0.10. For more information about software requirements, see "Checking the software prerequisites" on page 22.

For supported versions of Data Protection and Recovery, Databases and Process Management tools, see the **Prerequisites** tab of https://www.ibm.com/software/reports/compatibility/clarity-reports/report/html/softwareReqsForProduct?deliverableId=1386584144400.

For the software prerequisites of the OpenStack environment, check either the IBM Cloud Manager with OpenStack prerequisites (at http://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liacasoftware.html) or refer to the documentation of your vendor-specific OpenStack distribution.

Currency support for IBM SmartCloud® Cost Management 2106 ifix07:

Table 2.

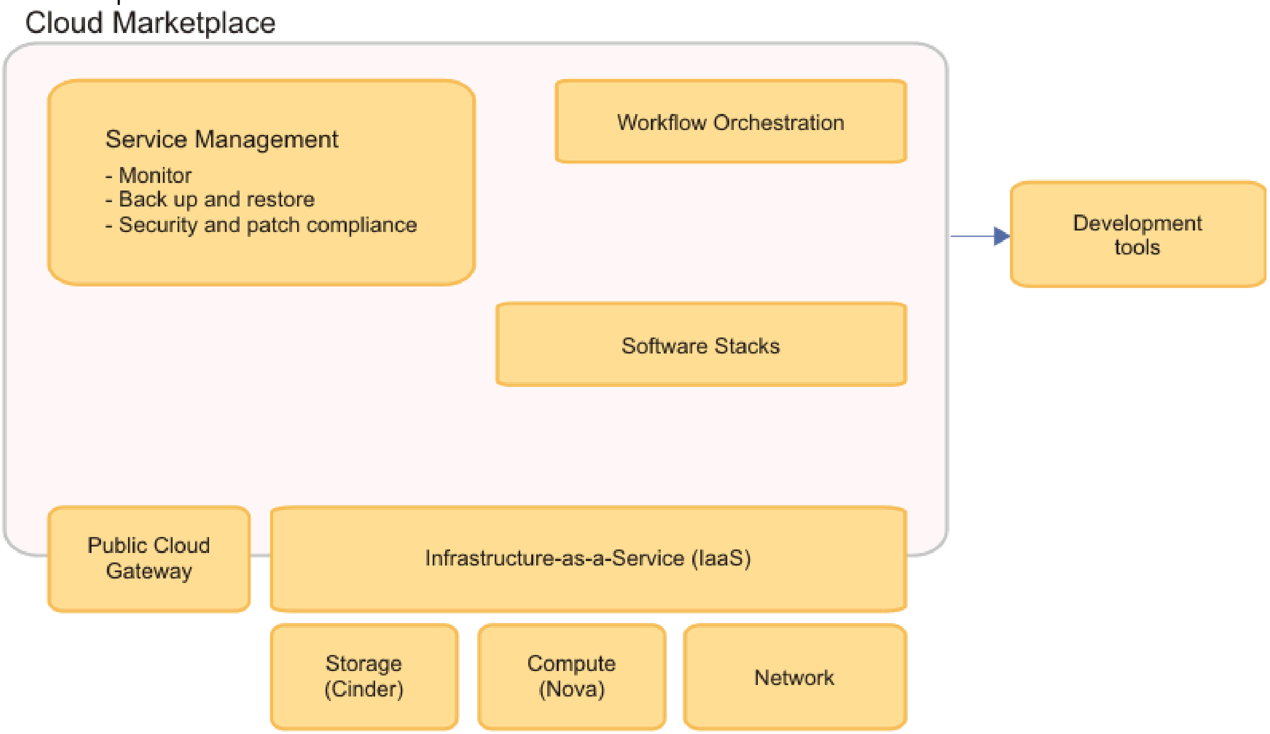
IBM SmartCloud Cost Management	Versions included
IBM Java JDK	8.0.5.40

Product architecture

IBM Cloud Orchestrator is a comprehensive product that integrates the capabilities of several other IBM solutions.

IBM Cloud Orchestrator provides seamless integration of private and public cloud environments. IBM Cloud Orchestrator is the ideal solution for IT organizations that want to implement a hybrid cloud delivery model. It automates the complete delivery of IT services on private cloud environments, and it enables the exploitation of the same services on resources running on public clouds such as Amazon EC2 or IBM SoftLayer, by using the Public Cloud Gateway component, or such as Microsoft Azure.

The main components of IBM Cloud Orchestrator are the process engine and the corresponding modeling user interface, which is used to create processes. For this purpose, IBM Cloud Orchestrator uses the capabilities of IBM Business Process Manager. It also integrates other domain-specific components that are responsible for such functions as monitoring, metering, and accounting. IBM Cloud Orchestrator bundles all these products and components and provides processes that are required to implement the domain-specific functions.



The following are the descriptions of the role each major component plays in IBM Cloud Orchestrator:

Infrastructure-as-a-Service
The Infrastructure-as-a-Service (IaaS) component is responsible for managing access to compute, storage, and networking resources in the virtual environment. All requests to provision services across these services are performed by this component. The IaaS component is delivered by using OpenStack, a leading open source, community-driven project for highly scalable, highly resilient cloud infrastructure management. IBM is one of the Platinum Members of the OpenStack Foundation.

Software Stacks
While not a specific component itself, Software Stacks represent the concept that when one or more virtual systems are deployed, it is also possible to specify multiple software packages to be deployed upon first boot of those systems. It can be done by starting simple installation scripts, but also other strong tools can be used such as Chef recipes and cookbooks for automated installation and configuration.

Workflow Orchestration
The Workflow Orchestration component provides a graphical editor that allows the user to easily customize and extend the procedures that are followed when a user request is initiated. In addition, it also provides the facilities to customize the self-service catalog so that users have access to various service request types that they can access. This component is delivered by embedding IBM's award-winning Business Process Manager technology along with a number of pre-built automation toolkits that make it possible to integrate workflow automation with the cloud platform and its other components. The graphical designer is highly flexible, providing many integration techniques ranging from invocation of simple scripts and calling web services to starting more sophisticated programs such as those written in Java™.

IBM Cloud Orchestrator Catalog
The IBM Cloud Orchestrator Catalog is a publicly accessible website where various forms of automation can be downloaded and used within IBM Cloud Orchestrator. It includes references to supported automation communities such as Chef, and various pre-built Workflow Orchestration routines, packages, and toolkits. It is designed to "ship when ready", meaning that new automation can become available at any time, regardless of IBM Cloud Orchestrator release schedules.

Public Cloud Gateway
The Public Cloud Gateway component is responsible for the integration with public clouds enabling the complete IT services delivery on resources running on public clouds such as Amazon EC2 or IBM SoftLayer.

Service Management
This box represents optional extra management functions that are included in IBM Cloud Orchestrator Enterprise Edition. It also highlights the ability to integrate through Workflow Orchestration other management tools and disciplines that might be important within your environment.

Development tools
This box represents the ability to integrate developer tools from IBM Rational Team Concert and a set of plug-ins within Cloud Continuous Delivery such as that a user can automate a "continuous delivery pipeline" from check-in of code, through build, deployment, test, and promotion. Those tools are not provided within IBM Cloud Orchestrator, but more information about them can be found on ibm.com®.

Product features and components

Read about the main features and components of IBM Cloud Orchestrator.

Managing OpenStack Heat

OpenStack Heat templates are suitable for scenarios that focus on the infrastructure. You can create resources such as instances, networks, volumes, security groups, and users, and define the relationships between these resources (for example, a volume must be attached to a specific instance, some instances are to be connected by using this network). It allows the addition of autoscaling services that are integrated with OpenStack Ceilometer. Images to be deployed through Heat templates require that cloud-init to be installed. For more information, see "Working with Heat templates and stacks" on page 222 and "Managing Heat templates" on page 232.

Supporting private, public, and hybrid cloud environments

IBM Cloud Orchestrator is a private cloud offering that significantly simplifies the task of managing an enterprise-grade cloud. You use a core set of open-source-based technologies to build enterprise-class cloud services that can be ported across hybrid cloud environments. You can use the Public Cloud Gateway to communicate with Amazon EC2 or IBM SoftLayer. For more information, see "Using the Public Cloud Gateway" on page 249.

The supported hypervisors are:

- In Heat: KVM, VMware, PowerVC, PowerKVM, z/VM®, and Hyper-V.
- In Hybrid: Amazon EC2, IBM SoftLayer, and Microsoft Azure.

Designing business processes

IBM Cloud Orchestrator is integrated with IBM Business Process Manager version 8.5.7, a workflow engine with graphical tools. You can extend the capabilities of IBM Cloud Orchestrator by using the simple drag-and-drop technology in Business Process Manager to create and edit complex workflows: you can design, run, monitor, and optimize business processes. For more information, see "Custom extensions" on page 7.

Promoting open source technology

IBM Cloud Orchestrator works with the open source OpenStack software (the *Kilo*, *Mitaka*, *Ocata*, and *Queens* releases). OpenStack is a collection of open source technologies that provide scalable computing software for both public and private clouds. For detailed information about OpenStack, see the OpenStack documentation. For more information, see "Overview of OpenStack" on page 5.

Administering the cloud infrastructure

Administrators can use the OpenStack Dashboard, which is extended by IBM Cloud Orchestrator, to easily manage and monitor the cloud infrastructure. Administrators can define networks and flavors; inspect the actual resource consumption; and manage users, roles, and projects. For more information about using the OpenStack Dashboard, see "Administering as cloud administrator" on page 172.

Customizing the Self-Service Catalog

The intuitive Self-service user interface provides users with a customizable catalog of offerings. The offerings can be grouped into categories that are created by administrators to suit the needs of the work environment. For more information about self-service, see "Designing self-service" on page 228.

Storing persistent data

An IBM DB2® version 10.5.0.10 database is used to store all the IBM Cloud Orchestrator persistent data. Business Process Manager uses this database. A DB2 instance is also used to store installation and configuration data.

Ensuring high availability

High availability is provided by introducing redundancy and improved recovery for core software components of the IBM Cloud Orchestrator management stack that is managed by using Tivoli® System Automation for Multiplatforms.

Note: IBM Cloud Orchestrator does *not* provide high availability for the OpenStack components. For information about high availability in IBM Cloud Manager with OpenStack, see the IBM Cloud Manager with OpenStack documentation.

Managing cost

The IBM SmartCloud Cost Management component of the Enterprise Edition provides functions for collecting, analyzing, reporting, and billing that is based on usage and costs of shared computing resources. With this tool, you can understand your costs and track, allocate, and invoice based on allocated or actual resource use by department, user, and many more criteria. For more information about cost management, see *Metering and billing of SCCM User's Guide*.

Within IBM Cloud Orchestrator, metering is primarily driven from the OpenStack layer to capture all virtual machine provisioning requests.

For more information about monitoring, see *OpenStack data collector section of SCCM User's Guide* topic.

Monitoring
In the Enterprise Edition of IBM Cloud Orchestrator, you can monitor workloads and instances by using IBM Tivoli Monitoring. With this component, you can measure the cost of cloud services with metering and charge-back capabilities. For more information about monitoring, see "Integrating with IBM Tivoli Monitoring" on page 291.

Overview of OpenStack

IBM Cloud Orchestrator works with the *Kilo*, *Mitaka*, *Ocata*, and *Queens* releases of OpenStack.

OpenStack is a collection of open source technologies that provide scalable computing software for both public and private clouds. For detailed information about OpenStack, see the OpenStack documentation and refer to the documentation of your OpenStack distribution.

You can choose between using IBM Cloud Manager with OpenStack or an OpenStack distribution of another vendor as underlying OpenStack infrastructure for IBM Cloud Orchestrator.

Supports integration with the following OpenStack services that are configured on HTTP and HTTPS:

- OpenStack Kilo services
- OpenStack Mitaka services
- OpenStack *Ocata* services
- OpenStack *Queens* services

You can reconfigure an existing IBM Cloud Orchestrator to a new instance of OpenStack *Mitaka* / *Ocata* / *Queens* or install a new instance of both IBM Cloud Orchestrator and OpenStack *Mitaka* / *Ocata* / *Queens*. For more information about installing and configuring OpenStack *Mitaka*, *Ocata*, or *Queens*, see "Checking OpenStack prerequisites" on page 17 and "Configuring OpenStack Mitaka / Ocata / Queens distribution " on page 43.

IBM Cloud Orchestrator uses the following components and services of OpenStack:

Dashboard (code named Horizon)
Provides a web-based user interface.

Identity (code named Keystone)
Provides authentication and authorizations for all OpenStack services.

Block Storage (code named Cinder)
Provides persistent block storage to guest virtual machines.

Image (code named Glance)
Provides a catalog and repository for virtual disk images. The virtual disk images are mostly used in the OpenStack Compute service component.

Orchestration (code named Heat)
Provides an engine to start multiple composite cloud applications based on templates.

Compute (code named Nova)
Provides virtual servers on demand.

Network (code named Neutron)
Provides network management.

Ceilometer
Collects metering data that is related to CPU and networking.

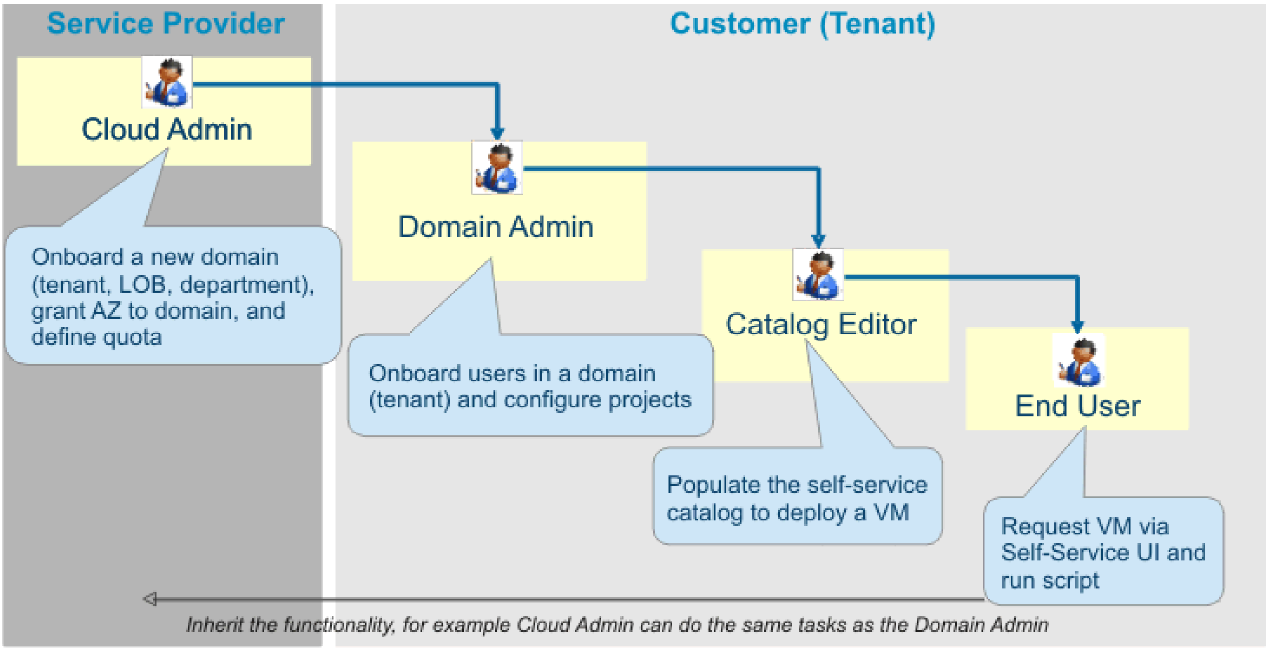
Multitenancy overview

This topic describes the roles and delegation in IBM Cloud Orchestrator.

Delegation means that a more powerful role can delegate certain tasks to a less powerful role. It includes two different types of persona:

Service provider
Is responsible to host IBM Cloud Orchestrator and provide the cloud infrastructure and services.

Service consumer
Consumes services from the service provider and acts only in the context of the tenant.


Service Provider
Customer (Tenant)
Cloud Admin
Onboard a new domain (tenant, LOB, department), grant AZ to domain, and define quota
Domain Admin
Onboard users in a domain (tenant) and configure projects
Catalog Editor
Populate the self-service catalog to deploy a VM
End User
Request VM via Self-Service UI and run script
Inherit the functionality, for example Cloud Admin can do the same tasks as the Domain Admin

IBM Cloud Orchestrator provides different user interfaces that are optimized for the user experience of a specific role. The following user interfaces exist:

OpenStack Dashboard
Only used by a Cloud Administrator. It is the OpenStack Horizon extended by IBM Cloud Orchestrator. It allows the configuration of the cloud infrastructure and identities. The view shows the resources in the context of a selected region.

IBM Process Designer and IBM Business Process Manager
Is only used by cloud administrators and content developers. It is the main user interface to develop new toolkits and catalog content like processes and human services. It can be used to load new content from the IBM Cloud Orchestrator Catalog.

Self-service user interface
It is used by tenant users, like domain administrator, catalog editors and users. It provides a self-service portal with dashboard, self-service catalog, and managing instances that are owned by the user. It further support configuration of the domain and catalog content.

User interfaces are used by the different personas. The following list explains the roles, starting from the most powerful, Cloud Administrator, to the most restrictive, End User:

Cloud Administrator (Service provider)
The Cloud Administrator is the most powerful role who can manage and administer the whole cloud infrastructure resources, identities, self-service catalog, and its artifacts across all tenants. A special persona of the Cloud Administrator is the content developer who implements content packs, processes, and coaches that implement the offerings. The Cloud Administrator can delegate certain tasks, like user, project, and catalog configuration to the Domain Administrator.

Domain Administrator (Service consumer)

The Domain Administrator is the most powerful role within a tenant but less powerful than the Cloud Administrator. The Domain Administrator is responsible to setup users, projects, and self-service catalog content in the domain. However, the Domain Administrator can only rely on resources that are assigned to the domain by the Cloud Administrator. The Domain Administrator can delegate certain tasks like self-service catalog configuration to the catalog editor.

Catalog Editor (Service consumer)

The catalog editor (or Service Designer) is responsible for configuring the self-service catalog for users and projects in the domain. The catalog editor relies on the content packs that are exposed to the domain by theCloud Administrator.

End User (Service consumer)

The End User can request service like virtual machines, and stacks, from the self-service catalog. The End User can also work with the instances, like start, stop virtual machine or install software on virtual machine. Furthermore, the End User can view the dashboard and can work on inbox assignments. The End User always works on-behalf-of a project.

Custom (Service consumer)

A role with same rights as the End User. The service provider can decide to implement an approval process for customers and might introduce the role **Approver**. The content packs might then use the **Approver** role to assign any approval request to users with that role. However, it would still be in the responsibility of the Domain Administrator to decide which user is the approver of a project.

For more information about the role of the Cloud Administrator, see the following topics:

- Chapter 2, “Installing,” on page 9
- Chapter 6, “Administering,” on page 157
- Chapter 13, “Reporting,” on page 297
- *Metering and billing of SCCM User's Guide*
- Chapter 15, “Troubleshooting,” on page 361

For more information about the responsibility of the Domain Administrator, see “Administering as domain administrator” on page 190.

For more information about the role of service designers, see the following topics:

- Chapter 7, “Managing orchestration workflows,” on page 197
- Chapter 8, “Working with self-service,” on page 209
- Chapter 9, “Managing virtual images,” on page 239

For more information about the role of an End User, see “Using self-service” on page 209.

Custom extensions

You create custom extensions to IBM Cloud Orchestrator in the Business Process Manager Process Designer tool and base them on Business Process Manager business processes. To implement user interface extensions, you can use Business Process Manager human services.

IBM Cloud Orchestrator delivers a set of Business Process Manager toolkits that cover the most common automation scenarios in the infrastructure-as-a-service and platform-as-a-service environments. Each toolkit provides a set of reusable artifacts:

Business processes

A business process is any course of action or procedure that an organization follows to achieve a larger business goal. When you break it down, a business process is a series of individual tasks or activities that are performed in a specific order. Business processes provide the primary means through which enterprise services are integrated.

Services

Services provide functions for a business process, which itself is a sequence of services. Creating services separately from a business process means that a service can be developed independently of a business process and that many types of business processes can reuse that service.

Human services

Human service includes an activity in your business process definition that creates an interactive task that process participants can perform in a web-based user interface.

Coaches

Coaches are the user interfaces for human services.

Business object definitions

Business objects carry the functional properties, data transformation information, and file content that the adapter needs to process requests and generate responses.

With the help of these artifacts, you can efficiently build custom extensions for IBM Cloud Orchestrator. The provided toolkits also contain numerous samples that show how to define custom extensions.

You can download more Business Process Manager toolkits from the IBM Cloud Orchestrator Catalog. These toolkits provide more content for different areas, such as networking or storage, and you can also use them to build IBM Cloud Orchestrator extensions.

Restriction: If you define more than one snapshot for Business Process Manager process application or toolkit, you are able to use only the artifacts of the top level to define a new extension in IBM Cloud Orchestrator.

IBM Platform Resource Scheduler overview

Platform Resource Scheduler, also referred as Enterprise Grid Orchestrator (EGO), is a key component of IBM Cloud Manager with OpenStack V4.3. Because IBM Cloud Orchestrator includes IBM Cloud Manager with OpenStack, the Platform Resource Scheduler can be used with the IBM Cloud Orchestrator product.

OpenStack is used to provision VM instances. EGO is used to schedule resources for OpenStack to make decisions on where to deploy the VM instance with specified resource selection criteria, such as migrating, resizing, and powering on.

Platform Resource Scheduler is installed with IBM Cloud Manager with OpenStack. For information about customizing the Platform Resource Scheduler, see Customizing the scheduler.

Once installed, you can learn more about Platform Resource Scheduler and start to use it.

Chapter 2. Installing

You install IBM Cloud Orchestrator by connecting to a freshly installed OpenStack environment or by connecting with IBM Cloud Manager keystone without OpenStack.

Installing IBM Cloud Orchestrator with IBM Cloud Manager with OpenStack

You install IBM Cloud Orchestrator by connecting to a freshly installed OpenStack environment.

For information about how to install OpenStack, see the documentation for your chosen OpenStack product; for example, see the IBM Cloud Manager with OpenStack documentation.

Installation checklist

Use this checklist to ensure that you complete all of the installation steps in the correct order.

Table 3. Installation Checklist

Step	Description	Done?	Comment
“Planning the installation” on page 14			
1	“Choosing the deployment topology” on page 14		
2	“Checking OpenStack prerequisites” on page 17		
3	“Checking the hardware prerequisites” on page 21		
4	“Checking the software prerequisites” on page 22		
“Installing the OpenStack servers” on page 23 - Run one of the following procedures:			
5a	“Installing IBM Cloud Manager with OpenStack and deploying an IBM Cloud Manager with OpenStack cloud” on page 23		
5b	“Installing an OpenStack distribution of another vendor” on page 23		
“[Optional] Installing the NSX plug-in” on page 24			
“Preparing for the IBM Cloud Orchestrator installation” on page 27			
6	“Downloading the required image files” on page 27		
7	“Preparing the IBM Cloud Orchestrator Servers ” on page 29		
8	“[Optional] Creating the Business Process Manager databases on an external IBM DB2 server” on page 30		

Table 3. Installation Checklist (continued)

Step	Description	Done?	Comment
“Configuring the OpenStack servers” on page 33 - Run one of the following procedures depending on your OpenStack environment:			
9a	“[Typical] Configuring the IBM Cloud Manager with OpenStack servers” on page 33		
9b	“[Advanced] Configuring the IBM Cloud Manager with OpenStack servers ” on page 37		
9c	“Configuring an OpenStack Kilo distribution ” on page 50		
Preparing the installer			
10	“Setting the deployment parameters” on page 60		
11	“Adding the OpenStack simple token to the response file” on page 64		
12	“Checking the installation prerequisites” on page 65		
Installing			
13	“Deploying the IBM Cloud Orchestrator Servers” on page 66		
14	“Verifying the installation” on page 68		
15	“Reconfiguring IBM Cloud Manager with OpenStack after updates” on page 71		
	“Installing high availability across multiple sites” on page 72		
15	“Troubleshooting the installation” on page 74		

Installing IBM Cloud Orchestrator with keystone topology

If you want to use the DirectDriver and do not want IBM Cloud Manager with OpenStack, then install IBM Cloud Orchestrator with keystone topology.

Installation checklist

Use this checklist to ensure that you complete all of the installation steps in the correct order.

Table 4. Installation Checklist

Step	Description	Done?	Comment
“Planning the installation” on page 14			

Table 4. Installation Checklist (continued)

Step	Description	Done?	Comment
1	“Choosing the deployment topology” on page 14		
3	“Checking the hardware prerequisites” on page 21		
4	“Checking the software prerequisites” on page 22		
“Preparing for the IBM Cloud Orchestrator installation” on page 27			
6	“Downloading the required image files” on page 27		
7	“Preparing the IBM Cloud Orchestrator Servers ” on page 29		
8	“[Optional] Creating the Business Process Manager databases on an external IBM DB2 server” on page 30		
9	“Installing IBM Cloud Orchestrator Keystone topology” on page 11		
Preparing the installer			
10	“Setting the deployment parameters” on page 60		
11	“Checking the installation prerequisites” on page 65		
Installing			
12	“Deploying the IBM Cloud Orchestrator Servers” on page 66		
13	“Verifying the installation” on page 68		
14	“Troubleshooting the installation” on page 74		

Installing IBM Cloud Orchestrator Keystone topology

If you want to use the DirectDriver and do not want IBM Cloud Manager with OpenStack, then install IBM Cloud Orchestrator with keystone topology. It installs IBM Cloud Orchestrator with IBM Cloud Manager keystone. You do not have to deploy or manage a three node topology that comprises of IBM Cloud Manager with OpenStack, controller node, and IBM Cloud Orchestrator.

Before you begin

- Set the yum repository:

  Note: For high-availability, run the following steps on both primary and secondary nodes.

  1. Copy the RPMS from `/opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/openstack_packages/` to another location if required.
  2. Run the following command to create a repository:

```
createrepo <path_where_the_openstack_package_rpms_are_copied>
```

If **createrepo** is not available, then install by using the following yum command:

```
yum install -y createrepo yum-utils
```

3. Create a repository file at /etc/yum.repos.d/ location and provide a name of your choice.

   For example, /etc/yum.repos.d/ibmos.repo.

   The file prefix has three forward slashes, for example, file:///opt/repofiles/folder.

   Sample repos file:

   ```
   # This file was generated by Chef
   # Do NOT modify this file by hand.

   [ibmos-noarch]
   name=IBM OpenStack package repository (noarch)
   baseurl="file:///<path_where_the_openstack_package_rpms_are_copied>"
   enabled=1
   gpgcheck=0
   priority=10
   sslverify=true
   ```

   Use **yum repolist** to check the repo status. Your new repo gets listed with 77 packages, for example, "ibmos-noarch IBM OpenStack package repository (noarch) 77".

- When you upgrade from IBM Cloud Orchestrator Keystone HA HTTPS, do the following:
  - Copy the certificate /etc/keystone/ssl/certs/sslca.crt to the <TMP_DIR> on both the primary and secondary nodes. The <TMP_DIR> is the value provided in the ico_install.rsp.

    For example, cp /etc/keystone/ssl/certs/sslca.crt /tmp/ico/openstack.crt

    Where /tmp/ico is the value of <TMP_DIR>.
  - Update OPENSTACK_HOST_NAME in ico_install.rsp with the virtual host name/ IP address that is dependent on the keystone endpoint of IBM Cloud Orchestrator setup.
- Disable SElinux on the virtual machine where IBM Cloud Orchestrator keystone topology is to be deployed before installation:
  1. Open file /etc/selinux/config for editing:

     ```
     set SELINUX=disabled
     ```
  2. Save the changes.
  3. Reboot the virtual machine for the changes to be reflected.

About this task

Set the IBM Cloud Orchestrator Keystone topology to true and update the ico_install.rsp file with your OpenStack host name details. Run the IBM Cloud Orchestrator installation along with the ico_install.rsp file.

If you are using this topology, you do not have to follow the procedure to configure IBM Cloud Manager with OpenStack.

Consider the following support-related information for IBM Cloud Orchestrator Keystone on high-availability and non high-availability topologies:

- High-availability and non high-availability - This IBM Cloud Orchestrator Keystone topology is applicable for both fresh installation and upgrade procedure ofIBM Cloud Orchestrator on HTTPS.

For troubleshooting information related to IBM Cloud Orchestrator Keystone topology, see “Troubleshooting IBM Cloud Orchestrator Keystone topology” on page 371.

Procedure

1. Update the `ico_install.rsp` with the following parameters:

   For non high-availability:

   ```
   KEYSTONE_ICO_INSTALLATION True
   OPENSTACK_HOST_NAME <IBM Cloud Orchestrator HOST NAME/IP>
   PROTOCOL https
   ```

   For high-availability:

   ```
   KEYSTONE_ICO_INSTALLATION True
   OPENSTACK_HOST_NAME <primary_hostname>
   PROTOCOL https
   ```

2. Log in to the server where you want to install IBM Cloud Orchestrator.
3. Run the following command to change directory to the installer directory:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   ```

4. Run the following command to install the IBM Cloud Orchestrator components on the server:

   - If you logged on as `root` user:

     ```
     ./ico_install.sh ico_install.rsp
     ```

   - If you did not log on as `root` user:

     ```
     sudo ./ico_install.sh ico_install.rsp
     ```

   You are prompted to accept the license agreement. Read the license agreement, and accept or decline the license terms. If you do not accept the license agreement, the installation exits.

   Note: To accept the license automatically without being prompted by the installer, set the **`LICENSE_ACCEPTED`** parameter to `True` in the response file.

   Subject to disk speed, the installation should complete within 2 hours.

5. If your install fails with an error, review the installation log file (`/var/log/ico_install/ico_install_YYYYMMDDhhmm.log`) to check why the installation was unsuccessful. If necessary, take appropriate action as indicated in the log file.
6. For high-availability, restart all the services.

   For the steps to restart the services, see “Managing the services” on page 158.

7. If not required, manually delete the endpoints v2 or HTTP endpoints.

   ```
   source /root/v3rc
   openstack endpoint delete
   ```

8. For IBM Cloud Orchestrator multi-tenancy support, make sure that the external authentication is configured in Keystone. Set the following values for the external parameter in the `[auth]` section of the `/etc/keystone/keystone.conf` file:

   ```
   [auth]
   external = keystone.auth.plugins.external.Domain
   ```

   If the `[auth]` section is not already available, create it.

What to do next

Complete the installation verification steps, as described in “Verifying the installation” on page 68.

Planning the installation

Before you start the installation, it is important to understand the installation flow.

Procedure

1. Install an OpenStack distribution. For more information, see "Installing the OpenStack servers" on page 23.
2. Configure the OpenStack distribution and add functional enhancements.

   The OpenStack distribution requires a set of configuration steps to satisfy the needs of the IBM Cloud Orchestrator. Also, you must add functional enhancements to the OpenStack distribution. For more information, see "Configuring the OpenStack servers" on page 33.
3. Install the IBM Cloud Orchestrator components.

   You must install the IBM Cloud Orchestrator specific components that are connected to the OpenStack distribution that you previously installed.

Choosing the deployment topology

Before you start to deploy IBM Cloud Orchestrator, you must decide which deployment topology to install for the IBM Cloud Orchestrator management stack. Depending on your needs and the available hardware resources, you can configure your environment as a single-server environment, or as a highly available environment spread across two servers.

Supported cloud types

When planning your installation, decide what type of cloud you want to manage: private, public, or both (hybrid):

- For private cloud, choose the hypervisor type: Hyper-V, KVM, PowerKVM, PowerVC, VMware, z/VM.
- For public cloud, choose the cloud provider: IBM SoftLayer, Amazon EC2, Microsoft Azure.
- For hybrid cloud, choose the hypervisor type and the cloud provider.

Supported deployment topologies

IBM Cloud Orchestrator supports the following deployment topologies:

- Single-Server topology
- Single-Server with external database topology
- Dual-Server high-availability with external database topology
  - Single location:

    The two servers are located on the same geographic site. Failover can be started automatically establishing a quorum.
  - Multiple location:

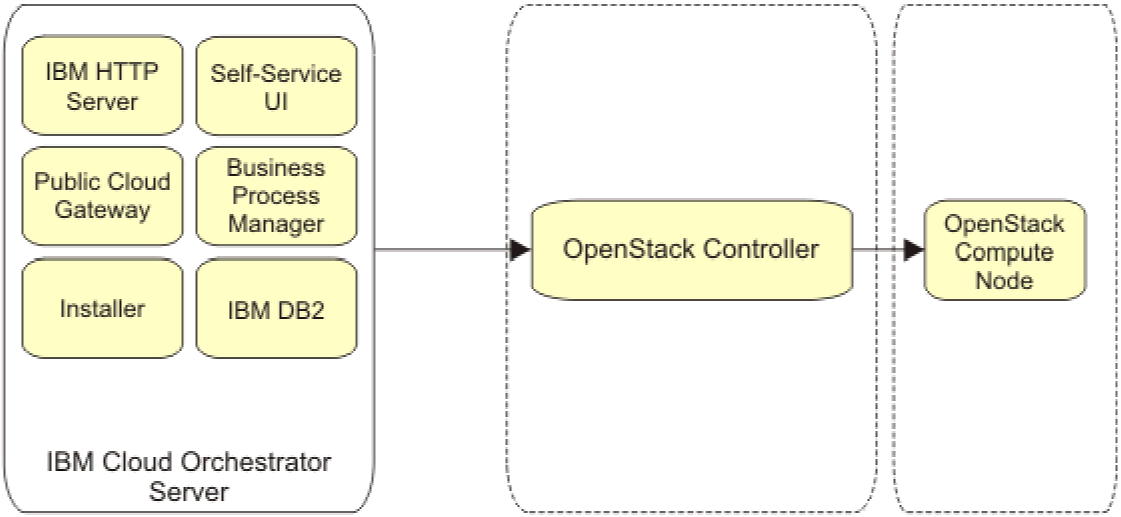
    The two servers are located on different sites. Failover must be started manually by an operator because the quorum cannot be established when the network connectivity between the two sites is down.
- DirectDriver Keystone topology

Tip: When you have chosen the deployment topology, review the deployment parameters that are listed in "Setting the deployment parameters" on page 60, and identify appropriate values for these parameters for your installation.

The images in this topic show an example of an IBM Cloud Orchestrator and OpenStack topology with one OpenStack Controller and one compute node.

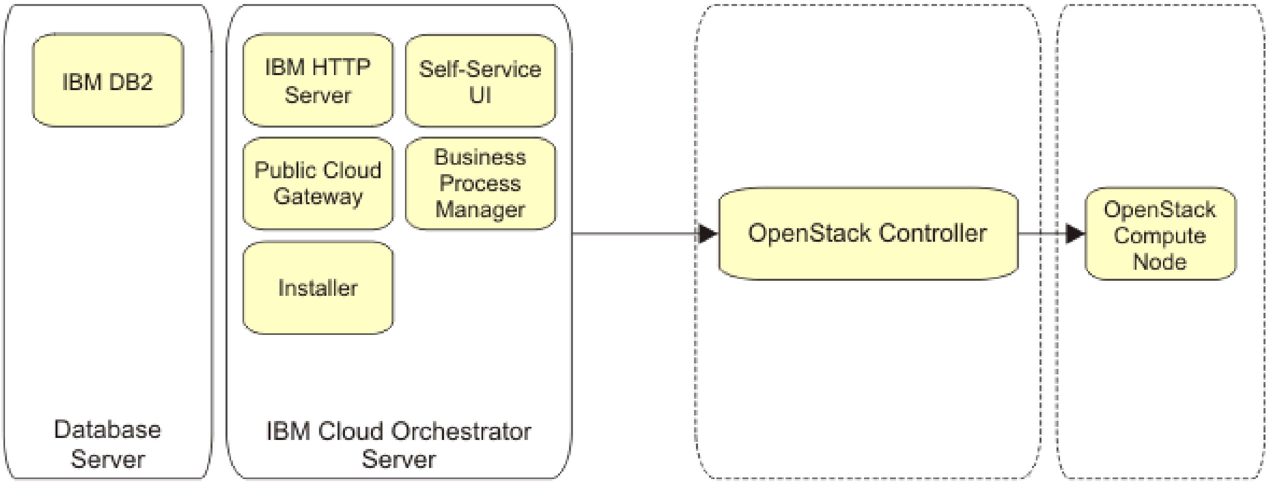
Single-Server topology

All of the IBM Cloud Orchestrator management components are installed on a single IBM Cloud Orchestrator Server.



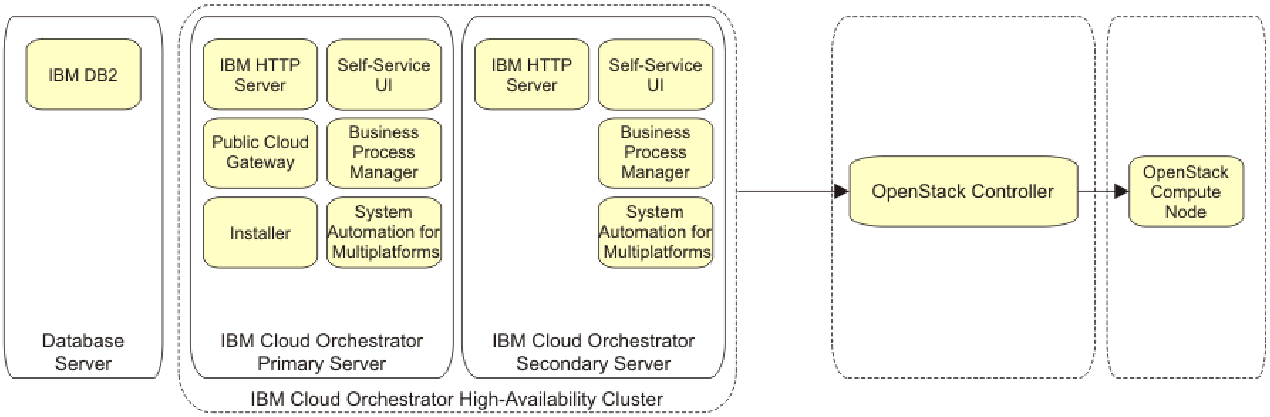
Single-Server with external database topology

The IBM Cloud Orchestrator management components are installed on a single IBM Cloud Orchestrator Server, pointing to an existing external IBM DB2 database server.



Dual-Server high-availability with external database topology

The IBM Cloud Orchestrator management components are installed on two IBM Cloud Orchestrator Servers in a high-availability configuration, pointing to an existing external IBM DB2 database server.



Note: The Public Cloud Gateway component is a single point of failure and it is not a highly-available component.

Note: IBM Cloud Orchestrator does *not* provide high availability for the OpenStack components. For information about high availability in IBM Cloud Manager with OpenStack, see the IBM Cloud Manager with OpenStack documentation.

For information about multi-site high availability installation, see "Installing high availability across multiple sites" on page 72.

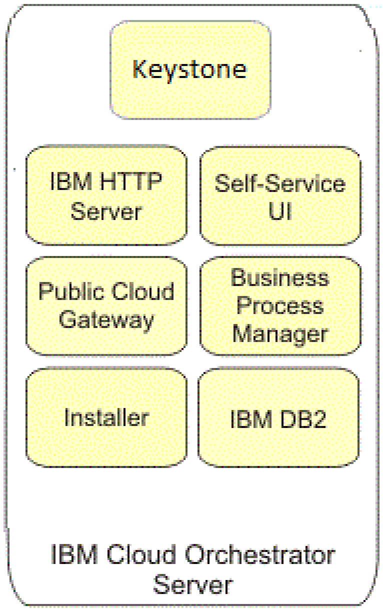
IBM Cloud Orchestrator Keystone topology

If you want to use the DirectDriver that is included in IBM Cloud Orchestrator, but do not want the IBM Cloud Manager with OpenStack, then install IBM Cloud Orchestrator with keystone. This topology installs IBM Cloud Orchestrator with IBM Cloud Manager keystone.IBM Cloud Orchestrator with DirectDriver implementation provisions without any OpenStack components. In this case, you do not have to deploy or manage a three node topology that comprises of IBM Cloud Manager with OpenStack, controller node, and IBM Cloud Orchestrator.

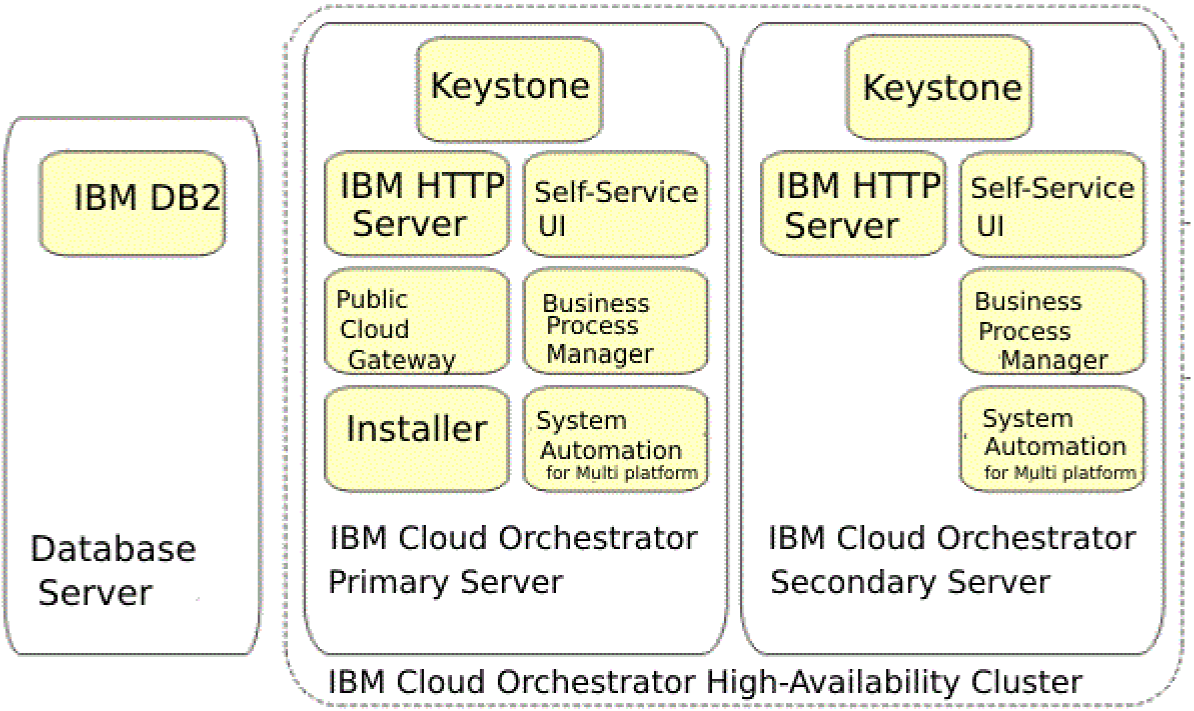
Note: Only the following scenarios are supported in this release:

- Fresh installation of high-availability HTTPS keystone topology
- Non high-availability upgrade of HTTP to HTTPS keystone topology
- Non high-availability with external database

IBM Cloud Orchestrator Keystone non high-availability topology:



IBM Cloud Orchestrator Keystone high-availability topology:


IBM DB2
Database Server
Keystone
IBM HTTP Server
Self-Service UI
Public Cloud Gateway
Business Process Manager
Installer
System Automation for Multi platform
IBM Cloud Orchestrator Primary Server
Keystone
IBM HTTP Server
Self-Service UI
Business Process Manager
System Automation for Multi platform
IBM Cloud Orchestrator Secondary Server
IBM Cloud Orchestrator High-Availability Cluster

For more information about the actual steps, see "Installing IBM Cloud Orchestrator Keystone topology" on page 11.

Deployment topology components

The main components of an IBM Cloud Orchestrator deployment topology are as follows:

IBM Cloud Orchestrator Server
This server hosts the core IBM Cloud Orchestrator management components. For the high-availability topology, there are two IBM Cloud Orchestrator Servers.

OpenStack Controllers (previously known as Region Servers)
Each OpenStack Controller communicates with a specific hypervisor management infrastructure:

- The Hyper-V OpenStack Controller requires one or more Hyper-V compute nodes to provide the compute resources.
- The KVM OpenStack Controller requires one or more KVM compute nodes to provide the compute resources.
- The PowerVC OpenStack Controller must connect to an existing PowerVC to provide virtual machines.
- The VMware OpenStack Controller must connect to an existing VMware Virtual Center to provide virtual machines.
- The z/VM OpenStack Controller must connect to an xCat management node on z/VM to provide virtual machines.

Compute nodes
[KVM or Hyper-V] The compute nodes manage the virtual machines through the interface that is provided by KVM or Hyper-V.

Checking OpenStack prerequisites

Ensure that your OpenStack environment meets the software prerequisites for your IBM Cloud Orchestrator installation. You can either install IBM Cloud Manager with OpenStack that is part of the IBM Cloud Orchestrator product or you can install an OpenStack distribution of another vendor (Bring your own OpenStack). You can connect IBM Cloud Orchestrator only to a single OpenStack instance (one-to-one mapping).

Prerequisites for IBM Cloud Manager with OpenStack

For IBM Cloud Orchestrator, ensure that your IBM Cloud Manager with OpenStack installation meets the following requirements:

- IBM Cloud Manager with OpenStack V 4.3 Fix Pack 14

  For information about IBM Cloud Manager with OpenStack prerequisites, see the IBM Cloud Manager with OpenStack documentation.
- Shared Keystone service.

  The IBM Cloud Manager with OpenStack installation uses a single Keystone service. In a multi-region installation, the Keystone service must run on the OpenStack Controller of the first region that is installed. OpenStack's Identity Federation is not supported by IBM Cloud Orchestrator.

  For more information about how to configure IBM Cloud Manager with OpenStack Controllers with multi-region support with Shared Keystone, refer to Deploying multi-region support.

  If some components of the IBM Cloud Orchestrator are deployed in public data center or cloud (on public network), enable HTTPS and VPN connection between IBM Cloud Orchestrator and OpenStack. For more information, see https://www.ibm.com/support/knowledgecenter/en/SST55W_4.3.0/liaca/liaca_hybrid_hybrid_cloud.html.
- Fresh installation.

  The IBM Cloud Manager with OpenStack installation is a fresh installation. No users, projects, or domains are defined other than those that are created during the basic OpenStack installation.
- One Heat engine per region.

  The IBM Cloud Manager with OpenStack installation must use one Heat engine per region. Configuring Heat with multi-region support is not supported by IBM Cloud Orchestrator.
- Use the OpenStack capabilities through the IBM Cloud Orchestrator interface.

  A parallel usage of the IBM Cloud Manager with OpenStack installation through IBM Cloud Orchestrator and standard OpenStack is not supported. If an IBM Cloud Manager with OpenStack installation is configured to be used by IBM Cloud Orchestrator, all user activities must be done through the IBM Cloud Orchestrator interface, and not through the OpenStack interfaces. The OpenStack Self-Service interface must be only used as described in the IBM Cloud Orchestrator documentation.
- One OpenStack region per hypervisor type.

  Each region must have only one hypervisor type. IBM Cloud Orchestrator does not support the usage of multiple types of hypervisor within a region.
- Customized simple token setup.

  The IBM Cloud Manager with OpenStack distribution provides the simple token enhancement that is required and used by IBM Cloud Orchestrator. For more information about how to set up the simple token, see Customizing passwords and secrets and Data bags in the IBM Cloud Manager with OpenStack documentation.
- Do not install the self-service user interface extension of IBM Cloud Manager with OpenStack.

  The self-service user interface extension of IBM Cloud Manager with OpenStack does not work with IBM Cloud Orchestrator and it must not be installed. If it is installed, see Uninstalling the self-service user interface on Linux to uninstall it.
- IBM Cloud Orchestrator Server supports only IPv4 address type. Do not use IPv6 address type in IBM Cloud Manager with OpenStack.

IBM Cloud Orchestrator supports the following hypervisors in an IBM Cloud Manager with OpenStack environment:

- Hyper-V
- KVM
- PowerKVM

- PowerVC
- VMware
- z/VM

For information about hypervisor requirements in an IBM Cloud Manager with OpenStack environment, see IBM Cloud Manager with OpenStack virtualization environment prerequisites .

Consider the following points whenever you work with OpenStack:

- If the OpenStack finds that a virtual machine does not have the expected host, then it deletes the virtual machine because it assumes that the virtual machine is already evacuated. To prevent such a deletion by the OpenStack, set the `destroy_after_evacuate=False`. For deployed regions, include the following information in the `[workarounds]` section of the Nova configuration files:

  ```
  [workarounds]
  destroy_after_evacuate=False
  ```

  If the `[workarounds]` section is not available, create the section and add the entry.

  In VMware hypervisor, when you run multiple Nova compute services that are configured to address the same server cluster, you must disable the automatic removal of the provisioned virtual machines.

  A Nova config file exists for every Nova compute service so ensure that you update this entry in all the config files. If you start a new OpenStack Nova service or restart an existing service without adding this entry in the config file, then all previously deployed virtual machines get automatically deleted.

  If you must consider additional configuration values, see Configuring Nova settings for virtual machines section of IBM Cloud Manager with OpenStack Knowledge Center.
- To support deployment to the hypervisors in an IBM Cloud Manager with OpenStack environment, you must set the **`force_config_drive`** parameter to `always` by running one of the following procedures:
  - On the IBM Cloud Manager with OpenStack deployment server, for each OpenStack server where the compute service is running, update the environment file by running the following command:

    ```
    knife environment edit <your-environment-name>
    ```

    and set the **`force_config_drive`** value to `always`. This is the recommended procedure to make the change persistent. For more information, see Updating a deployed topology.
  - Set the following option in the `[DEFAULT]` section of the Nova configuration file in the `/etc/nova` directory on the OpenStack server where the compute service is running:

    ```
    force_config_drive=always
    ```

    and restart the compute service. The Nova configuration file might be overwritten during IBM Cloud Manager with OpenStack environment maintenance. If you want to make the change persistent, do not use this procedure.
- To support VMware hypervisors in an IBM Cloud Manager with OpenStack environment, set the following option in the `[vmware]` section in the VMware Nova configuration file in the `/etc/nova` directory on the VMware OpenStack Controller (the file name is customized from the cluster name as, for example, `nova-vcenter-Cluster.conf`):

  ```
  customization_enabled = false
  ```

  and restart the compute service.
- If you install fix packs for IBM Cloud Manager with OpenStack or you do other maintenance work that updates the topology through `knife`, the configuration that was done for IBM Cloud Orchestrator might be overwritten. Rerun the `ICM_config_ico.sh` configuration script as described in "Configuring IBM Cloud Manager with OpenStack for IBM Cloud Orchestrator" on page 35.
- If you add a region to IBM Cloud Manager with OpenStack after it was configured for IBM Cloud Orchestrator, the new region is not correctly set up to work with SmartCloud Cost Management. Rerun

the `ico_configure.sh` configuration script as described in *Automated configuration section of SCCM User's Guide* to collect data from the new region.

Prerequisites for an OpenStack environment of a different vendor

For general OpenStack requirements, see the documentation for your chosen OpenStack product.

For IBM Cloud Orchestrator, ensure that your OpenStack installation (Bring your own OpenStack) meets the following requirements:

- OpenStack *Kilo* or OpenStack *Mitaka* or OpenStack *Ocata*, or OpenStack *Queens* release.

  OpenStack *Kilo* or *Mitaka* or *Ocata*, or OpenStack *Queens* release is installed. Refer to the documentation of your OpenStack product.
- Shared Keystone service.

  The OpenStack installation uses a single Keystone service. Federation is not supported. For more information, see your OpenStack documentation.
- Fresh installation.

  The OpenStack installation is a fresh installation. No users, projects, or domains are defined other than those that are created during the basic OpenStack installation.
- RefStack compliance.

  The OpenStack installation must be RefStack compliant.
- One Heat engine per region.

  The OpenStack installation must use one Heat engine per region. Configuring Heat with multi-region support is not supported by IBM Cloud Orchestrator.
- Use OpenStack capabilities through the IBM Cloud Orchestrator interface.

  A parallel usage of the OpenStack installation through IBM Cloud Orchestrator and standard OpenStack is not supported. If an OpenStack installation is configured to be used by IBM Cloud Orchestrator, all user activities must be done through the IBM Cloud Orchestrator interface, and not through the OpenStack interfaces. The OpenStack interfaces must be used only as described in the IBM Cloud Orchestrator documentation.
- The OpenStack nodes must support to receive HTTP or HTTPS calls based on OpenStack setting. IBM Cloud Orchestrator communicates with OpenStack Keystone through HTTP or HTTPS calls based on OpenStack settings. Make sure that your OpenStack nodes (firewall, OpenStack configuration, and so forth) are able to receive HTTP or HTTPS calls based on OpenStack settings.
- IBM Cloud Orchestrator Server supports only IPv4 address type. Do not use IPv6 address type in OpenStack.
- Shared Keystone service

  In multi-region installation, the Keystone must be run on the OpenStack Controller of the first region that is installed. The OpenStack's Identity Federation is not supported by IBM Cloud Orchestrator. It also does not support multiple types of hypervisor within a region.

You must configure your OpenStack distribution by following the procedure that is described in "Configuring an OpenStack distribution of another vendor" on page 43.

Depending on the database backend, OpenStack might be case-sensitive.

If you plan to reconfigure IBM Cloud Orchestrator that is an already deployed with OpenStack Kilo (IBM Cloud Manager with OpenStack V4.3 or external OpenStack Kilo) to OpenStack Mitaka or OpenStack *Ocata*, or OpenStack *Queens*, then consider the following limitations:

- You cannot reconfigure IBM Cloud Orchestrator back to OpenStack Kilo.
- The migration of IBM Cloud Orchestrator for OpenStack data to OpenStack *Mitaka* or OpenStack *Ocata* is not supported. You must create data again in OpenStack *Mitaka* or *Ocata*.

- The migration of Public Cloud Gateway data to OpenStack *Mitaka* or OpenStack *Ocata* is not supported. You must create data again in OpenStack Mitaka or OpenStack *Ocata*.

Checking the hardware prerequisites

Ensure that your environment meets the hardware prerequisites for your IBM Cloud Orchestrator installation.

IBM Cloud Orchestrator is installed on one or two *IBM Cloud Orchestrator Servers* depending on the topology you chose. The IBM Cloud Orchestrator Server can be a virtual machine or a physical server, and must meet the following minimum hardware requirements.

Table 5. Hardware prerequisites for a typical installation

				Free hard disk space by partition (GB)			
Server	**Processor (vCPU)**	**Memory (GB)**	**Total free hard disk space (GB)**	**/**	**/home**	**/opt**	**<TMP_DIR>**
IBM Cloud Orchestrator Server	4	8	66	10	10	36	20
IBM Cloud Orchestrator Server in a high-availability topology	4	8	68	10	2	46	20
IBM Cloud Orchestrator Keystone topology	4	8	66	10	10	36	20

Table 6. Hardware prerequisites for upgrade

				Free hard disk space by partition (GB)			
Server	**Processor (vCPU)**	**Memory (GB)**	**Total free hard disk space (GB)**	**/**	**/home**	**/opt**	**<TMP_DIR>**
IBM Cloud Orchestrator Server	4	8	60	10	2	28	20
IBM Cloud Orchestrator Server in a high-availability topology	4	8	65	10	2	30	20
IBM Cloud Orchestrator Keystone topology	4	8	60	10	2	28	20

Disk planning considerations:

- The specified hard disk space is the minimum free space that is required on the server before the IBM Cloud Orchestrator installation. Ensure that the hard disk has sufficient space for the required partitions.

- The specified hard disk space is for an installation in the default directories. The `/opt` value includes 16 GB for the installation files, which can be deleted after the installation completes successfully. If you install in other directories, ensure that you have sufficient space in those directories.
- The installation directory (`/opt/ibm/ico` by default) must not be a symbolic link.
- Extra space might be required on the `/home` partition after a period of time, depending on database size. Monitor the partition size. Use Logical Volume Manager (LVM) to manage the partition so that you can extend the size if required.
- Create the <`TMP_DIR`> on both the primary and secondary nodes with correct privileges. The default location for <`TMP_DIR`> is `/tmp/ico`. The <`TMP_DIR`> directory must not be mounted with the `noexec`, `nodev`, and `nosuid` options during the installation procedure. You can change the <`TMP_DIR`> directory configuration after IBM Cloud Orchestrator is installed.

Note: For the hardware prerequisites of the OpenStack environment, check either the IBM Cloud Manager with OpenStack prerequisites (at http://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liacasoftware.html) or refer to the documentation of your vendor-specific OpenStack distribution.

Checking the software prerequisites

Ensure that your IBM Cloud Orchestrator Server environment meet the software prerequisites for your IBM Cloud Orchestrator installation.

IBM Cloud Orchestrator is installed on one or two IBM Cloud Orchestrator Servers depending on the topology you chose. The operating system of the IBM Cloud Orchestrator Servers must be Red Hat Enterprise Linux, and the architecture must be x86_64. The supported versions of Red Hat Enterprise Linux are 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, and 7.6. For IBM Cloud Orchestrator Keystone topology supports only Red Hat Enterprise Linux 7.3, 7.4, and 7.5.

The IBM Cloud Orchestrator installer must access one of the following Red Hat Enterprise Linux repositories:

- Registered Red Hat Network
- Customer-provided `yum` repository
- Red Hat Enterprise Linux ISO

You must configure the IBM Cloud Orchestrator Servers as follows:

- The operating system must be installed with at least the standard minimal installation package.

  When installing the Red Hat Enterprise Linux operating system for the IBM Cloud Orchestrator Servers, the minimal installation package is sufficient, because the IBM Cloud Orchestrator deployment script installs the required packages from the corresponding YUM repository or Red Hat ISO files.
- To prevent the vulnerabilities described in CVE-2016-0777 and CVE-2016-0778, the minimum prerequisite level of OpenSSH is `openssh-6.6.1p1-23`.
- Host name resolution must work between the IBM Cloud Orchestrator Servers and the OpenStack servers. You can configure the IBM Cloud Orchestrator Servers with the corporate DNS. If no corporate DNS is available, you must update the `/etc/hosts` file on *each* of the required servers (for example, IBM Cloud Orchestrator Servers, OpenStack Controllers, compute nodes) to include *all* of the IBM Cloud Orchestrator and OpenStack server hosts. Each entry in the `/etc/hosts` file must specify both the fully qualified domain name and the host name, in that order. To verify that you configured the `/etc/hosts` file correctly, run the following commands:

  `host <IP_address>`
  This command must return the FQDN of the server (for example, `ico_server.subdomain.example.com`).

  `hostname --fqdn`
  This command must return the same FQDN as in the previous command.

  `hostname`
  This command must return the first part of the FQDN that is the host name (for example, `ico_server`).

- The ports that are used by IBM Cloud Orchestrator must be open. For a list of the ports that must be open, see "Ports used by IBM Cloud Orchestrator" on page 80.
- If you are installing a high-availability topology, the IBM Cloud Orchestrator Servers must be able to communicate each other by using SSH.
- [VMware only] If you install IBM Cloud Orchestrator on a VMware virtual machine, install VMware Tools to improve the performance of the virtual machine.

Note:

- For supported versions of Data Protection and Recovery, Databases and Process Management tools, see the **Prerequisites** tab of https://www.ibm.com/software/reports/compatibility/clarity-reports/report/html/softwareReqsForProduct?deliverableId=1386584144400.
- For the software prerequisites of the OpenStack environment check either the IBM Cloud Manager with OpenStack prerequisites (at http://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liacasoftware.html) or refer to the documentation of your vendor-specific OpenStack distribution.

Installing the OpenStack servers

Before you install IBM Cloud Orchestrator, you must install an OpenStack distribution. IBM Cloud Manager with OpenStack V4.3 is bundled with IBM Cloud Orchestrator. Alternatively, you can install an OpenStack distribution of another vendor (Bring Your Own OpenStack) which fulfills certain prerequisites of IBM Cloud Orchestrator.

Run one of the following procedures:

- "Installing IBM Cloud Manager with OpenStack and deploying an IBM Cloud Manager with OpenStack cloud" on page 23
- "Installing an OpenStack distribution of another vendor" on page 23

Note: Use standard naming convention whenever you configure OpenStack services. For more information about the standard naming convention, see https://docs.openstack.org/.

Installing IBM Cloud Manager with OpenStack and deploying an IBM Cloud Manager with OpenStack cloud

IBM Cloud Manager with OpenStack V4.3 is bundled with IBM Cloud Orchestrator. If you want to use IBM Cloud Manager with OpenStack with IBM Cloud Orchestrator, you can download and use it following the IBM Cloud Manager with OpenStack documentation. To download fixes and upgrade, follow the IBM Cloud Manager with OpenStack documentation. IBM Cloud Orchestrator requires at least IBM Cloud Manager with OpenStack V4.3 Fix Pack 14 to be installed. You can download the following bundled IBM Cloud Manager with OpenStack packages to the IBM Cloud Manager with OpenStack deployment server:

```
cloud_mgr_4.3_x86_rhel_1.tar.gz
cloud_mgr_4.3_x86_rhel_2.tar.gz
```

To install IBM Cloud Manager with OpenStack and deploy a IBM Cloud Manager with OpenStack cloud, see Installing and uninstalling IBM Cloud Manager with OpenStack and Deploying an IBM Cloud Manager with OpenStack cloud.

Installing an OpenStack distribution of another vendor

For information about how to install OpenStack, see the documentation of your chosen OpenStack product.

[Optional] Installing the NSX plug-in

After you install the OpenStack server, install the NSX plug-in, if required. See [Optional] Installing the NSX plug-in for details.

[Optional] Installing the NSX plug-in

The NSX plug-in can be installed and configured with IBM Cloud Manager with OpenStack V4.3 Fix Pack 13.

Before you begin

Ensure that the following prerequisites are installed and configured:

- VMware vSphere
- VMware NSX version 6.2.2, 6.2.4, 6.3.1, 6.4.1, 6.4.4, 6.4.5

  Note: vSphere 6.5.0 version is not supported with VMware NSX 6.3.1. You must use vSphere versions 6.5a or later.
- IBM Cloud Manager with OpenStack V4.3 Fix Pack 14 deployment server.

About this task

The NSX plug-in can be used for an IBM Cloud Manager with OpenStack deployment with a single or multiple VMware regions, including high-availability (HA) topology. The NSX plug-in is supported on a newly deployed VMware region that is configured with NSX capability.

The NSX plug-in supports the following features:

- Image discovery. Other discoveries, such as network and virtual machine, are not supported.
- Network creation and deletion
- Subnet addition and removal
- Router and network interface addition and deletion
- Source network address translation (SNAT) and destination network address translation (DNAT) configuration for external networks
- Security groups and rules addition and deletion

This procedure includes instructions to install and configure the NSX plug-in. The listed tasks must be completed on the IBM Cloud Manager with OpenStack deployment server.

Procedure

1. Log on to the IBM Cloud Manager with OpenStack deployment server as a root user.
2. Create a directory with the name `nsxplugin`.
3. Change directory to `nsxplugin`.

   ```
   cd nsxplugin
   ```
4. Copy the NSX plug-in package file from the `ICObuild/installer/NSX` directory on the IBM Cloud Orchestrator server to the current location.

   Note:

   - The package file `icm-fp10-F20180402-0827-nsx-patch.tgz` must be used for NSX 6.2.2, 6.2.4, 6.3.1.
   - The package file `icm-fp12-F20181021-1355-nsx-patch.tgz` must be used for NSX 6.4.1 and 6.4.4.
   - The package file `icm-fp14-F20190924-0603-nsx-patch.tgz` must be used for NSX 6.4.5.
5. Unpack the package file.

   For example, run the following command to unpack `icm-fp10-F20180402-0827-nsx-patch.tgz` file:

   ```
   gunzip icm-fp10-F20180402-0827-nsx-patch.tgz
   tar -xvf  icm-fp10-F20180402-0827-nsx-patch.tar
   ```

A directory with the name `icm-fp10-F20180402-0827-nsx-patch` is created with the following NSX plug-in files and folders:

- **`install_nsx_patch.sh`** is a script to install the NSX plug-in.
- **`product_files`** is a folder with the NSX plug-in files.
- **`tools`** is a folder with tools to collect vSphere information.
- **`md5sum.chk`** is a tool to check the integrity of the installation files.

6. Change directory to the NSX plug-in files location.

   ```
   cd icm-fp10-F20180402-0827-nsx-patch
   ```

7. Run the following script to install the NSX plug-in:

   ```
   ./install_nsx_patch.sh
   ```

   If the NSX plug-in script installs successfully, you see the following message displayed:

   `SUCCESS: Installation of 'NSX Plugin support patch' on Deployement Server is complete"`

   If you do not see the message, see the log files to identify the issues. The log files are stored in the working directory.

8. Optional: Run the following commands to collect the vSphere Managed Object IDs:

   a.
   ```
   yum install python-requests
   ```

   b.
   ```
   cd tools/
   ```

   c.
   ```
   ./nsxvcfghelp vcenter_IP nsx_mgr_IP vcenter_user vcenter_password
   ```

9. Configure the NSX plug-in based on your deployment type.

   For more information about deployment types with VMware compute nodes, see Deploying with VMware compute nodes.

   - **Configuring the NSX plug-in for a single or multiple VMware region topology**

     Note: A sample configuration is provided at `/opt/ibm/cmwo/cli/config/example-controller-vmware-nsx-driver-cloud.yml`.

     Prepare the cloud deployment YAML file:

     a. Delete the `network` section under the VMware `driver specific information (vmware)` section. If no `network` section is present, proceed with the next step.

     b. Add the following details under the VMware `driver specific information (vmware)` section:

     ```
     "use_nsx": true,
     ```

     ```
     "nsx": {
          "version": "version number of the NSX Manager",
          "manager_uri": "URI to access the NSX Manager",
          "user": "NSX Manager admin user name",
          "password": "NSX Manager admin password",
          "datacenter_moid": "Managed Object ID of the target datacenter",
          "cluster_moid": "Managed Object ID of the cluser",
          "resource_pool_id": "Managed Object ID of the resource pool",
          "datastore_id": "Managed Object ID of the datastore",
          "external_network": "Managed Object ID of the port group on which
                            the external network will be created",
          "vdn_scope_id": "Managed Object ID of the VDN Scope",
          "dvs_id": "Managed Object ID of the distributed virtual switch that
                  will be used to provision the Neutron networks"
     ```

     c. Deploy the changes by using the `knife os manage deploy cloud your-cloud.yml` command.

- **Configuring the NSX plug-in for an HA topology**

  Prepare the the environment and topology JSON files.

  Edit the environment JSON file:

  a. Locate the "vmware-driver" section under the "ibm-openstack" section.
  b. Replace the "network" section under the "vmware-driver" section with the following details:

  ```
  "use_nsx": true,
  ```

  ```
  "nsx": {
       "version": "version number of the NSX Manager",
       "manager_uri": "URI to access the NSX Manager",
       "user": "NSX Manager admin user name",
       "password": "NSX Manager admin user password",
       "datacenter_moid": "Managed Object ID of the target datacenter",
       "cluster_moid": "Managed Object ID of the cluster",
       "resource_pool_id": "Managed Object ID of the resource pool",
       "datastore_id": "Managed Object ID of the datastore",
       "external_network": "Managed Object ID of the port group on which
                            the external network will be created",
       "vdn_scope_id": "Managed Object ID of the VDN scope",
       "dvs_id": "Managed Object ID of the distributed virtual switch that
                  will be used to provision the Neutron networks"
  },
  ```

  c. Upload the environment file by using the command knife environment from file your-environment-name.json.
  d. Deploy the topology by using the command knife os manage deploy topology your-topology-name.json.

What to do next

Complete the installation of IBM Cloud Orchestrator.

Configuring IBM Cloud Manager with OpenStack for HTTPS

You can configure IBM Cloud Manager with OpenStack for a secure communication. This is a prerequisite procedure for enabling HTTPS on IBM Cloud Manager with OpenStack.

Before you begin

Configure IBM Cloud Manager with OpenStack for secure communication. For more information, see Customize deployment for secure communications.

If you upgraded IBM Cloud Manager with OpenStack in a non-high availability environment, log in to the OpenStack Controller node and run the following commands to delete all V3 endpoints. The endpoints will be added again when the configuration procedure is completed.

```
source /root/v3rc
openstack endpoint list | grep v3
openstack endpoint delete <endpoint-id>
```

About this task

Starting from version 4.3.0.7, IBM Cloud Manager with OpenStack provides complete support for configuring the communication across OpenStack components in HTTPS. This can be configured at the time of installation or later through an upgrade. If the IBM Cloud Manager with OpenStack is configured for HTTPS communication, then IBM Cloud Orchestrator also needs to be configured to communicate to it in HTTPS.

The certificate bundle is generated at IBM Cloud Manager with OpenStack after executing the procedure mentioned in Customize deployment for secure communications. By default, it is stored on the master controller node (that is the controller node of the first region that is installed which hosts the keystone service, or for high availability configuration the primary controller of the first region) and it is stored at /etc/pki/tls/icm/certs/ca_bundle.pem. This certificate bundle must be copied to the IBM Cloud Orchestrator Server.

To enable HTTPS during fresh installation or upgrade of IBM Cloud Orchestrator, do the following configuration steps:

Procedure

1. Set the protocol to HTTPS as described in "Setting the deployment parameters" on page 60.
2. Copy the /etc/pki/tls/icm/certs/ca_bundle.pem file to the <TMP_DIR>/openstack.crt file as described in "Adding the certificate bundle of IBM Cloud Manager with OpenStack to IBM Cloud Orchestrator Server " on page 66, where <TMP_DIR> is the temporary directory defined in the ico_install.rsp response file.

Preparing for the IBM Cloud Orchestrator installation

When you have completed your planning, prepare for the IBM Cloud Orchestrator installation by completing the following tasks.

Downloading the required image files

Before you install IBM Cloud Orchestrator, download the required image files.

About this task

Download the required image files to a temporary download directory on the IBM Cloud Orchestrator Server.

In this procedure, the example download directory is /opt/ico_download and the example installation directory is /opt/ico_install/2.5.0-CSI-ICO-FP00010. Replace these values with the appropriate values for your installation.

Procedure

1. Log on to the IBM Cloud Orchestrator Server as a root user.
2. Create a temporary download directory:

```
mkdir /opt/ico_download
```

3. Download the IBM Cloud Orchestrator V2.5 Fix Pack 10 from Fix Central to the /opt/ico_download directory. The package file name is 2.5.0-CSI-ICO-FP00010.tgz.

   For the complete list of all part numbers for IBM Cloud Orchestrator, see Passport Advantage eAssemblies list at http://www-01.ibm.com/support/docview.wss?uid=swg27045668.
4. Download the following IBM Business Process Manager packages from the IBM Passport Advantage® site to the /opt/ico_download directory:

```
BPM_V86_Linux_x86_1_of_3.tar.gz
BPM_V86_Linux_x86_2_of_3.tar.gz
BPM_V86_Linux_x86_3_of_3.tar.gz
```

5. Download the following IBM HTTP Server packages from the IBM Passport Advantage site to the /opt/ico_download directory:

```
WAS_V8.5.5_SUPPL_1_OF_3.zip
WAS_V8.5.5_SUPPL_2_OF_3.zip
WAS_V8.5.5_SUPPL_3_OF_3.zip
```

6. Unpack the 2.5.0-CSI-ICO-FP00010.tgz file from the download directory into the new /opt/ico_install/2.5.0-CSI-ICO-FP00010 install directory by running the following command:

```
tar -xvf /opt/ico_download/2.5.0-CSI-ICO-FP00010.tgz -C /opt/ico_install
```

7. Change directory to the installation subdirectory where the extracted contents are located:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010
   ```

8. Copy the IBM Business Process Manager files from the temporary download directory to the specified subdirectory:

   ```
   cp /opt/ico_download/BPM_V86_Linux_x86*.tar.gz ./data/orchestrator-chef-repo/
   packages/bpm_binaries
   ```

9. Copy the IBM HTTP Server files from the temporary download directory to the specified subdirectory:

   ```
   cp /opt/ico_download/WAS_V8.5.5_SUPPL_*.zip ./data/orchestrator-chef-repo/
   packages/ihs_binaries
   ```

10. If you want to install IBM Cloud Orchestrator with high-availability capabilities, download the following IBM Tivoli System Automation for Multiplatforms packages:

    - From IBM Passport Advantage, download `SA_MP_v4.1_Lnx.tar` and copy it to `/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo/packages/samp/` directory.

11. If you want to install IBM Cloud Orchestrator Enterprise Edition, perform the following additional steps:

    a. Download the following file from the IBM Passport Advantage site to the `/opt/ico_download` directory:

    ```
    ICO_Ent_V250_1of4.tar
    ```

    b. Create a temporary directory by running the following command, for example:

    ```
    mkdir -p /opt/ico_download/ICOEnterprise
    ```

    c. Unpack the `ICO_Ent_V250_1of4.tar` file from the download directory into the new temporary directory by running the following command:

    ```
    tar -xvf /opt/ico_download/ICO_Ent_V250_1of4.tar -C /opt/ico_download/ICOEnterprise
    ```

    d. Copy the `license` directory to the `/opt/ico_install/2.5.0-CSI-ICO-FP00010` directory by running the following command:

    ```
    cp --preserve /opt/ico_download/ICOEnterprise/license/* \
       /opt/ico_install/2.5.0-CSI-ICO-FP00010/license/
    ```

    Answer `yes` if you are prompted to overwrite any file.

12. If you want to install or upgrade the additional components for IBM Cloud Orchestrator Enterprise Edition, see the Download Document for a list of the images to be downloaded for the following products:

    - IBM SmartCloud Cost Management
    - IBM Tivoli Monitoring
    - IBM Tivoli Monitoring for Virtual Environments
    - Jazz® for Service Management

    For information about installing IBM Cloud Orchestrator Enterprise Edition, see "Installing IBM Cloud Orchestrator Enterprise Edition" on page 70.

Preparing the IBM Cloud Orchestrator Servers

Before you start the installation process, you must prepare one or two virtual machines or physical servers to install IBM Cloud Orchestrator, depending on the topology you chose.

Procedure

1. Ensure that your systems meet the hardware requirements as specified in "Checking the hardware prerequisites" on page 21.
2. Ensure that your systems meet the software requirements as specified in "Checking the software prerequisites" on page 22.
3. If you want to install IBM Cloud Orchestrator as a nonroot user, you must set the **umask** permission to 0022.
4. Ensure that the /home partition (or the / partition, if you do not have a separate /home partition) is *not* mounted with the nodev or nosuid options.

   IBM DB2 requires some setuid binaries that must be installed in the /home partition. If the partition is not mounted correctly, the IBM DB2 part of the server installation fails with the following error message:

   ```
   SQL1641N The db2start command failed because one or more DB2 database manager program
   files was prevented from executing with root privileges by file system mount settings.
   ```

   To check that the partition is mounted correctly, review the options that are listed for the file system in the /etc/fstab file. The following example shows the correct options for the /home mount point:

   ```
   <device_name> /home ext4 defaults 1 2
   ```

   The default options include dev and suid.
5. Ensure that Network Information System (NIS) is not enabled in your environment.

   To check whether NIS is enabled, run the following command:

   ```
   grep ^passwd /etc/nsswitch.conf
   ```

   If the command output includes a nis entry, NIS is enabled. If NIS is enabled, security requirements in the NIS (or LDAP) configured domain might prevent the local user account from being created properly when you try to create IBM Cloud Orchestrator users. Do not continue with the IBM Cloud Orchestrator installation until NIS is disabled by the system administrator.
6. Ensure that your system firewall is configured to open the ports required by IBM Cloud Orchestrator. For information about the required ports, see "Ports used by IBM Cloud Orchestrator" on page 80.
7. Ensure that the hardware clock is configured as UTC (Coordinated Universal Time). Run the **timedatectl status** command and verify that the same value is shown in the **Universal time** field and the **RTC time** field, as shown in the following example:

   ```
   # timedatectl status

         Local time: Mon 2015-06-22 13:35:05 IST
     Universal time: Mon 2015-06-22 12:35:05 UTC
           RTC time: Mon 2015-06-22 12:35:05
           Timezone: Europe/Dublin (IST, +0100)
        NTP enabled: yes
   NTP synchronized: yes
    RTC in local TZ: no
         DST active: yes
    Last DST change: DST began at
                     Sun 2015-03-29 00:59:59 GMT
                     Sun 2015-03-29 02:00:00 IST
    Next DST change: DST ends (the clock jumps one hour backwards) at
                     Sun 2015-10-25 01:59:59 IST
                     Sun 2015-10-25 01:00:00 GMT
   ```

If the **Universal time** and **RTC time** fields contain different values, run the following command:

```
timedatectl set-local-rtc 0
```

8. If you are using a virtual machine, ensure that the IBM Cloud Orchestrator Server clock is synchronized with the OpenStack Controller clock.
9. If you do not want to install IBM Cloud Orchestrator using the `root` account, set up another account with sudo access. For more information, see Configuring sudo access.
10. Create a backup of the IBM Cloud Orchestrator Servers. For example, if they are virtual servers, create the snapshots.
11. [Optional] When installing IBM Cloud Orchestrator with a Single-Server with external database topology or a Dual-Server high-availability with external database topology, before running the procedure in Checking the installation prerequisites, you can perform the following steps to set the credentials to access the external database server or the secondary server in the high-availability topology when running the `prereq-checker.sh` script:
    a) Log on to the IBM Cloud Orchestrator Server node by using root user.

    b) Ensure that the Expect package is installed. To install the Expect package, run the following command: `yum install expect`

    c) Change to the directory where the installer scripts are located: `cd /opt/ico_install/V2503/installer`

    d) Run the following command: `configure_ssh_passwordless.sh -h <remote_hostname> -u <username> -p <password> [-q SilentMode]` where

    `<remote_hostname>` is the host name or the IP address of the external database server or the secondary server in the high-availability topology.

    `<username>` and **`<password>`** are the credentials of a superuser or root user to access the specified server. The `configure_ssh_passwordless.sh` script can only be used in environments where existing password contains only a combination of `a-z A-Z 0-9 _` characters.
12. For OpenStack keystone endpoint on HTTPS, do the following steps:
    a) Copy SSL certificate from OpenStack and add it in <TMP_DIR> of IBM Cloud Orchestrator server.

    b) Rename certificate as `openstack.crt`.

    To configure IBM Cloud Orchestrator with OpenStack endpoint on HTTPS, the OpenStack certificate must be generated by using the host name.

    For IBM Cloud Orchestrator in HA topology, run steps "12.a" on page 30 and "12.b" on page 30 on both nodes.

[Optional] Creating the Business Process Manager databases on an external IBM DB2 server

If you want to use an existing external IBM DB2 database server, you must create the Business Process Manager databases before you install IBM Cloud Orchestrator.

Before you begin

Before you can copy the `create_dbs.sh` script, you must first extract the contents of the IBM Cloud Orchestrator image file, as described in "Downloading the required image files" on page 27.

In this procedure, the example installation directory on the IBM Cloud Orchestrator Server is `/opt/ico_install/2.5.0-CSI-ICO-FP00010`, and the example scripts directory on the external IBM DB2 server is `/opt/ico_scripts`. Replace these values with the appropriate values for your installation.

Procedure

1. Log on to the IBM DB2 database server as a root user.

2. Create a directory to store the scripts:

   ```
   mkdir /opt/ico_scripts
   ```

3. Transfer the following file from the `/opt/ico_install/2.5.0-CSI-ICO-FP00010/installer` directory on the IBM Cloud Orchestrator Server to the `/opt/ico_scripts` directory on the IBM DB2 database server:

   ```
   create_dbs.sh
   ```

   To run the `create_dbs.sh` script, use BASH V4.2 or above.

4. Export the environment variables to store the database user name and password, and create the Business Process Manager databases:

   ```
   export DB2_INSTANCE_USER=db2inst1
   export USER_DEFAULT_PWD=password
   ./create_dbs.sh central
   ```

   where `db2inst1` is the IBM DB2 instance owner user name and `password` is the password that you want to use for the Business Process Manager database users.

5. To apply the changes that are done by the `create_dbs.sh` script to the database management configuration parameters, you must restart the database manager. Because this operation affects all the databases in the IBM DB2 instance, ensure to have an appropriate time window to restart the IBM DB2 instance.

   To restart the IBM DB2 instance, run the following commands as user `db2inst1` on the external database server:

   ```
   db2 force applications all
   db2stop force
   db2start
   ```

Configuring external DB2 for TLS v1.2

For external DB2 instances, you must customize SSL certificates and then configure DB2 for TLSv1.2.

Procedure

1. Log in to the external DB2 database.
2. Customize SSL certificates:

   For the SSL/TLS communication with DB2, you must manage digital certificates for the database server. These certificates and encryption keys are used to establish the SSL connections. By default, the certificates are added with label `dbserver` to the key database at `/home/db2inst1/ssl_server/dbserver.kdb`, where `/home/db2inst1` is the DB2 installation directory on the database server.

   a) Create directory `/home/db2inst1/ssl_server` if not present.

   b) Copy `dbserver.kdb` and `dbserver.sth` from primary IBM Cloud Orchestrator location, `<ICO_installer_path>/data/orchestrator-chef-repo/chef-repo/cookbooks/db2/files/default/`, to `/home/db2inst1/ssl_server/`.

      Note: If keystores (`dbserver.kdb` and `dbserver.sth`) exists in `/home/db2inst1/ssl_server`, then no need to replace.

   c) Modify the owner and group to `db2inst1:db2iadm1` for the following directory and files:

      - `ssl_server` Directory
      - `dbserver.kdb` File
      - `dbserver.sth` File

3. Configure DB2 for TLSv1.2:

a) Check the DB2 configuration parameters to ensure that the connection concentrator is not active on that instance. If the connection concentrator is running, then the SSL support is not enabled on the DB2 instance. To check whether the connection concentrator is active, run the following commands as an instance owner:

```
su - db2inst1 db2 GET DATABASE MANAGER CONFIGURATION | grep MAX_CONNECTIONS
```

If the **max_connections** configuration parameter is set to a value greater than the value of the **max_coordagents** parameter, then the connection concentrator is activated.

b) Set up your DB2 server for SSL support:

1) Log in as the DB2 instance owner.

2) Set the following configuration parameters and the DB2COMM registry variable:

a) Set the **SSL_SVR_KEYDB** configuration parameter to the fully qualified path of the key database file. For example:

```
db2 update dbm cfg using SSL_SVR_KEYDB  /home/db2inst1/ssl_server/dbserver.kdb
```

b) Set the **SSL_SVR_STASH** configuration parameter to the fully qualified path of the stash file. For example:

```
db2 update dbm cfg using SSL_SVR_STASH /home/db2inst1/ssl_server/dbserver.sth
```

c) Set the **SSL_SVR_LABEL** configuration parameter to the label of the digital certificate of the server. If **SSL_SVR_LABEL** is not set, then the default certificate in the key database is used. In the absence of a default certificate in the key database, SSL is not enabled.

```
db2 update dbm cfg using SSL_SVR_LABEL dbserver
```

d) Set the **SSL_SVCENAME** configuration parameter to port 50001, which the DB2 uses for SSL connections:

```
db2 update dbm cfg using SSL_SVCENAME 50001
```

If TCP/IP and SSL are both enabled (the DB2COMM registry variable is set to 'TCPIP, SSL'), you must set **SSL_SVCENAME** to a different port than the port to which **SVCENAME** is set. The **SVCENAME** configuration parameter specifies the port that the DB2 database system listens to for TCP/IP connections. If you set **SSL_SVCENAME** to the same port as **SVCENAME**, then neither TCP/IP nor SSL is enabled. If **SSL_SVCENAME** is not set, then the SSL support is not enabled.

e) Set the **SSL_VERSIONS** configuration parameter to indicate that DB2 must use TLS v1.2 protocol:

```
db2 update dbm cfg using SSL_VERSIONS TLSV12
```

f) Add the SSL value to the DB2COMM registry variable. For example,

```
db2set -i db2inst1 DB2COMM=SSL,TCPIP
```

g) Check that all the configuration parameters are set correctly by running the following command:

```
db2 get dbm cfg | grep SSL
```

h) Restart the DB2 instance:

```
db2stop force
db2start
```

Configuring the OpenStack servers

To configure the OpenStack servers for your IBM Cloud Orchestrator environment, run one of the following procedures according to whether you are using OpenStack Refcompliant Mitaka / Ocata / Queens release or IBM Cloud Manager with OpenStack V4.3 or external OpenStack Kilo.

[Typical] Configuring the IBM Cloud Manager with OpenStack servers

After you install IBM Cloud Manager with OpenStack Kilo release, you must configure the IBM Cloud Manager with OpenStack installation for IBM Cloud Orchestrator. Follow this procedure if you installed IBM Cloud Manager with OpenStack using a prescribed topology for KVM or VMware.

Before you begin

Ensure that your IBM Cloud Manager with OpenStack installation meets the software requirements as described in "Checking OpenStack prerequisites" on page 17.

About this task

This procedure is for an IBM Cloud Manager with OpenStack installation using prescribed KVM and VMware topologies. For information about prescribed configurations, see Deploying with KVM or QEMU compute nodes and Deploying with VMware compute nodes.

To prepare your IBM Cloud Manager with OpenStack servers, run the following procedure.

Procedure

1. "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 33.
2. "Configuring IBM Cloud Manager with OpenStack for IBM Cloud Orchestrator" on page 35.
3. "[Optional] Configuring security groups" on page 36.
4. [For IBM Cloud Orchestrator Enterprise Edition only:] If you are using SmartCloud Cost Management, *Configuring IBM Cloud Orchestrator for metering section of SCCM User's Guide.*

What to do next

After you prepare the IBM Cloud Manager with OpenStack servers, continue the IBM Cloud Orchestrator installation by following the procedure in "Setting the deployment parameters" on page 60.

Copying the IBM Cloud Orchestrator scripts to the OpenStack servers

IBM Cloud Orchestrator scripts must be run on each IBM Cloud Manager with OpenStack controller and compute node.

Before you begin

Before copying the scripts, you must extract the contents of the IBM Cloud Orchestrator image file, as described in "Downloading the required image files" on page 27.

In this procedure, the installation directory on the IBM Cloud Orchestrator Server is `/opt/ico_install/2.5.0-CSI-ICO-FP00010`, and the scripts directory on the IBM Cloud Manager with OpenStack servers is `/opt/ico_scripts`.

Procedure

1. Identify the IBM Cloud Manager with OpenStack master controller server where the Keystone service is running. This is normally the OpenStack Controller of the first region that was installed.

   a. Log on as `root` to the controller server of the active controller and run the following command:

   ```
   cat ~/openrc | grep OS_AUTH_URL | cut -d "/" -f3 | cut -d ":" -f1
   ```

   b. For high-availability OpenStack, identify which node is active by running the following command:

      ```
      pcs status
      Cluster name: KVM_RegionOne
      Last updated: Mon Apr 11 14:08:31 2016
      Last change: Mon Apr 11 12:46:40 2016
      Stack: corosync
      Current DC: abc.123.cloud.com (1) - partition with quorum
      ```

      In this cluster, the `abc.123.cloud.com` host is the active node because it has quorum.

2. Identify all other IBM Cloud Manager with OpenStack controller and compute nodes. These servers are the OpenStack Controllers in a multiregion installation.

   a. Log on to the master controller as `root` and run the following commands to find the controller servers:

      ```
      source ~/openrc; keystone endpoint-list | grep $(keystone service-list \
          | awk '/nova/ {print $2}') | cut -d "/" -f3 | cut -d ":" -f1
      ```

   b. If you already converted your environment to the Keystone V3 API, then you must run a different command:

      ```
      source ~/v3rc; openstack endpoint list | grep nova \
          | awk '/admin/ {print $14}' | cut -d "/" -f3 | cut -d ":" -f1
      ```

      Then run the following command to find the compute nodes:

      ```
      nova service-list | awk '/nova-compute/ {print $6}'
      ```

3. Copy the scripts by completing the following steps for each OpenStack server that you identified in steps "1" on page 33 and "2" on page 34:

   a. Log on to the server as a `root` user.

   b. Create a directory to store the scripts:

      ```
      mkdir /opt/ico_scripts
      ```

   c. Copy all the files from the `/opt/ico_install/2.5.0-CSI-ICO-FP00010/utils/scripts` directory on the IBM Cloud Orchestrator Server to the `/opt/ico_scripts` directory on the IBM Cloud Manager with OpenStack servers.

      For example:

      ```
      scp root@[ICO server]:/opt/ico_install/2.5.0-CSI-ICO-FP00010/utils/scripts/* /opt/
      ico_scripts
      ```

   d. Copy the `sco_horizon.zip` file from the `/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo/packages/sco_horizon` directory on the IBM Cloud Orchestrator Server to the `/opt/ico_scripts` directory on the IBM Cloud Manager with OpenStack servers.

      For example:

      ```
      scp root@[ICO server]:/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo\
      /packages/sco_horizon/sco_horizon.zip /opt/ico_scripts
      ```

Results

Scripts are available on all the IBM Cloud Manager with OpenStack servers to be run.

Configuring IBM Cloud Manager with OpenStack for IBM Cloud Orchestrator

This topic describes how to configure IBM Cloud Manager with OpenStack for IBM Cloud Orchestrator.

Before you begin

Ensure that you copied the scripts from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 33.

Ensure that the `unzip` program is installed or that a `yum` repository is configured in all the IBM Cloud Manager with OpenStack controllers where the Horizon service is installed.

About this task

Run the `ICM_config_ico.sh` script in the IBM Cloud Manager with OpenStack environment to perform the following operations:

- Create the following roles, users, and projects in IBM Cloud Orchestrator:

Roles:	`netadmin`
	`sysadmin`
	`domain_admin`
	`catalogeditor`
	`member`

Users:	`demo` `domadmin`
Projects:	`demo`

  The access privileges are granted as follows:

  - The `demo` user is granted the `netadmin`, `sysadmin`, and `catalogeditor` roles on the `demo` project.
  - The `domadmin` user is granted the `domain_admin` role on the `admin` project.
  - The user `admin` is granted the `member` role on the `admin` project.

- Enable OpenStack V3 functions that provide capabilities like domain, extended policy, and LDAP in OpenStack.
- Install IBM Cloud Orchestrator extensions in Horizon that provide the following capability in the OpenStack Dashboard:
  - Support for availability zones on domains and projects.
  - Support for quotas on domains.
  - VMware region only: reassignment of on boarded virtual machines to other projects.
  - Creation of a default project when a domain is created.
  - Ability to add domain administrators to the default project of a domain.

The script output is stored in the `ICO-reconfigure.log` file that you can check if any problem occurs.

Note: You must run this procedure every time that IBM Cloud Manager with OpenStack is updated. For more information, see "Reconfiguring IBM Cloud Manager with OpenStack after updates" on page 71.

Procedure

1. Log on to the master OpenStack Controller as a `root` user. The master OpenStack Controller is the controller node in all-in-one systems and in systems with a single controller, and it is the primary controller in environments with multiple controllers.

   If IBM Cloud Manager with OpenStack is installed in a high-availability topology, log on to the active IBM Cloud Manager with OpenStack high-availability controller.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

```
cd /opt/ico_scripts
```

3. Run the following command:

   ```
   ./ICM_config_ico.sh master_controller
   ```

   Take note of the Identity (Keystone) public URL that is included in the command output. For example, `"http://192.0.2.67:5000/v3"`.
4. For each secondary OpenStack Controller in environments with multiple controllers, or for each IBM Cloud Manager with OpenStack high-availability controllers with the exception of the active high-availability controller (identified in step "1" on page 35) in a high-availability environment, perform the following steps:
   a. Log on as a `root` user.
   b. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

   c. Run the following command:

   ```
   ./ICM_config_ico.sh controller
   ```
5. For each Compute Node in your environment, perform the following steps:
   a. Log on as a `root` user.
   b. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

   c. Run the following command:

   ```
   ./ICM_config_ico.sh compute <OS_AUTH_URL>
   ```

   where *<OS_AUTH_URL>* is the Identity (Keystone) public URL included in the output of step "3" on page 36. For example:

   ```
   ./ICM_config_ico.sh compute "http://192.0.2.67:5000/v3"
   ```

Results
IBM Cloud Manager with OpenStack is now configured to be used with IBM Cloud Orchestrator.

[Optional] Configuring security groups

You can configure security groups to allow **ssh** and **ping** commands to access virtual machines that are deployed in your OpenStack environment. You can also configure security groups to enable RDP ports for Windows virtual machine instances.

About this task
This task is optional. Consider whether you want to allow such access. For more information, see the IBM Cloud Manager with OpenStack documentation.

Note: If you a working in a multi-domain environment, you must run this procedure for each domain in which you want to access the deployed virtual machines. Specify the domain in the RC file used in the step "2" on page 36 of the procedure.

Procedure

1. Log on to the OpenStack Controller as a `root` user.
2. Set the environment to the correct OpenStack values and run the following command:

   ```
   source /root/openrc
   ```

3. If you want to configure security groups to allow access by **ssh** and **ping** commands, run the following commands:

```
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```

4. If you want to enable RDP ports for Windows virtual machine instances, run the following commands:

```
nova secgroup-add-rule default tcp 3389 3389 0.0.0.0/0
nova secgroup-add-rule default udp 3389 3389 0.0.0.0/0
```

[Advanced] Configuring the IBM Cloud Manager with OpenStack servers

After you install IBM Cloud Manager with OpenStack Kilo release, you must configure the IBM Cloud Manager with OpenStack installation for IBM Cloud Orchestrator. Follow this procedure if you did an advanced IBM Cloud Manager with OpenStack installation and you did not use a prescribed topology for KVM or VMware.

About this task

The commands and scripts that are mentioned in the following topics are sample commands and sample scripts that are provided with IBM Cloud Orchestrator and that are done for an IBM Cloud Manager with OpenStack installation using a prescribed topology for KVM or VMware. If you did not use a prescribed topology, you might need to modify these scripts for your installation. For information about prescribed configurations, see Deploying with KVM or QEMU compute nodes and Deploying with VMware compute nodes.

IBM Cloud Orchestrator uses the simple token enhancement that is already part of IBM Cloud Manager with OpenStack. Before you install IBM Cloud Manager with OpenStack, see the IBM Cloud Manager with OpenStack documentation for information about how to configure the simple token.

The samples scripts and additional files can be found in the `/opt/ico_install/2.5.0-CSI-ICO-FP00010/utils/scripts` directory after you extract the contents of the IBM Cloud Orchestrator image files, as described in "Downloading the required image files" on page 27.

To prepare your IBM Cloud Manager with OpenStack servers, run the following procedure.

Note: You must run this procedure with the exception of step "3" on page 37 every time that IBM Cloud Manager with OpenStack is updated. For more information, see "Reconfiguring IBM Cloud Manager with OpenStack after updates" on page 71.

Procedure

1. "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 33.
2. "Changing state of the OpenStack services" on page 38.
3. "Adding roles, users, and projects to Keystone" on page 38.
4. "Installing the IBM Cloud Orchestrator extensions for Horizon" on page 39.
5. "Configuring V3 API endpoints for Keystone" on page 40.
6. "Configuring the OpenStack services to use the Keystone V3 API" on page 41.
7. "Starting all the OpenStack services and restarting Keystone" on page 43.
8. "[Optional] Configuring security groups" on page 36.
9.
10. [For IBM Cloud Orchestrator Enterprise Edition only:] If you are using SmartCloud Cost Management, see *Configuring IBM Cloud Orchestrator for metering topic in SCCM User's Guide*.

What to do next

After you prepare the IBM Cloud Manager with OpenStack servers, continue the IBM Cloud Orchestrator installation by following the procedure in "Setting the deployment parameters" on page 60.

Changing state of the OpenStack services

For a non-high availability installation, when configuring IBM Cloud Manager with OpenStack, stop all the OpenStack services to ensure that there are no ongoing activities. Only Keystone and all middleware services must still be running as Keystone is accessed during the reconfiguration process.

You must stop the services on all the OpenStack servers only, but the Keystone service must still be running.

For information about managing services in IBM Cloud Manager with OpenStack, see Managing IBM Cloud Manager with OpenStack services.

IBM Cloud Orchestrator provides a sample script that is named `control_services.sh` to easily stop the services for IBM Cloud Manager with OpenStack. Review the script, adapt it to your environment if needed, and run the following command on all the IBM Cloud Manager with OpenStack servers.

```
sh ./control_services.sh -o stop
```

This command stops all the services on all the nodes.

For an high availability, you must put pacemaker in maintenance mode by running the following command:

```
pcs property set maintenance-mode=true
```

Important: In a high availability installation, the IBM Cloud Manager with OpenStack services must not be in use during the time of script execution. The IBM Cloud Manager with OpenStack services must remain active to avoid synchronization errors.

Adding roles, users, and projects to Keystone

Create the roles, users, and projects that are required by IBM Cloud Orchestrator, and add them to OpenStack Keystone.

Before you begin

Ensure that you copied the `configure_ico_roles.sh` script from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 33.

Review the changes that are made by the script before running it.

About this task

You must run the `configure_ico_roles.sh` script on the OpenStack server where the Keystone service is running. If IBM Cloud Manager with OpenStack is installed in a high-availability topology, you must run this script only on the active IBM Cloud Manager with OpenStack high-availability controller and the changes will be reflected on the other IBM Cloud Manager with OpenStack high-availability controllers.

Note: You must run this script only once. Do not run the script again when IBM Cloud Manager with OpenStack is updated.

The script creates the following roles, users, and projects in IBM Cloud Orchestrator:

Roles	netadmin sysadmin domain_admin catalogeditor member
Users	demo domadmin
Projects	demo

The access privileges are granted as follows:

- The demo user is granted the demo role on the demo project.
- The domadmin user is granted the domain_admin role on the admin project.
- The user admin is granted the member role on the admin project.

Note: The Keystone collector synchronizes information every night and hence transparently handles the inclusion of new projects or domains.

In this procedure, the example scripts directory on the OpenStack Controller is /opt/ico_scripts. Replace this value with the appropriate value for your installation.

Complete the following steps:

Procedure

1. Log on to the OpenStack Controller as a root user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Set the environment to the correct OpenStack values and run the following command:

   ```
   source /root/openrc
   ```

   If the /root/openrc file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v2.0
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   ```

4. Run the script:

   ```
   ./configure_ico_roles.sh
   ```

Installing the IBM Cloud Orchestrator extensions for Horizon

To support the IBM Cloud Orchestrator functionality, you must extend the base OpenStack Horizon software to provide additional options in the OpenStack Dashboard.

Before you begin

Ensure that you copied the ICM_configure_ico_horizon_extensions.sh script from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 33.

Ensure that you set the default file permissions for the ICM_configure_ico_horizon_extensions.sh script to a value that is at least 0022.

Depending on your installation and OpenStack distribution, you may need to adapt the script.

Ensure that the unzip program is installed or that a yum repository is configured in all the IBM Cloud Manager with OpenStack controllers where the Horizon service is installed.

For IBM Cloud Manager with OpenStack, the self-service portal extensions must not be installed. For information about uninstall and disable it, see Uninstalling the self-service user interface on Linux.

About this task

The IBM Cloud Orchestrator extensions for Horizon provide the following capability in the OpenStack Dashboard:

- Support for availability zones on domains and projects
- Support for quotas on domains
- Reassignment of onboarded virtual machines to other projects
- Creation of a default project when a domain is created
- Ability to add domain administrators to the default project of a domain

In this procedure, the example scripts directory on the OpenStack Controller is `/opt/ico_scripts`. Replace this value with the appropriate value for your installation.

Complete the following steps on each server where the Horizon service is installed.

Procedure

1. Log on to the OpenStack Controller as a `root` user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. If you are running this procedure for the first time, set the environment to the correct OpenStack values and run the following command:

   ```
   source /root/openrc
   ```

   If the `/root/openrc` file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v2.0
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   ```

   Note: If you are running again this procedure because you updated IBM Cloud Manager with OpenStack, to set the environment to the correct OpenStack values, run the following command:

   ```
   source /root/v3rc
   ```

4. Review the `ICM_configure_ico_horizon_extensions.sh` script and adapt it to your installation if needed. Then run the following command:

   ```
   ./ICM_configure_ico_horizon_extensions.sh
   ```

Results

When you log in to the OpenStack Dashboard, you can work with the IBM Cloud Orchestrator extensions.

Configuring V3 API endpoints for Keystone

You must create Keystone V3 endpoints for IBM Cloud Orchestrator.

Before you begin

Ensure that the IBM Cloud Orchestrator extensions for Horizon are installed on each OpenStack Controller, as described in "Installing the IBM Cloud Orchestrator extensions for Horizon" on page 39.

Ensure that you copied the `configure_endpoints.sh` script from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 33.

About this task

You must run the `configure_endpoints.sh` on the OpenStack Controller where the Keystone service is running.

If IBM Cloud Manager with OpenStack is installed in a high-availability topology, the Keystone service is installed on several high-availability OpenStack Controllers. You must run the script on each high-availability OpenStack Controller.

The script creates endpoints for the Keystone V3 API in all the regions.

In this procedure, the example scripts directory on the OpenStack Controller is `/opt/ico_scripts`. Replace this value with the appropriate value for your installation.

Procedure

1. Log on to the OpenStack Controller as a `root` user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Set the environment to the correct OpenStack values and run the following command:

   ```
   source /root/openrc
   ```

   If the `/root/openrc` file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v2.0
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   ```

4. Run the script:

   ```
   ./configure_endpoints.sh
   ```

Results

The Keystone V3 endpoints are created.

Configuring the OpenStack services to use the Keystone V3 API

After the endpoints are configured to Keystone V3, you must configure the other OpenStack services to use the Keystone V3 API and to use the new Keystone V3 endpoints. You must configure the services on all the OpenStack servers in your installation.

Before you begin

Ensure that the Keystone endpoints are configured for Keystone V3, as described in "Configuring V3 API endpoints for Keystone" on page 40.

About this task

When you run the script in the following procedure, the following changes are made. Depending on your OpenStack distribution, you might need to adapt the script or to do the changes manually:

- The configuration files of the OpenStack services contains a section like, for example:

  ```
  [keystone_authtoken]
  auth_uri = http://192.0.2.67:5000/v2.0
  identity_uri = http://192.0.2.67:35357/
  auth_version = v2.0
  admin_tenant_name = service
  admin_user = nova
  admin_password = W0lCTTp2MV1iY3JhZmducHgtcGJ6Y2hncg==
  signing_dir = /var/cache/nova/api
  hash_algorithms = md5
  insecure = false
  ```

  The following parameters must be changed to use the new V3 endpoints:

```
auth_uri = http://192.0.2.67:5000/v3
```

```
auth_version = v3
```

- In the keystone.conf file, the following parameter must be set:

  ```
  [auth]
  external = keystone.auth.plugins.external.Domain
  ```

- In the Cinder api-paste.ini file, the following section must be added:

-
  ```
  [keystone_authtoken]
  auth_uri = http://192.0.2.67:5000/v3
  identity_uri = http://192.0.2.67:35357/
  auth_version = v3
  admin_tenant_name = service
  admin_user = cinder
  admin_password = W0lCTTp2MV1iY3JhZmducHgtcGJ6Y2hncg==
  signing_dir = /var/cache/nova/api
  hash_algorithms = md5
  ```

- The following changes (in bold) must be done to the Horizon local settings:

  ```
  "identity": 3
  ```

  ```
  OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
  ```

  ```
  OPENSTACK_KEYSTONE_URL = "http://192.0.2.67:5000/v3"
  OPENSTACK_KEYSTONE_ADMIN_URL = "http://192.0.2.67:35357/v3"
  ```

  ```
  OPENSTACK_KEYSTONE_DEFAULT_ROLE = "member"
  ```

  If the following line is not in the configuration file, it must be added:

  ```
  OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
  ```

The script also creates an updated version of the openrc file. This new file is called v3rc and is placed in the same directory as the openrc file.

Procedure

1. Log on to the OpenStack server as a root user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Set the environment to the correct OpenStack values and run the following command:

   ```
   source /root/openrc
   ```

   If the /root/openrc file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v3
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   export OS_IDENTITY_API_VERSION=3
   export OS_USER_DOMAIN_NAME=${OS_USER_DOMAIN_NAME:-"Default"}
   export OS_PROJECT_DOMAIN_NAME=${OS_PROJECT_DOMAIN_NAME:-"Default"}
   ```

   Alternatively, you can provide the OS_AUTH_URL value as command parameter in the following step.
4. Run the script:

```
./ICM_configure_files.sh [<OS_AUTH_URL>]
```

where *<OS_AUTH_URL>* is the Identity (Keystone) public URL that you must specify if you did not set it in the /root/openrc file.

Validation Step

If there is a change in password in IBM Cloud Manager with OpenStack for the admin user, you must update the password in the v3rc file created by IBM Cloud Orchestrator and keep it in sync with the openrc file of IBM Cloud Manager with OpenStack.

Results

All the OpenStack services are configured to use the new Keystone V3 endpoints. A new v3rc file with the environment variables to use with Keystone V3 is created.

Starting all the OpenStack services and restarting Keystone

After the configuration is done, you must start all the OpenStack services. You must also restart the Keystone service.

You must start the services on all the OpenStack servers.

For information about managing services in IBM Cloud Manager with OpenStack, see Managing IBM Cloud Manager with OpenStack services.

You can then individually start all the services by logging on to the servers and running the following command:

```
systemctl start <service>
```

In IBM Cloud Manager with OpenStack, the OpenStack services are all the services that begin with neutron-, openstack- or httpd, which is the Horizon service.

IBM Cloud Orchestrator provides a sample script named control_services.sh to easily start the services for IBM Cloud Manager with OpenStack. Review the script, adapt it to your environment if needed, and run the following commands:

```
./control_services.sh -o restart -k
./control_services.sh -o restart
```

In the IBM Cloud Manager with OpenStack high availability setup, after starting the OpenStack services and restarting the Keystone services, you must put pacemaker out of maintenance mode. In the IBM Cloud Manager with OpenStack high availability setup you must only run the following command on one of the IBM Cloud Manager with OpenStack high availability server, it will be reflected to other IBM Cloud Manager with OpenStack high availability servers automatically:

```
pcs property set maintenance-mode=false
```

Note: You must run the first command on the OpenStack server where the Keystone service is installed, and you must run the second command on all the servers.

Configuring an OpenStack distribution of another vendor

After you install an OpenStack distribution of another vendor, you must configure it for IBM Cloud Orchestrator.

Configuring OpenStack Mitaka / Ocata / Queens distribution

After you install an OpenStack *Mitaka*, *Ocata*, or *Queens* environment, you must configure the OpenStack installation to integrate with IBM Cloud Orchestrator.

About this task

Important: You must be a root user to run all the commands from OpenStack servers.

Procedure

1. Run the following command to source the environment file of your OpenStack distribution before you run any OpenStack command:

   `source <OpenStack distribution environment file>`

   The environment file of your OpenStack distribution must have the following parameters:

   ```
   export OS_USERNAME=<admin_user>
   export OS_PASSWORD=<admin_password>
   export OS_AUTH_URL=http://<keystone_ip>:5000/v3
   export OS_REGION_NAME=<region_name>
   export OS_PROJECT_NAME=admin
   export OS_USER_DOMAIN_NAME=Default
   export OS_PROJECT_DOMAIN_NAME=Default
   export OS_IDENTITY_API_VERSION=3
   ```

   Note:

   - Default domain must be created in BYOOS prior to IBM Cloud Orchestrator installation.
   - The parameter values are only sample values. Replace the values depending on your OpenStack environment.
   - For information about creating your own environment file, see "Creating OpenStack environment files " on page 50.

2. Add roles, users, and projects to Keystone. Run this step only once on the first controller or the master controller.

   a) Create the following IBM Cloud Orchestrator roles in Keystone:

      - `netadmin`
      - `sysadmin`
      - `domain_admin`
      - `catalogeditor`
      - `member`

      For example, run the following command to create a `netadmin` role:

      `openstack role create netadmin`

      Repeat the command for the other IBM Cloud Orchestrator roles.

   b) Assign a default `admin` project to the `admin` user by running the following command:

      `openstack user set --project admin admin`

   c) Verify whether the `member` role on the `admin` project is granted for the user `admin`. If it is not granted, run the following command:

      ```
      openstack role add --project admin --user admin member
      ```

   d) Create the demo project. Create the demo and `domadmin` users and grant roles to these users.

      1) To create the `demo` project, run the following command:

         `openstack project create demo`

      2) Run the following commands to create the `domadmin` and `demo` users:

         ```
         openstack user create domadmin
         openstack user create demo
         ```

      3) Grant the `domain_admin` role for the `domadmin` user on the `admin` project:

         ```
         openstack role add --project admin --user domadmin domain_admin
         ```

4) Grant the `member` role for the demo user on the demo project:

```
openstack role add --project demo --user demo member
```

3. For all the OpenStack regions, configure v3 API endpoints for Keystone.

   Verify whether your OpenStack installation created Keystone v3 endpoints in all regions. If not, you must create Keystone v3 endpoints for each region.

   Verify whether all OpenStack services are already configured to use Keystone API v3. If not, you must configure v3.

   For example, run the following commands for all OpenStack regions from the first controller or the master controller:

```
openstack endpoint create --region <your region name> --enable keystone admin \
http://<openstack_keystone_server>:35357/v3

openstack endpoint create --region <your region name> --enable keystone \
internal http://<openstack_keystone_server>:5000/v3

openstack endpoint create --region <your region name> --enable keystone \
public http://<openstack_keystone_server>:5000/v3
```

   Use the `<openstack_keystone_server>` value based on existing Keystone endpoints of your OpenStack distribution. For example, IP address or host name of the server that hosts the Keystone service.

   Note: If you are using HTTPS configuration, replace HTTP with HTTPS.

4. Install the simple token extension on all OpenStack controller servers where the Keystone service is running:

   a) On all OpenStack controller servers, stop all OpenStack services including Keystone, and take backup of the `keystone.conf` and `keystone-paste.ini` files that are in the `/etc/keystone` directory.

   b) Copy the `/2.5.0-CSI-ICO-FP00010/installer/ico_reconfig/simpletoken.py` file that is released in the IBM Cloud Orchestrator fix pack to the Keystone middleware directory of your OpenStack distribution.

   In case of Queens, copy the `/2.5.0-CSI-ICO-FP00010/installer/ico_reconfig/queens/simpletoken.py` file.

   c) Get the `simple_token_secret` value and use it for `SimpleToken` authentication between IBM Cloud Orchestrator and OpenStack Controller Servers. The secret must be a base64 encoded value.

   - If you have an existing IBM Cloud Orchestrator installation that is integrated with IBM Cloud Manager with OpenStack V4.3 or external OpenStack Kilo, copy the `simple_token_secret` value from `/etc/keystone/keystone.conf` file of your existing installation.
   - If it is a new installation of IBM Cloud Orchestrator, generate a unique value of `simple_token_secret`.

     To generate a secret, run the following command on any Linux server:

```
dd if=/dev/urandom bs=16 count=1 2>/dev/null | base64
```

     Note: Provide the `SimpleToken` that is used here as an input parameter to IBM Cloud Orchestrator installer.

   d) Add the following lines in the `/etc/keystone/keystone.conf` file:

```
[authentication]
simple_token_header = SimpleToken
simple_token_secret = <Value of simple_token_secret>
```

   e) For IBM Cloud Orchestrator multi-tenancy support, make sure that the external authentication is configured in Keystone. Set the following values for the external parameter in the `[auth]` section of the `/etc/keystone/keystone.conf`file:

```
[auth]
external = keystone.auth.plugins.external.Domain
```

If the [auth] section is not already available, create it.

f) Add the following lines to define a SimpleToken filter by editing the /etc/keystone/keystone-paste.ini file:

1) Add the following line in the /etc/keystone/keystone-paste.ini file:

```
[filter:simpletoken]
paste.filter_factory=keystone.middleware.simpletoken:SimpleTokenAuthentication.factory
```

2) Add the SimpleToken filter to the [pipeline:public_api], [pipeline:admin_api], and [pipeline:api_v3] pipelines. You must set the SimpleToken filter after the json_body filter, but before the actual application. For example:

```

[pipeline:public_api]
pipeline = cors sizelimit url_normalize request_id build_auth_context
token_auth json_body simpletoken ec2_extension public_service

[pipeline:admin_api]
pipeline = cors sizelimit url_normalize request_id build_auth_context
token_auth json_body simpletoken ec2_extension s3_extension admin_service

[pipeline:api_v3]
pipeline = cors sizelimit url_normalize request_id build_auth_context
token_auth json_body simpletoken ec2_extension_v3 s3_extension service_v3
```

3) Back up the policy.json file that is in the /etc/keystone/policy.json directory and replace it with the policy.json file that is extracted from the policy_json.tar file. The policy_json.tar file is included in IBM Cloud Orchestrator. After you replace the file, verify whether the permissions of the file are the same as before.

5. Enable the Cinder V1 API on all the controllers where Cinder is installed. In the OpenStack controller server(s), open the Cinder configuration file, cinder.conf, and ensure that the **enable_v1_api** parameter is set to true. Refer to your OpenStack documentation to find the correct file name and its location. Also, ensure it supports Cinder V1 API.

   You can ignore this step if you are using Queens distribution or BYOOS with Cinder V3 distribution.

6. On all OpenStack controller servers, start all OpenStack services including Keystone.

What to do next

After you prepare the OpenStack server, continue the IBM Cloud Orchestrator installation by following the procedure in "Setting the deployment parameters" on page 60.

Configuring Red Hat OpenStack Platform (RHOSP) Queens distribution

Apply the configuration steps to RHOSP 13 before you configure it with IBM Cloud Orchestrator.

About this task

Steps 1 and 2 in the procedure must be performed on RHOSP director node by sourcing overcloudrc file.

Procedure

1. Run the following command to source the overcloudrc file to use OpenStack commands:

```
source overcloudrc
```

2. Add roles, users, and projects to Keystone. Run this step only once on the first controller or the master controller.

   a. Run the following command to create a role:

```
openstack role create <role>
```

Create the following IBM Cloud Orchestrator roles in Keystone:

- netadmin
- sysadmin
- domain_admin
- catalogeditor
- member

For example:

```
openstack role create netadmin
```

b. Run the following command to assign a default admin project to the admin user:

```
openstack user set --project admin admin
```

c. Verify whether the member role on the admin project is granted for the user admin. If it is not granted, then run the following command:

```
openstack role add --project admin --user admin member
```

d. Create the demo project. Create the demo and domadmin users and grant roles to these users.

- To create the demo project, run the following command:

```
openstack project create demo
```

- Run the following commands to create the domadmin and demo users:

```
openstack user create domadmin
        openstack user create demo
```

- Grant the domain_admin role for the domadmin user on the admin project:

```
openstack role add --project admin --user domadmin domain_admin
```

- Grant the member role for the demo user on the demo project:

```
openstack role add --project demo --user demo member
```

Note: As per RHOSP 13 documentation, keystone endpoints are v3 by default. For more information, see https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html-single/users_and_identity_management_guide/index.

3. Run the following steps on keystone container that is located on RHOSP 13 controller:

a) Run the following command to connect to keystone container:

```
sudo docker exec -it keystone /bin/bash
```

b) Install the simple token extension on all OpenStack controller servers where the Keystone container is running.

c) On all OpenStack controller servers, stop all OpenStack services including Keystone, and take backup of the `keystone.conf` and `keystone-paste.ini` files that are in the `/etc/keystone` directory.

d) Copy the `/2.5.0-CSI-ICO-FP00010/installer/ico_reconfig/simpletoken.py` file that is released in the IBM Cloud Orchestrator fix pack to the Keystone middleware directory of your OpenStack distribution.

In case of Queens, copy the `/2.5.0-CSI-ICO-FP00010/installer/ico_reconfig/queens/simpletoken.py` file.

e) Get the simple_token_secret value and use it for SimpleToken authentication between IBM Cloud Orchestrator and OpenStack Controller Servers. The secret must be a base64 encoded value.

If you have an existing IBM Cloud Orchestrator installation that is integrated with IBM Cloud Manager with OpenStack V4.3 or external OpenStack Kilo, copy the simple_token_secret value from /etc/keystone/keystone.conf file of your existing installation.

If it is a new installation of IBM Cloud Orchestrator, then generate a unique value of simple_token_secret. To generate a secret, run the following command on any Linux server:

```
dd if=/dev/urandom bs=16 count=1 2>/dev/null | base64
```

f) Provide the SimpleToken that is used here as an input parameter to IBM Cloud Orchestrator installer.

g) Add the following lines in the /etc/keystone/keystone.conf file:

```
[authentication]
    simple_token_header = SimpleToken
    simple_token_secret = <Value of simple_token_secret>
```

h) For IBM Cloud Orchestrator multi-tenancy support, make sure that the external authentication is configured in Keystone. Set the following values. For the external parameter in the [auth] section of the /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf file on RHOSP controller and make sure that these changes are present in /etc/keystone/keystone.conf inside keystone container:

```
[auth]
    external = keystone.auth.plugins.external.Domain
```

If the [auth] section is not already available, create it. Add the following lines to define a SimpleToken filter by editing the /etc/keystone/keystone-paste.ini file inside keystone container: Add the following line in the /etc/keystone/keystone-paste.ini file:

```
 [filter:simpletoken]

paste.filter_factory=keystone.middleware.simpletoken:SimpleTokenAuthentication.factory
```

i) Add the SimpleToken filter to the [pipeline:public_api], [pipeline:admin_api], and [pipeline:api_v3] pipelines. You must set the SimpleToken filter after the json_body filter, but before the actual application. For example:

```
 [pipeline:public_api]
        pipeline = cors sizelimit url_normalize request_id build_auth_context
        token_auth json_body simpletoken ec2_extension public_service

        [pipeline:admin_api]
        pipeline = cors sizelimit url_normalize request_id build_auth_context
        token_auth json_body simpletoken ec2_extension s3_extension admin_service

        [pipeline:api_v3]
        pipeline = cors sizelimit url_normalize request_id build_auth_context
        token_auth json_body simpletoken ec2_extension_v3 s3_extension service_v3
```

j) Back up the policy.json file that is in the /etc/keystone/policy.json directory inside keystone container and replace it with the policy.json file that is extracted from the policy_json.tar file. The policy_json.tar file is included in IBM Cloud Orchestrator. After you replace the file, verify whether the permissions of the file are the same as before.

4. To manually copy the simple_token_secret value from the keystone container to ico_install.rsp, log in to the keystone container from the controller and run the following commands:

```
docker exec -it keystone /bin/bash
cd /etc/keystone/
cat /etc/keystone/keystone.conf | grep simple*
simple_token_header = SimpleToken
simple_token_secret = <value>
```

Additional OpenStack Ref compliant *Mitaka* or *Ocata* or *Queens* configuration

These additional configuration for OpenStack Ref compliant *Mitaka* / *Ocata* / *Queens* are optional.

Configuring role or user access to deploy heat stacks

Due to changes in the default heat configuration of the OpenStack Mitaka / Ocata / Queens release, you can optionally configure access for IBM Cloud Orchestrator users with member role such that they can deploy heat stacks.

Procedure

1. Edit the property **`trusts_delegated_roles`** in `/etc/heat/heat.conf` to include a role that has IBM Cloud Orchestrator users who deploy heat stacks. If you want all IBM Cloud Orchestrator users to deploy heat stacks, then include member role to property **`trusts_delegated_roles`**.

   For example, replace

   ```
   trusts_delegated_roles=heat_stack_owner
   ```

   with

   ```
   trusts_delegated_roles=member
   ```

   If you want all users to deploy heat stacks, then comment out the property **`trusts_delegated_roles`**:

   ```
   trusts_delegated_roles=heat_stack_owner
   ```

   with

   ```
   #trusts_delegated_roles=heat_stack_owner
   ```

2. If Public Cloud Gateway regions are configured, then update the **`region_name_for_services`** property in `/etc/heat/heat.conf` with the value of **`OS_REGION_NAME`**.

   ```
   region_name_for_services=<OS_REGION_NAME_value>
   ```

3. Restart heat services.

   Alternatively, assign the role of `heat_stack_owner` to specific users who need to deploy heat stacks. In this case, you do not have to restart heat services.

Enabling OpenStack Dashboard for multi-domain support

Optionally, you can enable the OpenStack dashboard for multi-domain support from the OpenStack server where the horizon component is installed.

Procedure

1. Log in to OpenStack server where horizon component is installed.
2. Open the `local_settings.py` file in edit mode. The location of the file varies from one OpenStack distribution to another. For example, `/etc/openstack-dashboard/local_settings.py`.
3. Verify whether the following lines are available in the `local_settings.py` file:

   ```
   "identity": 3
   OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
   OPENSTACK_KEYSTONE_URL = "http://<ip address>:5000/v3"
   OPENSTACK_KEYSTONE_ADMIN_URL = "http://<openstack_keystone_server>:35357/v3"
   OPENSTACK_KEYSTONE_DEFAULT_ROLE = "member"
   ```

   If the value of `OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT` is set to False, then set the value to True.

   If the following line is not in the configuration file, it must be added:

   ```
   OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
   ```

Use the <openstack_keystone_server> value based on existing keystone endpoints of your OpenStack distribution. For example, IP address or host name of the server that hosts the keystone service.

Other changes vary from one OpenStack distribution to another. For more information, see OpenStack documentation at https://wiki.openstack.org/wiki/Horizon/DomainWorkFlow.

4. Save the changes and restart keystone service for the changes to take effect.

Creating OpenStack environment files

You can optionally create OpenStack environment files, v3rc and openrc, on the first controller or the master controller.

Procedure

Create OpenStack environment file in the /root directory of controller node. Set details of admin user in Default domain and admin project.

Example of an v3rc OpenStack environment file that has keystone v3 endpoints:

```
export OS_USERNAME=admin
export OS_PASSWORD=<admin password>
export OS_AUTH_URL=http://<Controller IP or Controller Host name>:5000/v3
export OS_REGION_NAME=<Region Name>
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
```

Example of an openrc OpenStack environment file that has keystone v2 endpoints:

```
export OS_USERNAME=admin
export OS_PASSWORD=<admin password>
export OS_AUTH_URL=http://<Controller IP or Controller Hostname>:5000/v2.0
export OS_REGION_NAME=<Region Name>
export OS_TENANT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
```

Note: If you are using HTTPS configuration, replace HTTP with HTTPS. If you must export your certificate for HTTPS configuration, add export OS_CACERT=<CA Certificate location> line to your OpenStack environment file.

Configuring an OpenStack Kilo distribution

Follow this procedure if you are using an OpenStack Kilo distribution of another vendor.

Before you begin

Ensure that your OpenStack installation meets the software requirements as described in "Checking OpenStack prerequisites" on page 17.

About this task

The commands and scripts that are mentioned in the following topics are sample commands and sample scripts that are provided with IBM Cloud Orchestrator. You must modify these scripts for your installation. For more information about how to prepare an OpenStack installation, see your OpenStack documentation. The samples scripts and additional files can be found in the /opt/ico_install/2.5.0-CSI-ICO-FP00010/utils/scripts/ directory after you extract the contents of the IBM Cloud Orchestrator image files, as described in "Downloading the required image files" on page 27.

The sample scripts use the OpenStack command line client and require functionality that is provided by version 1.0.3 and later. Ensure that you have an acceptable version installed. You can do this by running on the OpenStack Controller:

```
openstack --version
```

If you cannot install the version providing the required functions, adapt the scripts accordingly or run the steps in the scripts manually.

To prepare your OpenStack servers, complete the following procedure.

Procedure

1. "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 51.
2. "Stopping all the OpenStack services except Keystone" on page 52.
3. "Adding roles, users, and projects to Keystone" on page 52.
4. "Installing the IBM Cloud Orchestrator extensions for Horizon" on page 53.
5. "Configuring V3 API endpoints for Keystone" on page 55.
6. "Creating a new RC file for Keystone V3" on page 56.
7. "Installing the simple token extension" on page 56.
8. "Configuring the OpenStack services to use the Keystone V3 API" on page 57.
9. "Enabling the Cinder V1 API" on page 59.
10. "Starting all the OpenStack services and restarting Keystone" on page 59.
11. "[Optional] Configuring security groups" on page 59.

What to do next

After you prepare the OpenStack server, continue the IBM Cloud Orchestrator installation by following the procedure in "Setting the deployment parameters" on page 60.

Copying the IBM Cloud Orchestrator scripts to the OpenStack servers

IBM Cloud Orchestrator scripts must be run on each OpenStack Controller and compute node.

Before you begin

Before copying the scripts, you must extract the contents of the IBM Cloud Orchestrator image file, as described in "Downloading the required image files" on page 27.

In this procedure, the installation directory on the IBM Cloud Orchestrator Server is `/opt/ico_install/2.5.0-CSI-ICO-FP00010`, and the scripts directory on the OpenStack servers is `/opt/ico_scripts`. This might be adapted to your OpenStack distribution, for example, if your OpenStack distribution is using docker containers or other methods of delivering the OpenStack distribution than installing the code on top of the operating system.

Note: The sample scripts are provided as is and they must be adapted depending on your OpenStack distribution and topology. It is important to change the variables at the begin of the files to adjust, for example, install paths, user names, groups, IP addresses, and region names.

Procedure

1. Identify the OpenStack server where the Keystone service is running.
2. Identify the OpenStack servers where the Horizon service is running.
3. Identify all the OpenStack servers where other OpenStack services are running.
4. Copy the scripts by completing the following steps for each OpenStack server that you identified in the previous steps:

   a. Log on to the server as a `root` user.

   b. Create a directory to store the scripts:

   ```
   mkdir /opt/ico_scripts
   ```

   c. Copy all the files from the `/opt/ico_install/2.5.0-CSI-ICO-FP00010/utils/scripts` directory on the IBM Cloud Orchestrator Server to the `/opt/ico_scripts` directory on the OpenStack servers.

d. Copy the `sco_horizon.zip` file from the `/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo/packages/sco_horizon` directory on the IBM Cloud Orchestrator Server to the `/opt/ico_scripts` directory on the OpenStack servers.

Stopping all the OpenStack services except Keystone

To ensure that there are no ongoing activities while your OpenStack installation is configured, you must stop the OpenStack services. Only Keystone and all middleware services must still be running as Keystone is accessed during the reconfiguration process.

You must stop the services on all the OpenStack servers only. Keystone service must still be running.

See the documentation of your OpenStack distribution for information about what the services are and on how to stop them. Many Linux distributions use the `systemctl` command or the service tools to control the services. For example, to stop a service, run the following command:

```
systemctl stop <service>
```

See the documentation of your OpenStack distribution for information about the names of the OpenStack services. Usually, they are all the services that begin with `neutron-`, `openstack-`, or `httpd`, which is the Horizon service.

If your distribution uses the `systemctl` command and OpenStack services with the mentioned names, you can use a sample script named `control_services.sh` to easily stop the services. Review the script, adapt it to your environment if needed, and run the following command:

```
sh ./control_services.sh -o stop
```

This command stops all the OpenStack services except the Keystone service, and all the middleware services. You must run the command on all the OpenStack servers.

If you are using an OpenStack distribution with different tools, adapt the sample script or stop all the OpenStack services except the Keystone service, and all the middleware services manually on all the OpenStack servers.

Adding roles, users, and projects to Keystone

Create the roles, users, and projects that are required by IBM Cloud Orchestrator, and add them to OpenStack Keystone.

Before you begin

Ensure that you copied the `configure_ico_roles.sh` script from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 51.

Review the changes that are made by the script before running it.

About this task

You must run the `configure_ico_roles.sh` script on an OpenStack server where the Keystone and OpenStack command line is installed and configured. Usually, this server is the master controller server where the Keystone service is running. You must run the script only once.

The script creates the following roles, users, and projects in IBM Cloud Orchestrator:

Roles	`netadmin`
	`sysadmin`
	`domain_admin`
	`catalogeditor`
	`member`

Users	`demo` `domadmin`
Projects	demo

The access privileges are granted as follows:

- The demo user is granted the demo role on the demo project.
- The domadmin user is granted the domain_admin role on the admin project.
- The user admin is granted the member role on the admin project.

In this procedure, the example scripts directory on the OpenStack Controller is /opt/ico_scripts. Replace this value with the appropriate value for your installation.

Complete the following steps:

Procedure

1. Log on to the OpenStack Controller as a root user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Set the environment to the correct OpenStack values. Most OpenStack distributions provide an RC file containing these values, for example /root/openrc or /root/keystonerc. Run the following command:

   ```
   source /root/openrc
   ```

   If an RC file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v2.0
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   ```

   Values might differ for your OpenStack distribution.
4. Run the keystone role-list command to check if a Member role exists. If it exists, delete it by running the following command:

   ```
   keystone role-delete Member
   ```

5. Run the script:

   ```
   ./configure_ico_roles.sh
   ```

Installing the IBM Cloud Orchestrator extensions for Horizon

To support the IBM Cloud Orchestrator functionality, you must extend the base OpenStack Horizon software to provide additional options in the OpenStack Dashboard.

Before you begin

Ensure that you copied the BYOOS_configure_ico_horizon_extensions.sh script from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 51.

Depending on your installation and OpenStack distribution, you may need to adapt the script.

About this task

The IBM Cloud Orchestrator extensions for Horizon provide the following capability in the OpenStack Dashboard:

- Support for availability zones on domains and projects
- Support for quotas on domains
- Creation of a default project when a domain is created

- Ability to add domain administrators to the default project of a domain

In this procedure, the example scripts directory on the OpenStack Controller is `/opt/ico_scripts`. Replace this value with the appropriate value for your installation.

Complete the following steps on each OpenStack server where the Horizon service is installed.

Procedure

1. Log on to the OpenStack Controller as a `root` user.
2. Make sure that the `unzip` and `msgfmt` utilities are installed.
3. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

4. Set the environment to the correct OpenStack values. Most OpenStack distributions provide an RC file containing these values, for example `/root/openrc` or `/root/keystonerc`. Run the following command:

   ```
   source /root/openrc
   ```

   If an RC file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v2.0
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   ```

5. Review the `BYOOS_configure_ico_horizon_extensions.sh` script and adapt it to your installation if needed. Then run the following command:

   ```
   ./BYOOS_configure_ico_horizon_extensions.sh
   ```

6. Make the following changes to the `<path_to_Python_site_package>/openstack_auth/user.py` file paying attention to the Python indentation:

   a) `import urlparse`

   b) Find the `available_services_regions(self)` method and add the following lines (in **bold**):

   ```
   @property
       def available_services_regions(self):
           """Returns list of unique region name values in service
           catalog."""
           regions = []
           if self.service_catalog:
               for service in self.service_catalog:
                   service_type = service.get('type')
                   if service_type is None or service_type == 'identity':
                       continue
                   for endpoint in service.get('endpoints', []):
                       # IBM ONLY ICO BEGIN
                       # ICO should also exclude PCG regions
                       # PCG regions have the region name as part of the URL
                       path = urlparse.urlparse(endpoint['url']).path
                       if endpoint['region'] in path:
                           continue
                       # IBM ONLY ICO END
                       region = utils.get_endpoint_region(endpoint)
                       if region not in regions:
                           regions.append(region)
               return regions
   ```

   c) Add the following new method:

   ```
   # IBM ONLY ICO BEGIN
       @property
       def all_services_regions(self):
           """
           Returns list of unique region name values found in service catalog
   ```

```
        Method used to load availablity zones of all regions including PCG region
        """
        regions = []
        if self.service_catalog:
            for service in self.service_catalog:
                service_type = service.get('type')
                if service_type is None or service_type == 'identity':
                    continue
                for endpoint in service.get('endpoints', []):
                    region = utils.get_endpoint_region(endpoint)
                    if region not in regions:
                        regions.append(region)
        return regions

    # IBM ONLY ICO END
```

7. In the `<path_to_Python_site_package>/openstack_auth/utils.py` file, add the following line (in **bold**), if the method exists, paying attention to the Python indentation:

```
default_services_region(service_catalog, request=None):
available_regions = [get_endpoint_region(endpoint) for service
                        in service_catalog for endpoint
                        in service.get('endpoints', [])
                        if (service.get('type') is not None
                        and service.get('type') != 'identity'
                        and endpoint.get('region') not in
urlparse.urlparse(endpoint['url']).path)]
```

8. Restart the Horizon service.

Results

When you log in to the OpenStack Dashboard, you can work with the IBM Cloud Orchestrator extensions.

Note: For VMware regions, the **Reassign Instances** functionality is not available in a generic OpenStack environment, even if the related button is displayed in the OpenStack Dashboard.

Configuring V3 API endpoints for Keystone

You must create Keystone V3 endpoints and delete V2 endpoints for IBM Cloud Orchestrator. Verify in the documentation of your OpenStack distribution if your installation already uses Keystone V3 API or still the V2 API. Also verify if the V2 endpoints still exists. The sample script as described in this procedure assumes that Keystone V2 is still used. This means that the V2 endpoints still exist and no V3 endpoints are defined. If your OpenStack installation is different, you must adapt the sample scripts accordingly.

Before you begin

Ensure that the IBM Cloud Orchestrator extensions for Horizon are installed on each OpenStack server, as described in "Installing the IBM Cloud Orchestrator extensions for Horizon" on page 53.

Ensure that you copied the `configure_endpoints.sh` script from the IBM Cloud Orchestrator server, as described in "Copying the IBM Cloud Orchestrator scripts to the OpenStack servers" on page 51.

Verify in the documentation of your OpenStack distribution if your installation already uses Keystone V3 API or still the V2 API. Also verify if the V2 endpoints still exists. The sample script in this procedure assumes that Keystone V2 is still used. This means that the V2 endpoints still exist and no V3 endpoint is defined. If your OpenStack installation is different, you must adapt the sample script accordingly.

About this task

You must run the `configure_endpoints.sh` script on an OpenStack server where the Keystone and OpenStack command line is installed and configured. Usually, this server is the master controller server where the Keystone service is running. You must run the script only once. OpenStack command line client must be version 1.0.3 or later.

The script creates endpoints for the Keystone V3 API and deletes the endpoints for Keystone V2 API. It does this operations for all the regions.

In this procedure, the example scripts directory on the OpenStack server is `/opt/ico_scripts`. Replace this value with the appropriate value for your installation.

Procedure

1. Log on to the OpenStack server as a `root` user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Set the environment to the correct OpenStack values. Most OpenStack distributions provide an RC file containing these values, for example `/root/openrc` or `/root/keystonerc`. Run the following command:

   ```
   source /root/openrc
   ```

   If an RC file is not provided by your OpenStack distribution, set the values manually, for example:

   ```
   export OS_USERNAME=admin
   export OS_PASSWORD=openstack1
   export OS_TENANT_NAME=admin
   export OS_AUTH_URL=http://192.0.2.68:5000/v2.0
   export OS_REGION_NAME=kvm-allinone2
   export OS_VOLUME_API_VERSION=2
   ```

   Values might differ for your OpenStack distribution.
4. Run the script:

   ```
   ./configure_endpoints.sh
   ```

Results

Keystone V3 endpoints are created and V2 endpoints are deleted.

Creating a new RC file for Keystone V3

This topic describes how to create a new RC file for Keystone V3.

Most OpenStack distributions provide an RC file containing OpenStack values, for example `/root/openrc` or `/root/keystonerc`. You can use this file to load the correct OpenStack values for the command line clients to interact with the services. After the change to Keystone V3, the existing file does not work any more. Add and change the values in the existing RC file, for example:

```
export OS_USERNAME=admin
export OS_PASSWORD=openstack1
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://192.0.2.68:5000/v3
export OS_REGION_NAME=kvm-allinone2
export OS_VOLUME_API_VERSION=2
export OS_IDENTITY_API_VERSION=3
export OS_USER_DOMAIN_NAME=${OS_USER_DOMAIN_NAME:-"Default"}
export OS_PROJECT_DOMAIN_NAME=${OS_PROJECT_DOMAIN_NAME:-"Default"}
```

Change the `OS_AUTH_URL` value and add the last 3 lines.

Also, the Keystone command line client only supports keystone V2 API. After the change, use the `openstack` client.

After you edited your RC file, run the following command:

```
source <name_of_your_RC_file>
```

Installing the simple token extension

For IBM Cloud Orchestrator to access your Keystone, the simple token extension must be installed.

About this task

Complete the following steps on the server where the Keystone service is installed. Refer to your OpenStack documentation.

Procedure

1. Log on to the OpenStack Controller as a `root` user.
2. Change directory to the directory where you store the OpenStack Controller scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Copy the `simpletoken.py` file to the middleware directory of your Keystone server.
4. In the `/etc/keystone/keystone.conf` file, define:

   ```
   [authentication]
   simple_token_header = SimpleToken
   simple_token_secret = Y6A8MiJGYDr1bzZPP/kt/A==
   ```

   The simple token here must also be given the IBM Cloud Orchestrator installer as an input parameter. The secret must be a base64 encoded value. To generate a secret, run the following command:

   ```
   dd if=/dev/urandom bs=16 count=1 2>/dev/null | base64
   ```

5. In `/etc/keystone/keystone-paste.ini` define the filter for simple token:

   ```
   [filter:simpletoken]
   paste.filter_factory=keystone.middleware.simpletoken:SimpleTokenAuthentication.factory
   ```

6. In the `/etc/keystone/keystone-paste.ini` file, add the filter to the pipeline that you want to run. The filter must come after the `json_body` and `xml_body` filters, but before the actual application in the pipelines `[pipeline:public_api]`, `[pipeline:admin_api]`, and `[pipeline:api_v3]`.
7. Restart Keystone by running the following command:

   ```
   sytemctl restart openstack-keystone
   ```

Configuring the OpenStack services to use the Keystone V3 API

After the endpoints are configured to Keystone V3, you must configure the other OpenStack services to use the Keystone V3 API and to use the new Keystone V3 endpoints. You must configure the services on all the OpenStack servers in your installation.

Before you begin

Ensure that the Keystone endpoints are configured for Keystone V3, as described in "Configuring V3 API endpoints for Keystone" on page 40.

About this task

The OpenStack services must be configured to use the new V3 Keystone endpoints. See your OpenStack documentation for information about all the installed OpenStack services and where their configuration files are stored.

When you run the script in the following procedure, the following changes are made. Depending on your OpenStack distribution, you might need to adapt the script or to do the changes manually:

- The configuration files of the OpenStack services contains a section like, for example:

  ```
  [keystone_authtoken]
  auth_uri = http://192.0.2.67:5000/v2.0
  identity_uri = http://192.0.2.67:35357/
  auth_version = v2.0
  admin_tenant_name = service
  admin_user = nova
  admin_password = W0lCTTp2MV1iY3JhZmducHgtcGJ6Y2hncg==
  signing_dir = /var/cache/nova/api
  hash_algorithms = md5
  insecure = false
  ```

  The following parameters must be changed to use the new V3 endpoints:

  ```
  auth_uri = http://192.0.2.67:5000/v3
  ```

```
auth_version = v3
```

- In the `keystone.conf` file, the following parameter must be set:

  ```
  [auth]
  external = keystone.auth.plugins.external.Domain
  ```

- In the Cinder `api-paste.ini` file, the following section must be added:
-
  ```
  [keystone_authtoken]
  auth_uri = http://192.0.2.67:5000/v3
  identity_uri = http://192.0.2.67:35357/
  auth_version = v3
  admin_tenant_name = service
  admin_user = cinder
  admin_password = W0lCTTp2MV1iY3JhZmducHgtcGJ6Y2hncg==
  signing_dir = /var/cache/nova/api
  hash_algorithms = md5
  ```

- The following changes (in bold) must be done to the Horizon local settings:

  ```
  "identity": 3
  ```

  ```
  OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
  ```

  ```
  OPENSTACK_KEYSTONE_URL = "http://192.0.2.67:5000/v3"
  OPENSTACK_KEYSTONE_ADMIN_URL = "http://192.0.2.67:35357/v3"
  ```

  ```
  OPENSTACK_KEYSTONE_DEFAULT_ROLE = "member"
  ```

  If the following line is not in the configuration file, it must be added:

  ```
  OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
  ```

Additional changes might be needed.

After the changes, the Keystone `policy.json` file must be replaced with the version delivered by IBM Cloud Orchestrator. To do this manually, copy the `keystone_policy.json` file into the Keystone configuration directory, for example `/etc/keystone`, as `policy.json`. See your OpenStack distribution documentation for the correct path and file name to replace. Then, change the owner of this file to the original values.

Procedure

1. Log on to the OpenStack server as a `root` user.
2. Change directory to the directory where you store the IBM Cloud Orchestrator scripts:

   ```
   cd /opt/ico_scripts
   ```

3. Set the environment to the correct OpenStack values. Use the new RC file that you created by following the procedure in "Creating a new RC file for Keystone V3" on page 56. Run the following command:

   ```
   source <name_of_your_RC_file>
   ```

   Alternatively, you can provide the `OS_AUTH_URL` value as command parameter in the following step.
4. Run the script:

   ```
   ./BYOOS_configure_files.sh [OS_AUTH_URL]
   ```

5. Copy the Keystone `policy.json` file to the Keystone configuration directory:

   ```
   cp keystone_policy.json <path_to_keystone_config>/policy.json
   ```

Results

All the OpenStack services are configured to use the new Keystone V3 endpoints.

Enabling the Cinder V1 API

Configure OpenStack Cinder to use the V1 API.

About this task

Complete the following steps on each OpenStack Controller.

Procedure

1. Log on to the OpenStack Controller as a root user.
2. Open the Cinder configuration file, for example `/etc/cinder/cinder.conf`, and ensure that the `enable_v1_api` parameter is set to `true`. Refer to your OpenStack documentation to find the correct file name and where it is stored.

Starting all the OpenStack services and restarting Keystone

After the configuration is done, you must start all the OpenStack services. You must also restart the Keystone service.

See the documentation of your OpenStack distribution for information about what the services are and on how to start them. Many Linux distributions use the `systemctl` command or the service tools to control the services. For example, to start a service, run the following command:

```
systemctl start <service>
```

See the documentation of your OpenStack distribution for information about the names of the OpenStack services. Usually, they are all the services that begin with `neutron-`, `openstack-`, or `httpd`, which is the Horizon service.

If your distribution uses the `systemctl` command and OpenStack services with the mentioned names, you can use a sample script named `control_services.sh` to easily start the services. Review the script, adapt it to your environment if needed, and run the following commands:

```
./control_services.sh -o restart -k
./control_services.sh -o
```

These commands restart the Keystone service and start all the services. You must run the command on all the OpenStack servers.

[Optional] Configuring security groups

You can configure security groups to allow **`ssh`** and **`ping`** commands to access virtual machines that are deployed in your OpenStack environment. You can also configure security groups to enable RDP ports for Windows virtual machine instances.

About this task

This task is optional. Consider whether you want to allow such access. For more information, see the documentation for your chosen OpenStack product.

Note: If you a working in a multi-domain environment, you must run this procedure for each domain in which you want to access the deployed virtual machines. Specify the domain in the RC file used in the step "2" on page 59 of the procedure.

Procedure

1. Log on to the OpenStack Controller as a `root` user.
2. Set the environment to the correct OpenStack values. Use the new RC file that you created by following the procedure in "Creating a new RC file for Keystone V3" on page 56. Run the following command:

```
source <name_of_your_RC_file>
```

3. If you want to configure security groups to allow access by **ssh** and **ping** commands, run the following commands:

```
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```

4. If you want to enable RDP ports for Windows virtual machine instances, run the following commands:

```
nova secgroup-add-rule default tcp 3389 3389 0.0.0.0/0
nova secgroup-add-rule default udp 3389 3389 0.0.0.0/0
```

Setting the deployment parameters

Before you install, you must provide valid values for the parameters that are used by the IBM Cloud Orchestrator installer to deploy IBM Cloud Orchestrator.

In this procedure, the example installation directory is /opt/ico_install/2.5.0-CSI-ICO-FP00010. Replace this value with the appropriate value for your installation.

Use this procedure to set the deployment parameters when installing the following topologies:

- Single-Server topology: skip steps "4" on page 62 and "5" on page 63 in the procedure.
- Single-Server with external database topology: skip step "5" on page 63 in the procedure.
- Dual-Server high-availability with external database topology: run all the steps in the procedure.
- IBM Cloud Orchestrator Keystone topology

Procedure

1. Log on to the IBM Cloud Orchestrator Server as a root user.
2. Change directory to the installer directory:

```
cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
```

3. Edit the ico_install.rsp response file to specify the value of each mandatory parameter, as described in the following table. If any parameter does not have a default value, or if the default value is not suitable for your environment, update the response file to specify an appropriate value. Use the following format for each parameter entry in the response file:

```
parameter_name parameter_value
```

Table 7. Mandatory deployment parameters

Name	Description
LICENSE_ACCEPTED	The flag that indicates whether you accept the license terms. To accept the license automatically without being prompted by the installer, set this parameter to True. If you set this parameter to True, you confirm that you accept the license terms.
OPENSTACK_ADMIN_PASSWORD	The current password for the OpenStack Controller admin user. This password is also used for the IBM Cloud Orchestrator admin user (admin).

Table 7. Mandatory deployment parameters (continued)

Name	Description
`OPENSTACK_HOST_NAME`	The fully qualified domain name of the OpenStack Controller where the master OpenStack Keystone service is installed. **Note:** Specify the IBM Cloud Orchestrator node Host name / IP if `KEYSTONE_ICO_INSTALLATION` is set to True.
`ORCHESTRATOR_PASSWORD`	The password for the Business Process Manager admin users (`bpm_admin` and `tw_admin`) and for the IHS keystore. This password is also used for the Business Process Manager database user (`bpmuser`) and the IBM DB2 users (`db2inst1`, `db2das1`, `db2fenc1`), unless you specify different passwords for these users elsewhere in the response file. The password can contain only the following characters: `` a-z A-Z 0-9 - . _ ` ~ @ `` The password cannot contain spaces. **Note:** For security reasons, the password is removed from the response file when the IBM Cloud Orchestrator deployment is completed.
`SIMPLE_TOKEN_SECRET`	The string that is used for simple token authentication for OpenStack. To populate this value in the response file, run the `update-token.sh` script, as described in "Adding the OpenStack simple token to the response file" on page 64. For `IBM Cloud Orchestrator Keystone` topology, you do not have to set any value for this parameter. It is mandatory only if you configure IBM Cloud Manager with OpenStack. For `IBM Cloud Orchestrator Keystone` topology, see `KEYSTONE_ICO_INSTALLATION` parameter.

Table 7. Mandatory deployment parameters (continued)

Name	**Description**
PROTOCOL	Value can be HTTP or HTTPS. It depends on whether you are configuring IBM Cloud Orchestrator with OpenStack having keystone endpoint on HTTP or HTTPS. If the protocol is HTTPS and IBM Cloud Orchestrator Server is integrated with IBM Cloud Manager with OpenStack configured for secure communication as mentioned in "Configuring IBM Cloud Manager with OpenStack for HTTPS" on page 26, you must add the Cloud Manager certificate bundle in the <TMP_DIR> directory. See "Adding the certificate bundle of IBM Cloud Manager with OpenStack to IBM Cloud Orchestrator Server " on page 66 for the procedure.
ENDPOINT_BY_HOSTNAME_OR_IP	Value can be ip or hostname. This identifies if the Keystone V3 endpoint available on the OpenStack Controller is using an IP address or a hostname.
KEYSTONE_ICO_INSTALLATION	Set the value to True to select IBM Cloud Orchestrator Keystone topology wherein keystone is installed on IBM Cloud Orchestrator node. You do not have to follow the procedure to configure IBM Cloud Manager with OpenStack. For the actual steps to configure and use this topology, see "Installing IBM Cloud Orchestrator Keystone topology" on page 11.
BYOOS	This flag indicates whether BYOOS is configured with IBM Cloud Orchestrator or not. Set the flag if the BYOOS does not support cinder v1 API.

4. [For Single-Server with external database topology or Dual-Server high-availability with external database topology:] Edit the following additional entries in the ico_install.rsp response file. Specify the value of each parameter, as described in the following table.

Table 8. Additional mandatory deployment parameters for external database topology

Name	**Description**
BPM_DB_ADDR	The host name of the server that hosts the Business Process Manager Process Server database. If you set this parameter, an external database is enabled.
BPM_DB_PASSWORD	The password for the Process Server database user.
BPM_DB_USERNAME	The user name of the Process Server database user.

Table 8. Additional mandatory deployment parameters for external database topology (continued)

Name	Description
DBPORT	The port that is used to access IBM Cloud Orchestrator databases during installation. If you are using an external pre-installed DB2, use the corresponding port that allows connection from IBM Cloud Orchestrator node. During installation, use the value of the HTTP port (50000 by default). Note that after the installation the port to connect to the database might be set to a different value depending on the communication protocol you choose (HTTP or HTTPS).
EXT_DB_PASSWORD	The root password to access the external database server. If you do not specify this parameter in the response file, you are prompted for the root password during the installation.

If you want to use an existing external IBM DB2 database server, you must create the Business Process Manager databases before you install IBM Cloud Orchestrator. For information about how to create the databases, see "[Optional] Creating the Business Process Manager databases on an external IBM DB2 server" on page 30.

5. **[For both multisite and single site Dual-Server high-availability with external database topology:]** Edit the following additional entries in the ico_install.rsp response file. Specify the value of each parameter, as described in the following table.

Table 9. Additional mandatory deployment parameters for high-availability topology

Name	Description
HA_SECONDARY_HOST	The fully qualified domain name of the second server to install IBM Cloud Orchestrator.
HA_SECONDARY_USER	The user to be used for the IBM Cloud Orchestrator installation on the second server.
HA_SECONDARY_KEY	The path to the SSH key to use when connecting using SSH to HA_SECONDARY_HOST.
HA_SECONDARY_PASSWORD	The password to use when connecting using SSH to HA_SECONDARY_HOST. **Note:** You must configure either a password or an SSH key.
HA_VIRTUAL_IP_ADDRESS	The virtual IP address to be used in the high availability configuration. This IP address must be used in the same subnet as the IP addresses of the two servers where IBM Cloud Orchestrator is installed highly available.
HA_VIRTUAL_NETMASK	Netmask to be used for the virtual IP address.
HA_VIRTUAL_TIEBREAKER	The IP address of the default gateway of the subnet that is used for the virtual machines of the IBM Cloud Orchestrator management stack. The System Automation for Multiplatforms cluster uses network connectivity to this default network gateway to determine whether a node is still up and running. If a cluster split occurs, the tiebreaker determines which part of the cluster gets quorum and can manage active resources. For more information about quorums and tiebreakers, see Operational quorum.

Table 9. Additional mandatory deployment parameters for high-availability topology (continued)

Name	Description
HA_VIRTUAL_IP_HOSTNAME	The fully qualified domain name of the virtual IP address.
HA_VIRTUAL_NET_INTERFACE	The network interface that is used for communication between the IBM Cloud Orchestrator management components, for example, ens192. This value must be consistent for both servers.

For details, for a multi site IBM Cloud Orchestrator high availability installation, refer to "Installing high availability across multiple sites" on page 72.

6. [Optional:] If you want to change the directory where IBM Cloud Orchestrator is installed or the temporary directory that is used during the installation, change the parameters that are described in the following table.

Table 10. Optional deployment parameters

Name	Description
INSTALL_ROOT	The base directory where the IBM Cloud Orchestrator components are installed. The default value is **/opt/ibm/ico** and it is used in this product documentation to refer to the installation path of the IBM Cloud Orchestrator components. The installation directory must not be a symbolic link.
TMP_DIR	The temporary directory that is used during the installation of the IBM Cloud Orchestrator components. Ensure that the <TMP_DIR> is available on both the primary and secondary nodes with correct privileges. The default location for <TMP_DIR> is /tmp/ico. The temporary directory must not be mounted with the noexec, nodev, and nosuid options.

Adding the OpenStack simple token to the response file

Run the update-token.sh script to insert the OpenStack simple token in the response file.

Before you begin

- Before you run the script, ensure that the value of the **OPENSTACK_HOST_NAME** parameter is specified in the response file, as described in "Setting the deployment parameters" on page 60. In this procedure, the example installation directory is /opt/ico_install/2.5.0-CSI-ICO-FP00010. Replace this value with the appropriate value for your installation.
- If some components of the IBM Cloud Orchestrator are deployed in public data center or cloud (on public network), enable HTTPS and VPN connection between IBM Cloud Orchestrator and OpenStack. For more information, see https://www.ibm.com/support/knowledgecenter/en/SST55W_4.3.0/liaca/liaca_hybrid_hybrid_cloud.html.

About this task

If you are using the IBM Cloud Orchestrator Keystone topology, you do not have to add the OpenStack simple token to the response file.

Procedure

1. Log on to the IBM Cloud Orchestrator Server.
2. Change to the directory where the installer scripts are located:

```
cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
```

3. Run the script:

```
./update-token.sh response_file user_name
```

where

response_file
The name of the response file that is used for the installation (for example, `ico_install.rsp`). If the response file is not in the current directory, specify the full path name.

user_name
The name of an operating system user who can log on to the OpenStack Controller that is specified in the **OPENSTACK_HOST_NAME** parameter in the response file. The specified user must have permission to run **sudo** commands on the OpenStack Controller.

4. When prompted, enter the password for the specified user.

Results

The **SIMPLE_TOKEN_SECRET** parameter in the response file is set to the simple token authentication string.

Tip: If the `update-token.sh` script can connect to the remote server but cannot find the token, identify the token manually by running the following command on the OpenStack Controller:

```
grep simple_token_secret /etc/keystone/keystone.conf
```

Then manually edit the response file on the IBM Cloud Orchestrator Server to replace the **SIMPLE_TOKEN_SECRET** parameter value with the output of the **grep** command.

Checking the installation prerequisites

Run the `prereq-checker.sh` script to check various installation prerequisites.

Before you begin

Before you run the `prereq-checker.sh` script, specify the values of the mandatory parameters in the response file, as described in "Setting the deployment parameters" on page 60 and "Adding the OpenStack simple token to the response file" on page 64.

About this task

The `prereq-checker.sh` script completes the following system validation checks:

- Checks that all system hardware resources such as CPU, memory, and free disk space are correct, as described in "Checking the hardware prerequisites" on page 21.
- Checks that the operating system flavor and version are correct, as described in "Checking the software prerequisites" on page 22.
- Tests OpenStack Keystone connectivity, as described in "Checking the software prerequisites" on page 22.
- Tests the connectivity of other OpenStack services, such as Cinder (V1 and V2), Glance, Nova, and Neutron..
- Ensures that the required packages are installed in the correct location, as described in "Downloading the required image files" on page 27.
- Validates the installation parameters, as described in "Setting the deployment parameters" on page 60.

Note: The `prereq-checker.sh` script also installs the required `"Expect"` package on your system.

In this procedure, the example installation directory is `/opt/ico_install/2.5.0-CSI-ICO-FP00010`. Replace this value with the appropriate value for your installation.

Procedure

1. Log on to the IBM Cloud Orchestrator Server.
2. Change to the directory where the installer scripts are located:

```
cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
```

3. Run the script:

   ```
   ./prereq-checker.sh response_file
   ```

   where *response_file* is the name of the response file that is used for the installation (for example, ico_install.rsp).

Results

The script output indicates whether you can proceed with the installation, or whether you need to take action first.

Adding the certificate bundle of IBM Cloud Manager with OpenStack to IBM Cloud Orchestrator Server

For secure communication in IBM Cloud Manager with OpenStack, a Privacy Enhanced Mail (PEM) bundle is configured for certificate validation. You must add the bundle to the <TMP_DIR> directory of IBM Cloud Orchestrator Server so that it is available during the installation process.

Procedure

1. Log in to the IBM Cloud Orchestrator Server.
2. Create <TMP_DIR> directory.
3. Run the following command to copy the certificate bundle:

   ```
   scp root@<ICM_controller>: /etc/pki/tls/icm/certs/ca_bundle.pem /<TMP_DIR>/openstack.crt
   ```

   where <ICM_controller> is the IP address or the fully qualified host name of IBM Cloud Manager with OpenStack master controller wherein the Keystone service is run. For multi-region setup, it is the controller of the first region. For high availability configuration, it is the primary controller of the first region installed.
4. Optionally, run the following command to check the certificate bundle content:

   ```
   openssl x509 -noout -text -in <TMP_DIR>/openstack.crt
   ```

Deploying the IBM Cloud Orchestrator Servers

Use a shell script to deploy one or two IBM Cloud Orchestrator Servers depending on the topology you chose, based on the deployment parameter values that you provided in the response file.

Before you begin

Edit the response file to provide appropriate values for the deployment parameters, as described in "Setting the deployment parameters" on page 60 and "Adding the OpenStack simple token to the response file" on page 64. Check that your servers meet the hardware and software prerequisites, as described in "Checking the installation prerequisites" on page 65.

In this procedure, the example temporary directory where you unpacked the installation image files is /opt/ico_install/2.5.0-CSI-ICO-FP00010, and the example response file is ico_install.rsp. Replace these values with the appropriate values for your installation.

Note: The IBM Cloud Orchestrator user interfaces and documentation are translated into several languages, but the installer user interface is provided only in English.

Procedure

1. Log on to the IBM Cloud Orchestrator Server in a Single-Server topology installation, or log on to the primary IBM Cloud Orchestrator Server in a Dual-Server high-availability topology installation.

Important: You must have root authority to run the IBM Cloud Orchestrator installation.

2. Change directory to the installer directory:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   ```

3. Run the following command to install the IBM Cloud Orchestrator components on the server:

   - If you logged on as `root` user:

     ```
     ./ico_install.sh ico_install.rsp
     ```

   - If you did not log on as `root` user:

     ```
     sudo ./ico_install.sh ico_install.rsp
     ```

   You are prompted to accept the license agreement. Read the license agreement, and accept or decline the license terms. If you do not accept the license agreement, the installation exits.

   Note: To accept the license automatically without being prompted by the installer, set the **`LICENSE_ACCEPTED`** parameter to `True` in the response file.

   Subject to disk speed, the installation should complete within 2 hours.

4. If your install fails with an error, review the installation log file (`/var/log/ico_install/ico_install_YYYYMMDDhhmm.log`) to check why the installation was unsuccessfully. If necessary, take any appropriate action as indicated in the log file.

5. After installing IBM Cloud Orchestrator, for security reasons, complete the following steps to prevent unrestricted access to the user and group lists in Business Space when using REST APIs:

   a. Log in as bpm_admin to the WebSphere® Application Server Integrated Solutions console at `https://$ico_server:9043/ibm/console/logon.jsp`.

   b. Navigate to **Resources** > **Resource Environment** > **Resource environment providers** > **Mashups_ConfigService** > **Custom properties**.

   c. Create a new `String` type property with the following values:

      ```
      Scope = cells:PCCell1:clusters:SingleCluster
      Name = com.ibm.mashups.usersearch.blocked
      Value = true
      Type = java.lang.String
      ```

      Setting the value to true restricts the global user or group search via the Business Space REST APIs.

   d. Apply and save the custom property in the master configuration and log out from the WebSphere Application Server Integrated Solutions console.

   e. Restart the Business Process Manager server by running the following command on the IBM Cloud Orchestrator Server:

      ```
      systemctl restart bpm
      ```

Results

One or two IBM Cloud Orchestrator Servers depending on the topology you chose are installed.

- If you have multiple regions in OpenStack topology, then do the following steps to copy the certificate of other region in <TMP_DIR> directory and import them in Business Process Manager. The certificate must be named as `openstack.crt`.

  1. Run the following command to import the certificate of other region in Business Process Manager:

     ```
     source /etc/profile.d/jdk.sh;keytool -import -v -noprompt -trustcacerts -alias <alias>
     -file /tmp/openstack.crt -storepass changeit -keystore <ICO_INSTALL_DIRECTORY>
     /BPM/v8.5/java_1.7_64/jre/lib/security/cacerts
     ```

  2. Run the following command to import the certificate of other region in Self Service user interface:

```
source /etc/profile.d/jdk.sh;keytool -noprompt -import -file
/tmp/openstack.crt -alias <alias> -keystore <ICO_INSTALL_DIRECTORY>
/wlp/usr/servers/scui/resources/security/keystore -storepass password
```

3. After you import certificates, restart IBM Cloud Orchestrator.

What to do next

Complete the installation verification steps, as described in “Verifying the installation” on page 68.

If you want to install high availability across multiple sites, see “Installing high availability across multiple sites” on page 72.

If IBM Cloud Orchestrator installation is for HTTPS configuration, then import Self-service user interface certificate in an OpenStack server. For the actual procedure, see “Importing SCUI certificate in an OpenStack Server” on page 114.

Verifying the installation

When you have completed the installation of your IBM Cloud Orchestrator Servers, you can verify the installation by completing the following steps.

Procedure

1. Verify that the status of the IBM Cloud Orchestrator components is correct.
   - For a non high-availability installation:
     a. Run the following command on the IBM Cloud Orchestrator Server:

     ```
     /opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --status
     ```

     b. Verify that the status of each IBM Cloud Orchestrator component is `online`, as shown in the following example output:

     ```
     ===>>> Collecting Status for IBM Cloud Orchestrator
     ===>>> Please wait ======>>>>>>

             Component                     Hostname                      Status
             ------------------------------------------------------------------
             IHS                           192.0.2.84                    online
             bpm-dmgr                      192.0.2.84                    online
             bpm-node                      192.0.2.84                    online
             bpm-server                    192.0.2.84                    online
             db2                           192.0.2.84                    online
             pcg                           192.0.2.84                    online
             swi                           192.0.2.84                    online

     ===>>> Status IBM Cloud Orchestrator complete
     ```

   - For a high-availability installation:
     a. Run the following command on the IBM Cloud Orchestrator Server:

     ```
     lssam
     ```

     this is a sample output of the command:

     ```
     Online IBM.ResourceGroup:central-services-rg Nominal=Online
             |- Online IBM.Application:bpm-node
                  |- Online IBM.Application:bpm-node:primaryiconode
                  '- Online IBM.Application:bpm-node:secondaryiconode
             |- Online IBM.Application:bpm
                  |- Online IBM.Application:bpm:primaryiconode
                  '- Online IBM.Application:bpm:secondaryiconode
             |- Online IBM.Application:ihs
                  |- Offline IBM.Application:ihs:primaryiconode
                  '- Online IBM.Application:ihs:secondaryiconode
     ```

```
        |- Online IBM.Application:scui
             |- Online IBM.Application:scui:primaryiconode
             '- Online IBM.Application:scui:secondaryiconode
        '- Online IBM.ServiceIP:cs-ip
             |- Offline IBM.ServiceIP:cs-ip:primaryiconode
             '- Online IBM.ServiceIP:cs-ip:secondaryiconode
Online IBM.ResourceGroup:pcg-rg Nominal=Online
        '- Online IBM.Application:pcg
             '- Online IBM.Application:pcg:primaryiconode
Online IBM.Equivalency:cs-network-equ
        |- Online IBM.NetworkInterface:ens192:primaryiconode
        '- Online IBM.NetworkInterface:ens192:secondaryiconode
```

Note: If the status of the services is not shown online, then reboot the node first and then restart the services.

2. Verify that you can access and log in to the following IBM Cloud Orchestrator user interfaces:

   - For a non high-availability installation:

     – Self-service user interface

     ```
     https://ico_server_fqdn:443
     ```

     – Business Process Manager user interface

     ```
     https://ico_server_fqdn:443/ProcessCenter/login.jsp
     ```

     where *ico_server_fqdn* is the fully qualified domain name (for example, `host.example.com`) of the IBM Cloud Orchestrator Server.

   - For a high-availability installation:

     – Self-service user interface:

     ```
     https://virtualIP_fqdn:443
     ```

     – Business Process Manager user interface:

     ```
     https://virtualIP_fqdn:443/ProcessCenter/login.jsp
     ```

     where *virtualIP_fqdn* is the fully qualified domain name (for example, `host.example.com`) of the virtual IP address.

   To access each user interface, use the following credentials:

   - Domain: `Default`
   - User: `admin` (OpenStack user)
   - Password: The password that you specified with the `OPENSTACK_ADMIN_PASSWORD` parameter in the response file

   For more information about accessing the IBM Cloud Orchestrator user interfaces, see Chapter 5, “Accessing the IBM Cloud Orchestrator user interfaces,” on page 155.

   If you cannot access the Self-service user interface, ensure that your firewall is configured to open the ports that are required by IBM Cloud Orchestrator. For information about the required ports, see “Ports used by IBM Cloud Orchestrator” on page 80.

3. Verify that the Self-Service Catalog is populated. In the Self-service user interface, click **SELF-SERVICE CATALOG** and explore the categories and offerings.
4. Confirm that the IBM Cloud Orchestrator extensions for Horizon are installed:

   a) Log on to the OpenStack Dashboard:

   ```
   https://openstack_server_fqdn/
   ```

   where *openstack_server_fqdn* is the fully qualified domain name (for example, `host.example.com`) of the OpenStack Controller.

   b) Click **Identity** > **Domains**, and click **Edit a Domain**.

c) Confirm that the **Availability Zones** tab is visible.

For more information about installing the extensions, see "Installing the IBM Cloud Orchestrator extensions for Horizon" on page 39.

What to do next

Before you can use IBM Cloud Orchestrator to manage your cloud environment, you must configure your environment as described in Chapter 4, "Configuring," on page 117. At a minimum, you must assign zones to domains and projects, as described in "Assigning zones to domains and projects" on page 117.

You can then test the configuration by creating and registering an image, and then deploying the image to a region, as described in Chapter 9, "Managing virtual images," on page 239.

IBM provides in a bulletin information of security vulnerabilities that are found after product shipment with proposed remediation and fixes. It is suggested that after installation you review any bulletin that might apply at IBM Security Bulletins searching for IBM Cloud Orchestrator V2.5.

Installing IBM Cloud Orchestrator Enterprise Edition

For more control over your cloud environment, IBM Cloud Orchestrator Enterprise Edition bundles three extra products.

- Jazz for Service Management
- IBM Tivoli Monitoring
- IBM SmartCloud Cost Management

These components can be installed on physical or virtual machines, subject to the relevant hardware and software requirements. For more information about these components, visit the following links:

- Quick Start Guide for Jazz for Service Management.
- Quick Start Guide for IBM SmartCloud Cost Management
- Quick Start Guide for IBM Tivoli Monitoring and Tivoli Monitoring for Virtual Environments

Installation procedure

The first part of the IBM Cloud Orchestrator Enterprise Edition installation is the same as for the base version. Subsequently, you can install the additional products on separate machines. If you have an existing IBM Cloud Orchestrator V2.5 installation, follow the procedure described in "Upgrading from IBM Cloud Orchestrator V2.5" on page 106 before installing or upgrading the additional products.

1. Refer to the installation procedure in the Chapter 2, "Installing," on page 9 section to install IBM Cloud Orchestrator and its services.
2. Install Jazz for Service Management V1.1.2.1. For instructions, see the Jazz for Service Management Quick Start Guide.
3. Install IBM Tivoli Monitoring V6.3.0.2. For instructions, see "Installing IBM Tivoli Monitoring" on page 292.
4. [Optional] Install IBM Tivoli Monitoring for Virtual Environments V7.2.0.2. For instructions, see IBM Tivoli Monitoring for Virtual Environments Quick Start Guide.
5. Install IBM SmartCloud Cost Management V2.1.0.6. For instructions, see "Quick start guide for metering and billing" on page 70.

Quick start guide for metering and billing

Use this topic as a start guide when you configure SmartCloud Cost Management for metering and billing.

The following list provides information about configuration steps that are required for metering and billing:

Install Tivoli Common Reporting 3.1.2.1

For information about this task, see the installing Tivoli Common Reporting section in the Jazz for Service Management information center.

Install SmartCloud Cost Management

For information about this task, see *Installing SmartCloud Cost Management 2.1.0.6 ifix07 of SCCM User's Guide*.

Configuration that is required for metering

For more information about the configuration that is required for metering, see the Automated configuration topic.

Configure job processing

- For information about configuring processing paths, see the *Setting processing options topic of SCCM User's Guide*.
- For information about defining rates and rate templates, see *Administering the system topic of SCCM User's Guide*.
- For information about customizing jobs, see *Administering data processing topic of SCCM User's Guide*.

Reconfiguring IBM Cloud Manager with OpenStack after updates

You must reconfigure IBM Cloud Manager with OpenStack after you make any updates.

After the IBM Cloud Manager with OpenStack topology is modified or updated by using the procedure that is described in Modifying or updating a cloud deployment, the configuration changes made at IBM Cloud Orchestrator installation time are reverted. To solve this issue, you must reconfigure the IBM Cloud Manager with OpenStack servers by running one of the procedures that are described in "Configuring the OpenStack servers" on page 33. If you run the advanced procedure, follow all the steps to reconfigure except the steps for adding roles, users, and projects to Keystone.

You must run the configuration after a deployed IBM Cloud Manager with OpenStack topology is updated by using the procedure that is described in Updating a deployed topology.

Ensure that you use the most recent version of IBM Cloud Orchestrator scripts to reconfigure IBM Cloud Manager with OpenStack. In case IBM Cloud Manager with OpenStack upgrade is required to support a more recent version of IBM Cloud Orchestrator, use the scripts from the target version of IBM Cloud Orchestrator as they might have latest updates.

Run the procedure on the following servers:

- On all the IBM Cloud Manager with OpenStack servers after:
  - An IBM Cloud Manager with OpenStack fix pack is deployed to the environment.
  - IBM Cloud Manager with OpenStack passwords or secrets are changed by using a redeployment.
  - Other redeployments are done on the IBM Cloud Manager with OpenStack topology by using the **`knife os manage update`** commands.
  - IBM Cloud Manager with OpenStack is reconfigured for secure communication in HTTPS.

  Note: Before you reconfigure IBM Cloud Manager with OpenStack, set the environment variables in the shell by running the command to use V2 APIs instead of V3APIs. For example, run the **`/root/openrc`** command to set the environment variables. If you run the **`/root/v3rc`** command before the reconfiguration is completed, the reconfiguration might fail. If you are not sure of the status, start a new shell.
- On the master controller and on the new controller after a new controller is added to the IBM Cloud Manager with OpenStack topology.
- On the new compute nodes after new compute nodes are added to the IBM Cloud Manager with OpenStack topology.

After you configure, verify whether IBM Cloud Orchestrator is working correctly. For more information, see "Verifying the installation" on page 68.

Installing high availability across multiple sites

IBM Cloud Orchestrator can be installed in high-availability configuration in two data centers in different sites, if the connection speed between them is as good as in a LAN.

The following requirements apply:

- Use DB2 HADR (high availability disaster recovery) replication to ensure data replication between the sites. For information about implementing DB2 HADR, see Initializing high availability disaster recovery (HADR).
- In the two sites, you must define the same subnets and VLANs and they must have IP addresses in the same range.
- All the IBM Cloud Orchestrator required ports must be opened between the IBM Cloud Orchestrator instances on one site to the other IBM Cloud Orchestrator instances in the secondary site. For more information, see "Ports used by IBM Cloud Orchestrator" on page 80.

A typical topology is shown in the following picture.

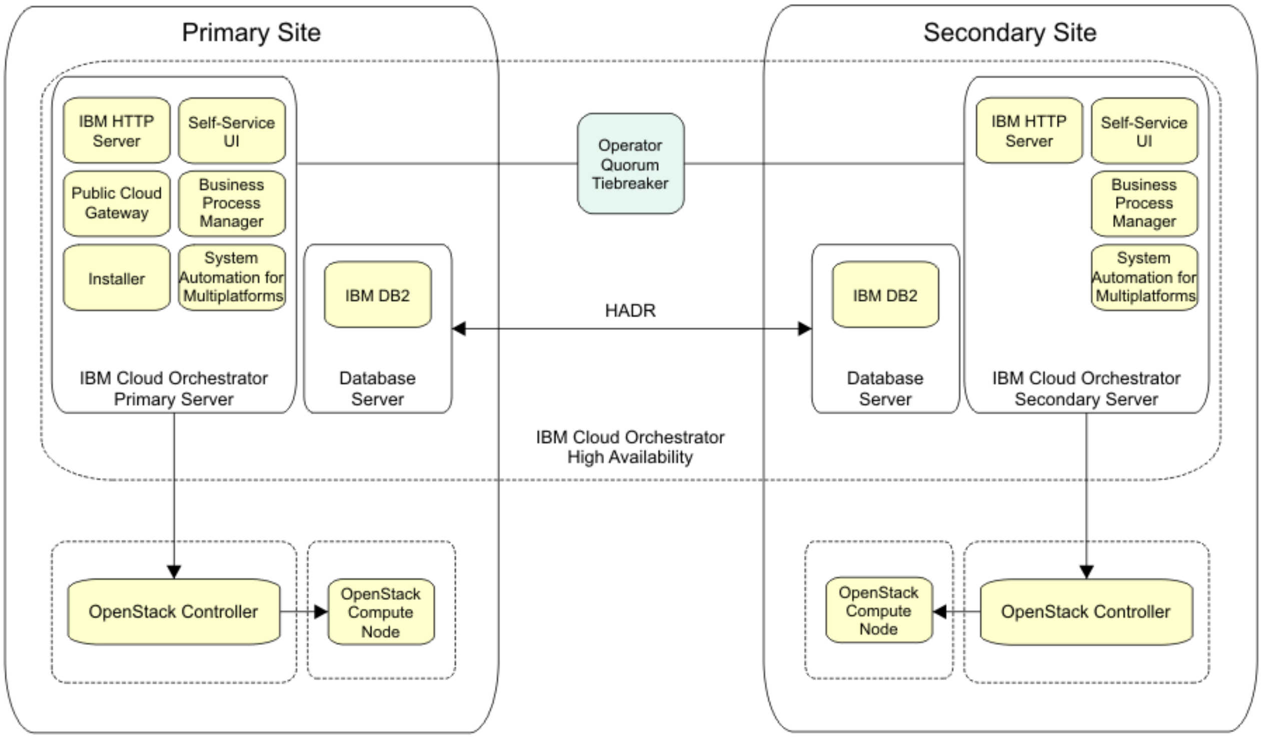


Figure 1. High availability across multiple sites

Note: The Public Cloud Gateway component is a single point of failure and it is not a highly-available component.

Note: IBM Cloud Orchestrator does *not* provide high availability for the OpenStack components. For information about high availability in IBM Cloud Manager with OpenStack, see the IBM Cloud Manager with OpenStack documentation.

The multisite high-availability configuration is installed like the standard high availability choosing nodes on different datacenter.

As an option, if you are using the default temporary folder for the installation (`/tmp`), you can copy the installation binaries to the secondary node to reduce the time spent to perform the binary transfer during installation by performing the following steps:

1. Download the Business Process Manager binaries in the `/tmp/bpm` directory.
2. Put all the other IBM Cloud Orchestrator binaries in the `/tmp` directory.

This configuration is active-active, with all the traffic managed by the load balancer that divides the load to both the sites as in standard high availability configuration. The high-availability configuration covers the failover of the IBM Cloud Orchestrator components with the exception of Public Cloud Gateway that is not in high availability and is a single point of failure.

If the connection between the primary data center and the secondary data center fails and the two nodes cannot connect to each other, a tiebreaker is needed to decide which node should take control and do the necessary automation actions to keep the resources available.

In the multi site high availability scenario, the operator is asked to act as a tiebreaker, granting quorum to the appropriate IBM Cloud Orchestrator servers in the primary or secondary data center. If you installed IBM Cloud Orchestrator in two sites, you must set the quorum tiebreaker to `Operator`. This configuration is needed to allow an administrator user to decide which node should receive quorum if a tie occurs due to a network failure or node failure, and the node cannot establish the quorum automatically. For information about setting the quorum tiebreaker, see r"Configuring the quorum tiebreaker" on page 73.

Configuring the quorum tiebreaker

Set the quorum tiebreaker to `Operator` if you require to give privilege to end user to decide which node should have quorum in case a tie occurs due to a network or node failure.

By default the EXEC quorum tiebreaker is used for IBM Cloud Orchestrator high availability. The gateway IP address is used as tiebreaker and it determines which part of the cluster gets quorum and can manage active resources. For more information about quorums and tiebreakers, see Operational quorum.

To verify what node has currently the quorum, run the `lssam` command and search for the node with `IBM.ServiceIP:cs-ip` that is in online status. For example, if the following output is displayed, the `primaryiconode` node has currently the quorum:

```
Online IBM.ResourceGroup:central-services-rg Nominal=Online
        |- Online IBM.Application:bpm-node
                |- Online IBM.Application:bpm-node:primaryiconode
                '- Online IBM.Application:bpm-node:secondaryiconode
        |- Online IBM.Application:bpm
                |- Online IBM.Application:bpm:primaryiconode
                '- Online IBM.Application:bpm:secondaryiconode
        |- Online IBM.Application:ihs
                |- Online IBM.Application:ihs:primaryiconode
                '- Online IBM.Application:ihs:secondaryiconode
        |- Online IBM.Application:scui
                |- Online IBM.Application:scui:primaryiconode
                '- Online IBM.Application:scui:secondaryiconode
        '- Online IBM.ServiceIP:cs-ip
                |- Online IBM.ServiceIP:cs-ip:primaryiconode
                '- Offline IBM.ServiceIP:cs-ip:secondaryiconode
Online IBM.ResourceGroup:pcg-rg Nominal=Online
        '- Online IBM.Application:pcg
                '- Online IBM.Application:pcg:primaryiconode
Online IBM.Equivalency:cs-network-equ
        |- Online IBM.NetworkInterface:ens192:primaryiconode
        '- Online IBM.NetworkInterface:ens192:secondaryiconode
```

To set the quorum tiebreaker to `Operator`, perform the following steps:

1. Make sure that both IBM Cloud Orchestrator nodes are up and can talk to each other:
2. Run the following command:

   ```
   chrsrc -c IBM.PeerNode OpQuorumTieBreaker="Operator"
   ```

   If the command runs successfully, no output is displayed.
3. You can verify if the quorum definition changed by running the following command:

   ```
   lsrsrc -c IBM.PeerNode OpQuorumTieBreaker
   ```

   The following output is displayed, for example:

```
Resource Class Persistent Attributes for IBM.PeerNode
resource 1:
        OpQuorumTieBreaker = "Operator"
```

4. If a tie occurs, run the following command to verify the quorum state:

   ```
   lssrc -ls IBM.RecoveryRM|grep Operational
   ```

   The following output is displayed if there is a tie:

   ```
   Operational Quorum State: PENDING_QUORUM
   ```

5. If there is a node or network failure, there is no automatic failover because the quorum is in `PENDING_QUORUM` state and the operator input is required.

   You can give quorum to a node and make the node as primary by running the following command:

   ```
   runact -c IBM.PeerDomain ResolveOpQuorumTie Ownership=1
   ```

   The following output is displayed, for example:

   ```
   Resource Class Action Response for ResolveOpQuorumTie
   ```

If you need to set the quorum tiebreaker to the default EXEC value for the IBM Cloud Orchestrator high availability, perform the following steps:

1. Run the following command to set the quorum tiebreaker to EXEC:

   ```
   chrsrc -c IBM.PeerNode OpQuorumTieBreaker="cs-network-tiebreaker"
   ```

   If the command runs successfully, no output is displayed.

2. Verify if the EXEC quorum tiebreaker is configured correctly by running the following command:

   ```
   lsrsrc -c IBM.PeerNode OpQuorumTieBreaker
   ```

   The following output is displayed, for example:

   ```
   Resource Class Persistent Attributes for IBM.PeerNode
   resource 1:
           OpQuorumTieBreaker = "cs-network-tiebreaker"
   ```

If the quorum tiebreaker is set to EXEC, if there is a tie, the quorum automatically decides which node is active, and failover and failback occur accordingly.

Troubleshooting the installation

Learn how to troubleshoot installation problems.

Known installation errors

- If errors occur because of prereq-checker.sh during installation or upgrade, see "Errors when you run prereq-checker.sh " on page 380.
- **IBM HTTP Server error during installation or upgrade**

  The following error can occur during installation or upgrade:

  ```
  ================================================================================
  Error executing action `run` on resource 'execute[start Admin Server]'
  ================================================================================

  Mixlib::ShellOut::ShellCommandFailed
  ------------------------------------
  Expected process to exit with [0], but received '3'
  ---- Begin output of /ico/ibm/ico/HTTPServer/bin/adminctl start ----
  STDOUT: /ico/ibm/ico/HTTPServer/bin/adminctl start: admin http could not be started
  ```

```
STDERR: Syntax error on line 7 of /ico/ibm/ico/HTTPServer/conf/admin.conf:
Port must be specified
---- End output of /ico/ibm/ico/HTTPServer/bin/adminctl start ----
Ran /ico/ibm/ico/HTTPServer/bin/adminctl start returned 3
```

The error can occur due to APAR PI85714. To resolve the error, upgrade the WebSphere Customization Toolbox version to 8.5.5.13 and then run the install or upgrade steps. For the binaries and the steps to install/upgrade, see WebSphere Customization Toolbox.

- **Backup and restore the ACL data during upgrade**

  If you want to back up ACL data for all the custom offering, then back up the ORCHESTRATOR.ACL database table. Restore it after upgrade so that all the offerings have proper ACL configuration.

- **Masking passwords in ico_install.rsp**

  After the installation is completed successfully, the passwords in ico_install.rsp are masked automatically. For example, ORCHESTRATOR_PASSWORD [HIDDEN]. If any of the passwords are not masked automatically, then you must manually make them hidden.

- **Validation error post upgrade from V2.5.0.5 HTTP to V2.5.0.6 HTTPS keystone topology**

  When you perform any action on the region or clone a virtual machine for the first time after you upgrade from V2.5.0.5 HTTP to V2.5.0.6 HTTPS keystone topology, the following error might get displayed:

  ```
  Region validation failed, hostname or credentials might be wrong. If you do not want this
  region, please delete it using delete region action.
  ```

  Ignore this error message and click **OK** to save the region.

- The SIMPLE_TOKEN_SECRET and OPENSTACK_SERVICES prerequisites might fail whenever the operating system is below RHEL 7.3.

  Do the following steps to resolve the error:

  1. Verify whether all values are correct in the ico_install.rsp.
  2. Update python package version to 2.7.5-48.el7 if the python package is at a lower version.

- During installation, ignore the following error message that might get triggered because of Chef:

  ```
  Error : . warning: already initialized constant Chef::Recipe::Delete_Cert
  warning: previous definition of Delete_Cert was here
  ```

- **Upgrade fails because of signer certificate issue**

  The IBM Cloud Orchestrator upgrade or installation command might fail because the CLI installation client is not able to connect to server process due to the signer certificate issue.

  As a workaround, manually run the command that fails and restart the upgrade without reverting the snapshots. Repeat this workaround for all failing commands.

- **Cannot find 32-bit library files libstdc++.so.6 and /lib/libpam.so***

  If the 32-bit library files cannot be found on the 64-bit operating system platform, the db2prereqcheck.log file might contain errors similar to the following errors:

  ```
  Validating "32 bit version of "libstdc++.so.6" " ...

  Found the 64 bit "/usr/lib64/libstdc++.so.6" in the following directory "/usr/
  lib64".

  DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library
  file:
            "libstdc++.so.6".

  Validating "/lib/libpam.so*" ...

  DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library
  file:
            "/lib/libpam.so*".
  ```

```
Validating "32 bit version of "libstdc++.so.6" " ...

DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library
file:
          "/lib/libpam.so*".

WARNING : Requirement not matched.
Requirement not matched for DB2 database "Server" . Version: "10.5.0.2".
Summary of prerequisites that are not met on the current system:

DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library
file:
          "/lib/libpam.so*".
```

The errors are not critical and can be ignored.

- **The domain name that is specified in the IBM Cloud Orchestrator deployment parameter must match the domain name format that is used in cookies** .

  Cookies that are used to implement user interface security features can be set only for domain names that are not top-level domain names and that adhere to the public suffix list.

  To resolve this problem, provide a domain name that matches the cookie requirements.

Error in IBM Cloud Orchestrator installation on Red Hat Enterprise Linux

The IBM Cloud Orchestrator HA installation on Red Hat Enterprise Linux V7.2 might fail with `DB_VERSION_MISMATCH - rpmdb open failed` error.

```
STDOUT:
STDERR: error: db5 error(-30969) from dbenv->open: BDB0091
DB_VERSION_MISMATCH: Database environment version mismatch
error: cannot open Packages index using db5 -  (-30969)
error: cannot open Packages database in /var/lib/rpm
yum-dump General Error: Error: rpmdb open failed
```

Do the following steps to resolve the error:

Note: For IBM Cloud Orchestrator HA installation, run this command on both nodes.

1. Run the following command to update the `libdb-5.3.28-14.ibm.el7.x86_64` package:

   `yum install -y libdb-5.3.28-14.ibm.el7.x86_64`
2. Run the following command to rebuild the RPM database:

   `rpmdb --rebuilddb`
3. Confirm whether the **`yum repolist`** command is successful.

   `yum repolist`
4. Install IBM Cloud Orchestrator.

Cannot create the external database

When you try to create the external database, the **`create_dbs.sh`** command fails with an error `Failed to create database` *`database_name`* `with user` *`user_name`*.

Reason:

IBM DB2 is stopped.

Solution:

Start IBM DB2.

Example:

```
# su - db2inst1
$ db2stop
11/20/2014 02:49:33     0   0   SQL1032N  No start database manager command was issued.
SQL1032N  No start database manager command was issued.  SQLSTATE=57019
```

```
$ db2start
11/20/2014 02:49:46     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
$ exit
logout
# ./create_dbs.sh central
Creating databases: 5/8
Creating databases: 6/8
```

High-availability upgrade fails while enabling JDK 1.7

While upgrading a high-availability setup to 2.5.0.10, sometimes high-availability does not come online in the final phases of the upgrade, because Business Process Manager services do not start correctly.

Reason

The root cause is that sometimes Dmgr node fails to stop all the processes correctly. The following error is seen in the installation log file, especially while enabling the JDK 1.7.

```
ERROR: execute[Java7 stop Dmgr]
(bpm::bpm_upgrade_enablejdk line 49) had an error:
Mixlib::ShellOut::ShellCommandFailed: Expected process to exit with [0],
but received '255'---- Begin output of
/opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/bin/stopServer.sh dmgr ----
STDOUT: ADMU0116I: Tool information is being logged in file
/opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/logs/dmgr/stopServer.log
ADMU0128I: Starting tool with the DmgrProfile profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU3111E: Server stop requested but failed to complete.
ADMU0111E: Program exiting with error:
   com.ibm.websphere.management.exception.AdminException:
ADMU3060E: Timed out waiting for server shutdown.
ADMU1211I: To obtain a full trace of the failure, use the -trace option.
ADMU0211I: Error details may be seen in the file:
         /opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/logs/dmgr/stopServer.log
STDERR:
----End output of/opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/bin/stopServer.sh dmgr ----
Ran /opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/bin/stopServer.sh dmgr
returned 255; ignore_failure is set, continuing
```

Solution

It is advised to revert to the earlier snapshot of the system and rerun the upgrade after commenting the step to enable JDK 1.7 automatically as part of the upgrade. JDK 1.7 gets installed, but needs to be manually enabled. Steps for both commenting the JDK 1.7 enablement during upgrade, as well as manually enabling it are listed below.

Changing upgrade process to disable JDK 1.7

To remove the step, which automatically enables the JDK 1.7 from the installation procedure, comment out the following line `include_recipe 'bpm::bpm_upgrade_enablejdk` from the `default.rb` from the following directory `data/orchestrator-chef-repo/chef-repo/cookbooks/bpm/recipes` of the installer and then rerun the installer.

Manually enabling the JDK

After the installation is successful, you need to manually enable JDK 1.7 for Business Process Manager.

The following steps must be executed sequentially on the primary and slave nodes.

Note: Sometimes while stopping the Business Process Manager nodes and servers, the Business Process Manager processes are not killed properly. In this scenario, the processes need to be manually killed before restarting the services.

Switch TSAMP to manual mode using the following command:

```
samctrl -M t
```

On the primary node

1. From the Business Process Manager Home/bin directory (by default /opt/ibm/ico/BPM/v8.5/bin), run the following commands:

   a. Set Java7 as default for new profiles.

   ```
   managesdk.sh -setNewProfileDefault -sdkName 1.7_64
   ```

   b. Set Java7 as default for scripting operations.

   ```
   managesdk.sh -setCommandDefault -sdkName 1.7_64
   ```

   c. Set Java7 as default for Dmgr profile.

   ```
   managesdk.sh -enableProfile -profileName DmgrProfile -sdkname 1.7_64 -enableServers
   ```

   d. Set Java7 as default for Node1 profile.

   ```
   managesdk.sh -enableProfile -profileName Node1Profile -sdkname 1.7_64 -enableServers
   ```

2. From the Node1Profile/bin directory (by default /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/bin), run the following commands:

   a. Stop the node.

   ```
   stopNode.sh
   ```

   b. Sync the node.

   ```
   syncNode.sh #{bpm_server_hostname} 8879
   ```

   where #{bpm_server_hostname} is the host name of the primary server.

   c. Stop the server.

   ```
   stopServer.sh SingleClusterMember1
   ```

3. From the DMgr/bin directory (by default /opt/ibm/ico/BPM/v8.5/profiles/DMgrProfile/bin), run the following commands:

   a. Stop the Dmgr.

   ```
   stopServer.sh dmgr
   ```

   b. Start the Dmgr.

   ```
   startServer.sh dmgr
   ```

4. From the Node1Profile/bin directory (by default /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/bin), run the following commands:

   a. Start the node.

   ```
   startNode.sh
   ```

   b. Start the server.

   ```
   startServer.sh SingleClusterMember1
   ```

On the secondary node

1. From the Business Process Manager Home/bin directory (by default /opt/ibm/ico/BPM/v8.5/bin), run the following commands:

   a. Set Java7 as default for Node2 profile.

```
managesdk.sh -enableProfile -profileName Node2Profile -sdkname 1.7_64 -enableServers
```

b. Set Java7 as default for new profiles on Node2.

```
managesdk.sh -setNewProfileDefault -sdkName 1.7_64
```

c. Set Java7 as default for scripting operations on Node2.

```
managesdk.sh -setCommandDefault -sdkName 1.7_64
```

2. From the `Node2Profile/bin` directory (by default `/opt/ibm/ico/BPM/v8.5/profiles/Node2Profile/bin`), run the following commands:

   a. Stop node on Node2:

   ```
   stopNode.sh
   ```

   b. Sync node on Node2.

   ```
   syncNode.sh #{bpm_server_hostname_HA} 8879
   ```

   where `#{bpm_server_hostname_HA}` is the name of the master server.

   c. Start node on Node2.

   ```
   startNode.sh
   ```

   d. Stop Server on Node2.

   ```
   stopServer.sh SingleClusterMember2
   ```

   e. Start Server on Node2.

   ```
   startServer.sh SingleClusterMember2
   ```

After these steps are executed, make sure the Business Process Manager is up and running properly. Change the TSAMP control back to automatic using the following command:

```
samctrl -M f
```

Installation reference

Learn more about the installation process, including customizations, modifications, and tips.

System files modified by the installation procedure

Several system files are modified during the IBM Cloud Orchestrator installation.

During the IBM Cloud Orchestrator installation, the following system files are modified on the IBM Cloud Orchestrator Server:

Table 11. System files modified by the installation procedure

File	Updates
`/etc/passwd`	The following users are added to the `/etc/passwd` file: • `bpmuser` • `db2das1` • `db2fenc1` • `db2inst1` • `ihsadmin` • `pcg` • `scui`
`/etc/group`	The following groups are added to the `/etc/group` file: • `bpmuser` • `db2das1` • `db2fenc1` • `db2inst1` • `ihsadmin` • `pcg` • `scui`

Ports used by IBM Cloud Orchestrator

This topic lists the ports that are opened by IBM Cloud Orchestrator when running the default configuration. The actual port usage might differ if the configuration was changed. Configure your firewall to allow inbound traffic to the port that must accept incoming communication.

Table 12. Ports used by IBM Cloud Orchestrator

Port	Protocol	Program	User	Incoming external hosts
50000	HTTP	db2sysc	root	IBM Cloud Orchestrator Server, OpenStack Controllers, OpenStack Neutron servers **Note:** Port 50000 is used for HTTP and port 50001 for HTTPS. The default is HTTPS.
523	IPC (TCP)	db2dasrrm	root	

Table 12. Ports used by IBM Cloud Orchestrator (continued)

Port	Protocol	Program	User	Incoming external hosts
7060, 7277, 9352, 9402, 9420, 9809, 11006	TCP	WebSphere Deployment Manager	root	
8879	SOAP	WebSphere Deployment Manager	root	
9043	HTTPS	WebSphere Deployment Manager	root	HTTP and HTTPS client using Administrator profile
9060, 9403	HTTP	WebSphere Deployment Manager	root	
9100	ORB	WebSphere Deployment Manager	root	
9632	IPC (TCP)	WebSphere Deployment Manager	root	
2809, 7062, 7272, 9353, 11004	TCP	WebSphere Node Agent	root	
8878	SOAP	WebSphere Node Agent	root	
9201, 9202	RMI/ IIOP, SSL	WebSphere Node Agent	root	
9629	IPC (TCP)	WebSphere Node Agent	root	

Table 12. Ports used by IBM Cloud Orchestrator (continued)

Port	Protocol	Program	User	Incoming external hosts
9900	ORB	WebSphere Node Agent	root	
9420, 11006, 9632		Deployment Manager (inside Business Process Manager)		
7062, 11004, 9629		The node agent used by Business Process Manager		
7276, 7286, 9044, 9101, 9354, 11008	TCP	Business Process Manager	root	
8880	SOAP	Business Process Manager	root	
9061	HTTP	Business Process Manager	root	
9633		Business Process Manager		
9080	HTTP	Business Process Manager	root	Public Cloud User, HTTP and HTTPS client using Administrator profile
9405, 9406	RMI/IIOP. SSL	Business Process Manager	root	
9443	HTTPS	Business Process Manager	root	HTTP and HTTPS client using Administrator profile
9633	IPC (TCP)	Business Process Manager	root	
9810	ORB	Business Process Manager	root	
7443	HTTPS	Self-service user interface	root	Public Cloud User, HTTP and HTTPS client using Administrator profile

Table 12. Ports used by IBM Cloud Orchestrator (continued)

Port	Protocol	Program	User	Incoming external hosts
9797	HTTP	Public Cloud Gateway	pcg	
443	HTTPS	IBM HTTP Server	root	Public Cloud User, HTTP and HTTPS client using Administrator profile, OpenStack Controllers
8008, 8480	HTTP	IBM HTTP Server	root ihsadmin	
5000		Used to connect to Keystone		
5001	UPD6	Business Process Manager		
22	SSH	Used for general operations and to communicate with the deployed virtual machines		**Note:** Using a different port for SSH may cause unexpected results with normal functioning of IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack.
3389	RXA	Used to communicate with the deployed virtual machines		
445	HTTPS	Used for general operations with the deployed virtual machines		
35357	EMPTY	BYOOS	root	

Note: The ports used by System Automation for Multiplatforms are specified at https://www.ibm.com/support/knowledgecenter/en/SSRM2X_3.2.2/com.ibm.samp.doc_3.2.2/welcome.html.

Token configuration parameter

List of token configuration parameters.

Table 13. Token configuration parameter

Name	Value	Description
`assertion_expiration_time`	3600 (IntOpt)	Default Time to Leave (TTL), in seconds, for any generated SAML assertion created by Keystone. SAML assertion Contains information about a user as provided by an IdP. It is an indication that a user is authenticated.

Table 13. Token configuration parameter (continued)

Name	Value	Description
`token_expiration`	Number in seconds	The time at which the token becomes invalid (expires), in seconds.

Chapter 3. Upgrading

To upgrade IBM Cloud Orchestrator, complete one of the following procedures.

Migrating from IBM Cloud Orchestrator V2.4.0.2 or later

You can migrate a non high-availability IBM Cloud Orchestrator environment with VMware, KVM, or PowerVC regions which use Neutron networks from IBM Cloud Orchestrator V2.4 Fix Pack 2 or later fix pack to IBM Cloud Orchestrator V2.5.0.10 which uses IBM Cloud Manager with OpenStack as OpenStack distribution.

If you want to migrate from a version older than 2.4.0.2, you must first upgrade to IBM Cloud Orchestrator V2.4.0.2. For more information about upgrading to IBM Cloud Orchestrator V2.4.0.2, see Upgrading. If you want to upgrade from IBM Cloud Orchestrator V2.5, V2.5.0.1, V2.5.0.1 interim fix 1, or V2.5.0.2, see "Upgrading from IBM Cloud Orchestrator V2.5" on page 106.

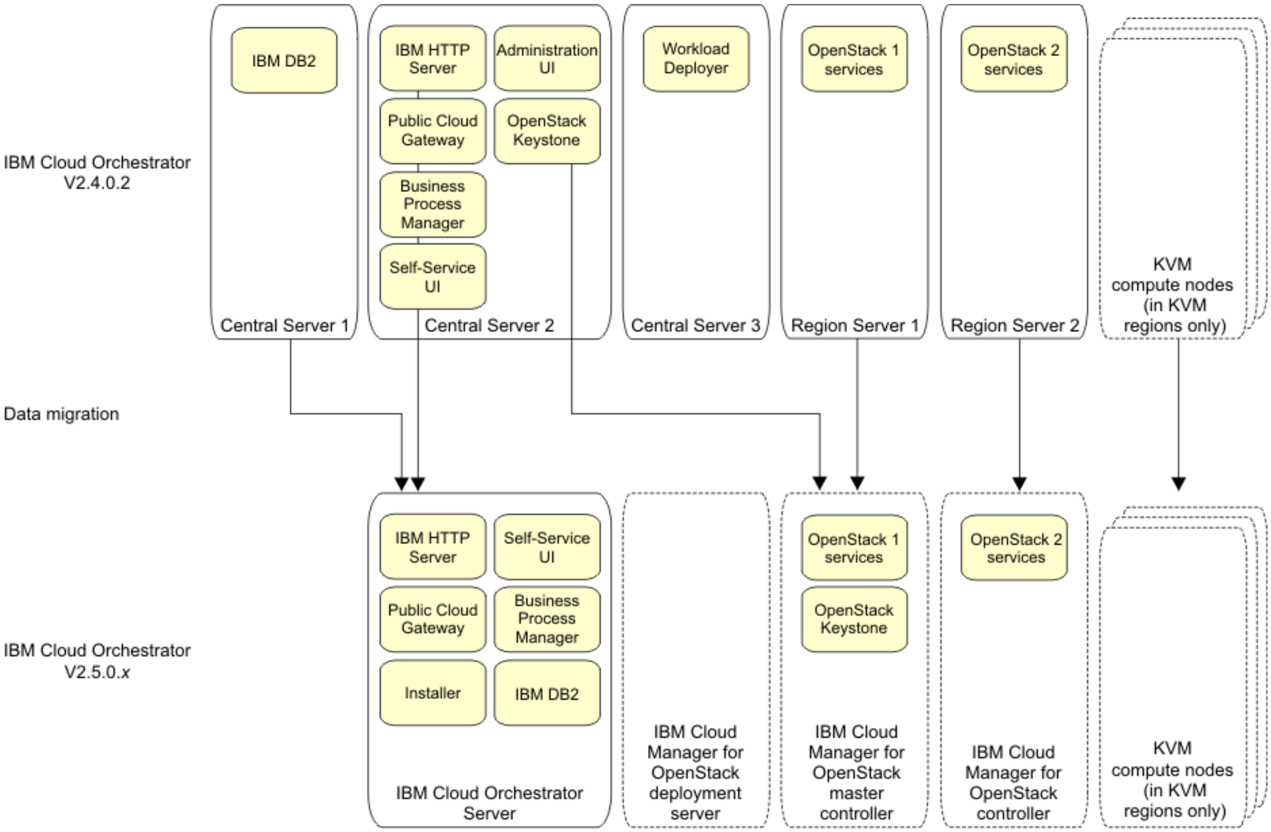
The following IBM Cloud Orchestrator V2.4.0.2 configurations or resources are *not* supported for migration to IBM Cloud Orchestrator V2.5.0.10:

- IBM Cloud Orchestrator systems in high-availability configuration.
- Regions that use z/VM or HyperV hypervisors.
- Regions that use Nova networks.
- Virtual system patterns that are defined by using Workload Deployer. Existing deployed patterns are migrated as individual instances. In IBM Cloud Orchestrator V2.5.0.10, you can use OpenStack Heat templates.
- Any customization that was done to the OpenStack configuration files in the IBM Cloud Orchestrator V2.4.0.2 environment to support a specific configuration (for example, the VMware advanced configuration). You must manually reapply your customization after you deploy the IBM Cloud Manager with OpenStack controllers.

  Important: Before you reapply your OpenStack customization, see the hypervisor details in Prerequisites for IBM Cloud Manager with OpenStack section.
- You cannot reuse the host name of the source environment in the upgraded V2.5.0.10 environment.

Migration overview

During the migration, data from the servers in the IBM Cloud Orchestrator V2.4.0.2 environment is ported to the new servers in the IBM Cloud Orchestrator V2.5.0.10 environment as shown in the following picture.



- The IBM Cloud Orchestrator V2.5.0.10 environment uses IBM Cloud Manager with OpenStack on the Kilo release whereas the IBM Cloud Orchestrator V2.4.0.2 environment used OpenStack on the Icehouse release.
- Keystone data is migrated from the IBM Cloud Orchestrator V2.4.0.2 Central Server 2 to the IBM Cloud Manager with OpenStack master controller in the IBM Cloud Orchestrator V2.5.0.10 environment. The remaining OpenStack data is migrated from the Region Servers in the IBM Cloud Orchestrator V2.4.0.2 environment to the IBM Cloud Manager with OpenStack controllers in the IBM Cloud Orchestrator V2.5.0.10 environment.
- When migrating VMware regions, the existing vCenters and virtual machines are used in the new IBM Cloud Orchestrator V2.5.0.10 environment. The virtual machines are unaffected by the migration.
- When migrating KVM regions, the KVM compute nodes are migrated to new compute nodes in the IBM Cloud Orchestrator V2.5.0.10 environment. Resources such as virtual machines, disks, and volumes on the IBM Cloud Orchestrator V2.4.0.2 compute nodes are cloned to new virtual machines, disks, and volumes on the new IBM Cloud Orchestrator V2.5.0.10 compute nodes before being re-enabled. The virtual machines are stopped prior to migration and are in a powered off state on migration completion. Depending on the number and size of the virtual machines and volumes deployed on the compute nodes in the region, the end-to-end migration may take a number of hours.

Migration checklist

Use this checklist to ensure that you complete all of the migration steps in the correct order.

Table 14. Migration Checklist

Step	Description	Done?	Comment
Planning the migration			
1	“Checking prerequisites” on page 88		
2	“Choosing the migration strategy” on page 89		
“Preparing for the migration” on page 90			
3	Installing the IBM Cloud Manager with OpenStack deployment server		
4	Preparing the IBM Cloud Orchestrator V2.5.0.10 Server and download the required image files		
5	Preparing the IBM Cloud Manager with OpenStack controller for the first region and deploy an IBM Cloud Manager with OpenStack cloud		
6	Configuring the IBM Cloud Manager with OpenStack controller for the first region		
7	Installing and configuring the IBM Cloud Orchestrator V2.5.0.10 Server		
8	[For IBM Cloud Orchestrator Enterprise Edition only:] Upgrading SmartCloud Cost Management		
9	[Optional:] Backing up the newly installed environment		
10	Preparing the upgrade attribute mapping file		
11	Discovering the IBM Cloud Orchestrator topologies		
12	Migrating the images from the IBM Cloud Orchestrator V2.4.0.2 Region Server		

Table 14. Migration Checklist (continued)

Step	Description	Done?	Comment
13	Preparing the migration for the remaining regions. You must run the following steps for each region to be migrated. You may choose to prepare the remaining regions for migration at any time before migrating them in step 15.		
	a. Preparing the IBM Cloud Manager with OpenStack controller and deploy an IBM Cloud Manager with OpenStack cloud		
	b. Configuring the IBM Cloud Manager with OpenStack controller		
	c. Preparing the upgrade attribute mapping file		
	d. Discovering the IBM Cloud Orchestrator topologies		
	e. Migrating the images from the IBM Cloud Orchestrator V2.4.0.2 Region Server		
Migrating			
14	"Migrating regions" on page 93		
15	"Migrating IBM Cloud Orchestrator data" on page 102		
16	*Import toolkit section of Content Development Guide*		
17	"Migrating the remaining regions" on page 102		
18	"Completing the migration" on page 103		

Migrating a non high-availability environment

To migrate a non high-availability environment, perform the following steps.

1. "Planning the migration" on page 88
2. "Preparing for the migration" on page 90
3. "Migrating regions" on page 93
4. "Migrating IBM Cloud Orchestrator data" on page 102
5. "Migrating the remaining regions" on page 102
6. "Completing the migration" on page 103

Planning the migration

Before starting the migration, ensure that the system prerequisites are met and choose the migration strategy to be followed.

Checking prerequisites

Before starting the migration, ensure that:

- The configuration of the existing IBM Cloud Orchestrator V2.4.0.2 environment is supported for migration to IBM Cloud Orchestrator V2.5.0.10. For more information, see "Migrating from IBM Cloud Orchestrator V2.4.0.2 or later" on page 85.
- You have a server, where you will install the IBM Cloud Orchestrator V2.5.0.10 Server, that meets the hardware and software prerequisites described in "Checking the hardware prerequisites" on page 21 and "Checking the software prerequisites" on page 22.
- You have the servers, where you will install IBM Cloud Manager with OpenStack. You must install one IBM Cloud Manager with OpenStack Deployment Server and an IBM Cloud Manager with OpenStack controller for each IBM Cloud Orchestrator V2.4.0.2 Region Server. For information about IBM Cloud Manager with OpenStack prerequisites, see IBM Cloud Manager with OpenStack prerequisites and "Prerequisites for IBM Cloud Manager with OpenStack" on page 18.
- If you are migrating KVM regions, you have new servers which will act as compute nodes in the IBM Cloud Orchestrator V2.5.0.10 environment. The existing IBM Cloud Orchestrator V2.4.0.2 KVM compute nodes are not reused in IBM Cloud Orchestrator V2.5.0.10.

  Note: If you are migrating VMware regions, the existing IBM Cloud Orchestrator V2.4.0.2 vCenters are reused in the IBM Cloud Orchestrator V2.5.0.10 environment.
- If you are migrating KVM regions and you use shared storage on your compute nodes, you have details of the configuration available. For more information, see "Migrating KVM compute nodes" on page 98.
- You have the `root` and `db2inst` credentials for the IBM Cloud Orchestrator V2.4.0.2 and V2.5.0.10 servers.
- Each IBM Cloud Manager with OpenStack controller is on the same network as the IBM Cloud Orchestrator V2.4.0.2 Region Server from which it is migrated.
- The `expect`, `python-requests`, `python-dateutil`, `pytz`, `rsync`, and `zip` packages are installed on the following servers:
  - The IBM Cloud Orchestrator V2.4.0.2 Deployment Server, Central Server 2 and Region Servers
  - The IBM Cloud Orchestrator V2.5.0.10 Server
  - The IBM Cloud Manager with OpenStack Deployment Server and Controller

Choosing the migration strategy

Before starting the migration, take the following decisions based on your environment:

- Choose the configuration of your servers in the IBM Cloud Orchestrator V2.5.0.10 environment (server size, networks, server names, domains, IP addresses) according to your current IBM Cloud Orchestrator V2.4.0.2 environment. For more information, see the "Planning the installation" on page 14 section.
- Choose the region to be migrated first. The first migrated region will be the IBM Cloud Manager with OpenStack master controller, that is the controller which hosts the Keystone component which is shared with all the other IBM Cloud Manager with OpenStack controllers. For information about the multi-region environment, see Deploying multi-region support.

  A staged migration of the OpenStack regions allows the IBM Cloud Orchestrator V2.4.0.2 and V2.5.0.10 environments to continue to work in parallel until the migration is fully completed. In this case, the outage of the region management is minimized.

  Be aware that the migration of historical instance data like, for example, request history and workflow, process and Inbox data, is not supported. Workflows that are not in `complete` state before the migration are not supported in the new migrated environment so you will need to ensure that all the workflows that you want to migrate are in `complete` state before the migration. For information about exporting the request history from the IBM Cloud Orchestrator V2.4.0.2 database, see step "4" on page 102 in the procedure that is described in "Migrating IBM Cloud Orchestrator data" on page 102.
- The custom toolkits are not migrated automatically. To migrate the custom toolkit, see *Import toolkit section of Content Development Guide*. The procedure must be done after the migration of the first region and IBM Cloud Orchestrator data.

- Choose the order that you want to perform the preparation of regions for migration. You can choose to prepare the second and subsequent regions for migration before or after you migrate the first region. Preparation in advance allows you to migrate more regions in the same outage period.
- Consider cleaning up your IBM Cloud Orchestrator V2.4.0.2 environment in order to save migration time and resource usage in future. For example, you might choose to remove unused virtual machines, virtual machines created in error, unused images, or images that are not compatible with IBM Cloud Orchestrator V2.5.0.10.

Preparing for the migration

To prepare your environment for migration, you must install a new IBM Cloud Orchestrator V2.5.0.10 environment which is in place in parallel with your IBM Cloud Orchestrator V2.4.0.2 environment by running the following procedure.

Before you begin

On the IBM Cloud Orchestrator V2.4.0.2 system, take note of the following items:

- The availability zones that are added to each domain and project. You must manually recreate them after the migration.
- The Domain quotas for the `default` domain. You must manually adjust the quotas for the `default` domain after the migration.
- The Amazon EC2 instances that were created by the `admin` user. You must manually adjust the Amazon EC2 instances for the `admin` user after the migration.

Procedure

1. Install the IBM Cloud Manager with OpenStack deployment server and apply the latest fix pack.

   For information about installing the IBM Cloud Manager with OpenStack deployment server, see Installing IBM Cloud Manager with OpenStack on Linux. For information about applying the latest fix pack, see Applying fixes and updates.
2. Prepare the IBM Cloud Orchestrator V2.5.0.10 Server and download the required image files as described in "Preparing the IBM Cloud Orchestrator Servers " on page 29 and "Downloading the required image files" on page 27.
3. Run of the following procedures depending on the region that you are migrating:

   - [For KVM and VMware regions:] Prepare the IBM Cloud Manager with OpenStack controller for the first region and deploy an IBM Cloud Manager with OpenStack cloud according to the topology of the first IBM Cloud Orchestrator V2.4.0.2 region that you want to migrate. For information, see Deploying an IBM Cloud Manager with OpenStack cloud.

     Note: The first region is migrated to the IBM Cloud Manager with OpenStack master controller that will also contain the Keystone component shared among the regions.

     Important: When preparing for the migration of a KVM region, ensure that the compute node topology being created for the IBM Cloud Orchestrator V2.5.0.10 environment is the same as the topology present on the IBM Cloud Orchestrator V2.4.0.2 environment. In particular, ensure that there are the same number of compute nodes in both the environments and that the IBM Cloud Orchestrator V2.5.0.10 compute nodes have sufficient capacity to contain the virtual machines being migrated to them.
   - [For PowerVC regions:] Perform the following steps:

     a. Log in to the PowerVC server V1.2.1.2 in the IBM Cloud Orchestrator environment and back up data by running the **powervc-backup** command. The `powervc-backup.tar.gz` file is created. For more information, see Backing up IBM Power® Virtualization Center data.

     b. Copy the `powervc-backup.tar.gz` file to another backup machine.

     c. Shut down the PowerVC server V1.2.1.2.

     d. Install and configure a new PowerVC server V1.2.3.1.

Note: For the new PowerVC server V1.2.3.1, use the same IP address that was used for the PowerVC server V1.2.1.2.

   e. Copy the `powervc-backup.tar.gz` file to the PowerVC server V1.2.3.1.
   f. Log in to the PowerVC server V1.2.3.1 and restore the back up file by running the **`powervc-restore`** command. For more information, see Recovering IBM Power® Virtualization Center data.
   g. Deploy an IBM Cloud Manager with OpenStack cloud environment to manage the new PowerVC server V1.2.3.1. For information, see Deploying an IBM Cloud Manager with OpenStack cloud.
   h. After the PowerVC region is installed and configured, shut down the PowerVC server V1.2.3.1.
   i. Power on the PowerVC Server V1.2.1.2.

4. Configure the IBM Cloud Manager with OpenStack controller for the first region to be managed by IBM Cloud Orchestrator V2.5.0.10 by running the procedure described in "[Typical] Configuring the IBM Cloud Manager with OpenStack servers" on page 33.
5. Install the IBM Cloud Orchestrator V2.5.0.10 Server by running the following procedures:
   a. "Setting the deployment parameters" on page 60
   b. "Adding the OpenStack simple token to the response file" on page 64
   c. "Checking the installation prerequisites" on page 65
   d. "Deploying the IBM Cloud Orchestrator Servers" on page 66

   Note: The commands used in this procedure assume that the `/opt/ico_install/2.5.0-CSI-ICO-FP00010` installation directory is used when installing IBM Cloud Orchestrator V2.5.0.10. Replace this value with the appropriate value for your installation.
6. [For IBM Cloud Orchestrator Enterprise Edition only:] Upgrade SmartCloud Cost Management by following the procedure in *Upgrading from SmartCloud Cost Management 2.10.5 to 2.10.6 ifix07 SCCM User's Guide*.
7. Optional: Back up the newly installed IBM Cloud Manager with OpenStack servers and IBM Cloud Orchestrator V2.5.0.10 Server. If the migration procedure fails, you can recover the newly installed environment and restart the migration procedure from the following step.
8. Prepare the upgrade attribute mapping file to specify region environmental information which the discovery process is unable to retrieve automatically.

   On the IBM Cloud Orchestrator V2.5.0.10 Server, as user `root`, run the following commands to edit the `upgrade_map.csv` file:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/upgrade
   vi upgrade_map.csv
   ```

   The `upgrade_map.csv` file is a two-column CSV file. The first column contains attributes from the source system, the second column contains attributes from the destination system. In the `upgrade_map.csv`, you must specify the following information:

   Physical network mapping

   OpenStack Neutron has a concept of a `physical_network` that is configured in the Neutron plug-in configuration files. Each plug-in (for example, ML2 or Open vSwitch) configures one or more physical networks to match the underlying network topology. Before the migration, you must configure the Neutron plug-ins on the destination system to match the source system. The names of the physical networks does not need to be the same, but if they are different, you must add the related entry to the `upgrade_map.csv` file.

   For example, if you configured a Cisco Nexus switch (via the `/etc/neutron/plugins/ml2/ml2_conf_cisco.ini` file) as `physnet0` on the source system, and it is called `nexus0` on the destination system, you must add the following line to the `upgrade_map.csv` file:

   ```
   physnet0,nexus0
   ```

In the upgrade_map.csv file, you must add all the instances where the source physical network is different from the destination physical network.

[For KVM regions only:] Hypervisor host name mapping

Neutron uses the binding:host_id attribute to map network ports to hypervisors. This attribute contains the host name of the KVM hypervisor, and it is used for message routing purposes. For each KVM compute node to be migrated, you must add an entry to the upgrade_map.csv file.

For example, if you are migrating from the source KVM compute node compute00 to the destination KVM compute node compute11, you must add the following line to the upgrade_map.csv file:

```
compute00,compute11
```

Note: For PowerVC regions, you do not need to populate the upgrade_map.csv file. You must create an empty upgrade_map.csv file.

9. Discover the current IBM Cloud Orchestrator V2.4.0.2 and IBM Cloud Orchestrator V2.5.0.10 topologies by running the following procedure:

   a. On the IBM Cloud Orchestrator V2.5.0.10 Server, as user root, run the following commands to edit the ico_upgrade.rsp:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   vi ico_upgrade.rsp
   ```

   and specify the following information:

   DEPLOYMENT_SERVER
   The IP address or the fully qualified domain name (FQDN) of the IBM Cloud Orchestrator V2.4.0.2 Deployment Server.

   DEPLOYMENT_SERVER_PASSWORD
   The root password of the IBM Cloud Orchestrator V2.4.0.2 Deployment Server.

   ICM_DEPLOYMENT_SERVER
   The IP address or the fully qualified domain name (FQDN) of the IBM Cloud Manager with OpenStack deployment server.

   ICM_DEPLOYMENT_SERVER_PASSWORD
   The root password of the IBM Cloud Manager with OpenStack deployment server.

   SOURCE_ICO_SERVER
   The IP address or the fully qualified domain name (FQDN) of the IBM Cloud Orchestrator V2.4.0.2 Central Server 2.

   DEST_ICO_SERVER
   The fully qualified domain name (FQDN) of the IBM Cloud Orchestrator V2.5.0.10 Server.

   DEST_ICO_SERVER_PASSWORD
   The root password of the IBM Cloud Orchestrator V2.5.0.10 Server.

   b. Perform the upgrade discovery prerequisite check on the system that you have specified in the upgrade response file by running the following command:

   ```
   ./upgrade-prereq-checker.py ico_upgrade.rsp
   ```

   If the prerequisite check passes successfully, continue with the next step. Otherwise, follow the instructions outputted by the prerequisite checker and edit the upgrade response file again to fix any issues.

   c. Discover the topologies by running the following command:

   ```
   ./upgrade.py ico_upgrade.rsp --discover
   ```

   The discovery reports are created in the /tmp/discovery directory.

View the report of migrated regions, unmigrated regions, and unattached pre-prepared regions by viewing the `discoveryMigrationReport.html` report file in a browser. Ensure that the details are correct. At this stage, there are no entries in the `Migrated Regions` section.

10. Migrate all the images that are currently registered in Glance on the IBM Cloud Orchestrator V2.4.0.2 Region Server to the IBM Cloud Manager with OpenStack controller by running the following procedure:

    a. On the IBM Cloud Orchestrator V2.5.0.10 Server, as user `root`, run the following commands to edit the `ico_upgrade.rsp` file:

    ```
    cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
    vi ico_upgrade.rsp
    ```

    and specify the following information according to the details in the discovery report:

    `SOURCE_UNMIGRATED_REGION`
    The fully qualified domain name (FQDN) of the IBM Cloud Orchestrator V2.4.0.2 Region Server to migrate from.

    `SOURCE_CENTRAL_DB_PASSWORD`
    The `root` password for the IBM Cloud Orchestrator V2.4.0.2 server on which the DB2 database resides. By default it is Central Server 1, otherwise it is where your external database resides.

    `SOURCE_REGION_PASSWORD`
    The `root` password for the IBM Cloud Orchestrator V2.4.0.2 Region Server.

    `SOURCE_CENTRAL_SERVER_PASSWORD`
    The `root` password for the IBM Cloud Orchestrator V2.4.0.2 Central Server 2.

    `SOURCE_ICO_ADMIN_PASSWORD`
    The password of the IBM Cloud Orchestrator V2.4.0.2 `admin` user.

    `DEST_UNATTACHED_REGION`
    The fully qualified domain name (FQDN) of the IBM Cloud Manager with OpenStack controller.

    `DEST_REGION_PASSWORD`
    The `root` password of the IBM Cloud Manager with OpenStack controller.

    `DEST_ICO_ADMIN_PASSWORD`
    The password of the IBM Cloud Orchestrator V2.5.0.10 `admin` user.

    b. Perform the upgrade migration prerequisite check on the system specified in the upgrade response file by running the following command:

    ```
    ./upgrade-prereq-checker.py ico_upgrade.rsp --check-regions
    ```

    If this check passes successfully, continue with the next step. Otherwise, follow the instructions given in output by the prerequisite checker and edit the upgrade response file to fix any issues.

    c. Migrate the images by running the following command:

    ```
    ./upgrade.py ico_upgrade.rsp --copy-images
    ```

11. If you have more than one region, you may choose to prepare the second and subsequent regions for migration at any time before migrating them. Preparation in advance allows you to migrate more regions in the same outage period.

    To prepare the migration for the remaining regions, repeat steps “3” on page 90, “4” on page 91, “8” on page 91, “9” on page 92, and “10” on page 93 for each region that you want to migrate.

Migrating regions

Migrate the regions from the IBM Cloud Orchestrator V2.4.0.2 environment to the IBM Cloud Manager with OpenStack controllers in your IBM Cloud Orchestrator V2.5.0.10 environment.

Before migrating the regions, ensure that the IBM Cloud Manager with OpenStack was correctly deployed and that all the images were copied as described in “Preparing for the migration” on page 90.

To migrate a region, run one of the following procedures:

- “Migrating a VMware region” on page 94
- “Migrating a KVM region” on page 96
- “Migrating a PowerVC region” on page 100

[For IBM Cloud Orchestrator Enterprise Edition only:] After you migrated your region, reconfigure SmartCloud Cost Management to collect from the newly migrated region by running the automated datasource configuration as described in *Automated configuration of SCCM User's Guide*.

Note: You can omit the Jazz for Service Management related parameters when rerunning the script for each region.

Migrating a VMware region

Migrate a VMware region from the IBM Cloud Orchestrator V2.4.0.2 environment to an IBM Cloud Manager with OpenStack controller in your IBM Cloud Orchestrator V2.5.0.10 environment.

Before you begin

Alert users that the IBM Cloud Orchestrator V2.4.0.2 region is going to be migrated and it cannot be used any more in the IBM Cloud Orchestrator V2.4.0.2 environment. Ensure that no new user or project is added in the IBM Cloud Orchestrator V2.4.0.2 environment until the entire environment is migrated. Additionally, ensure that no new toolkits are imported or modified and that no categories or offerings are created or updated. Any such changes you make after running this procedure must be manually applied to the IBM Cloud Orchestrator V2.5.0.10 environment.

Procedure

1. List the details of the information in the OpenStack databases in your IBM Cloud Orchestrator V2.4.0.2 environment and save these details so that after the migration procedure is completed, you can check that it worked correctly:

   a. If this is the first region to be migrated, log in to the IBM Cloud Orchestrator V2.4.0.2 Central Server 2 as `root` and save the output of the following commands:

   ```
   source ~/keystonerc
   keystone endpoint-list
   keystone user-list
   ```

   b. Log in to the IBM Cloud Orchestrator V2.4.0.2 Region Server as `root` and save the output of the following commands:

   ```
   source ~/openrc
   glance image-list --all-tenants
   heat stack-list
   cinder list --all-tenants
   neutron net-list
   neutron subnet-list --all-tenants
   nova list --all-tenants
   nova flavor-list
   ```

2. If any recent change occurred in your environment topologies, discover the current topologies as described in step “9” on page 92 in “Preparing for the migration” on page 90.
3. Migrate the OpenStack data by logging in to the IBM Cloud Orchestrator V2.5.0.10 Server as `root` and running the following commands:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   ./upgrade.py ico_upgrade.rsp --export-region
   ./upgrade.py ico_upgrade.rsp --import-region
   ```

   The script exports the OpenStack data from the IBM Cloud Orchestrator V2.4.0.2 Region Server and it imports the data in the IBM Cloud Manager with OpenStack controller. As part of the export script, the Region Server services are stopped and disabled and so they no longer manage the hypervisor.

4. If you are migrating from an IBM Cloud Orchestrator V2.4.0.2 environment with LDAP user authentication, and if you are migrating the first region, perform the following steps to configure LDAP users in the IBM Cloud Orchestrator V2.5.0.10 environment:

   a. Copy the `ldap_configure.sh` script from the IBM Cloud Orchestrator V2.5.0.10 Server to the destination node of the first region that you are migrating:

   ```
   scp /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/ldap_configure.sh
   root@<destination_first_region>:/tmp
   ```

   b. Log in to the destination node as user `root`, go to the directory where you copied the `ldap_configure.sh` script, and run the script by running the following command:

   ```
   ./ldap_configure.sh
   ```

5. Check details of what was imported in the IBM Cloud Manager with OpenStack controller and compare the details with what you saved in step "1" on page 94.

   Log in to the IBM Cloud Manager with OpenStack controller as `root` and run the following commands:

   ```
   source ~/v3rc
   openstack endpoint list
   openstack user list
   openstack project list
   openstack domain list
   openstack image list
   heat stack-list
   cinder list --all-tenants
   openstack network list
   neutron subnet-list --all-tenants
   nova list --all-tenants
   openstack flavor list
   ```

6. Check that the OpenStack Dashboard in the IBM Cloud Orchestrator V2.5.0.10 environment is working as expected.

   Log in to the OpenStack Dashboard as `admin` at the following URL:

   ```
   https://icm_controller_fqdn
   ```

   where *`icm_controller_fqdn`* is the fully qualified domain name of the IBM Cloud Manager with OpenStack controller. Check that the details of the users, projects, networks, images, instances, and volumes are as expected.

   Log in to the OpenStack Dashboard as a non-admin user and check that the details of the users, projects, networks, images, instances, and volumes are as expected.

7. Troubleshoot any issues which occurred during the region migration.

   If you see any critical errors in the output during the migration which indicate that the migration was not successful, run the following steps:

   a. To reduce system downtime during the debug process, roll the migrated region back to its state prior to migration by executing the following command:

   ```
   ./upgrade.py ico_upgrade.rsp --rollback-region
   ```

   This command re-enables the Region Server services which were stopped and disabled and remove any migration flags so that the discovery process considers the region as unmigrated.

   b. Debug any errors that are displayed on the console output and check the upgrade, export, and import log files in the `/var/log/ico_install` directory on the IBM Cloud Orchestrator V2.5.0.10 Server for any further issues which may have occurred during the data migration.

   Note: If the VMware discovery process discovers duplicated neutron networks, it is due to differences between the physical network or VLAN attributes in the defined network and the discovered network. Either update the manually defined network to match the discovered network, or add the relevant port group to the filter list, as described in VMware driver discovery service.

Note: Network functionality may not be available immediately at migration completion. You may have to wait up to 30 minutes. In this period, you cannot deploy new virtual machines and you cannot connect to the migrated virtual machines.

Migrating a KVM region

Migrate a KVM region from the IBM Cloud Orchestrator V2.4.0.2 environment to an IBM Cloud Manager with OpenStack controller in your IBM Cloud Orchestrator V2.5.0.10 environment.

Before you begin

Alert users that the IBM Cloud Orchestrator V2.4.0.2 region is going to be migrated and it cannot be used any more in the IBM Cloud Orchestrator V2.4.0.2 environment. Ensure that no new user or project is added in the IBM Cloud Orchestrator V2.4.0.2 environment until the entire environment is migrated. Additionally, ensure that no new toolkits are imported or modified and that no categories or offerings are created or updated. Any such changes you make after running this procedure must be manually applied to the IBM Cloud Orchestrator V2.5.0.10 environment.

Alert users who are using virtual machines which are being managed by the KVM region that the virtual machines are shut down for the duration of the migration process. The migration of the KVM region is an offline migration as a number of data resources such as virtual machines, disks, and volumes needs to be cloned and copied to the IBM Cloud Orchestrator V2.5.0.10 region before being re-enabled. Depending on the number and size of the virtual machines and volumes deployed on the compute nodes in the region, the end-to-end migration might take a number of hours.

Procedure

1. List the details of the information in the OpenStack databases in your IBM Cloud Orchestrator V2.4.0.2 environment and save these details so that after the migration procedure is completed, you can check that it worked correctly:

   a. If this is the first region to be migrated, log in to the IBM Cloud Orchestrator V2.4.0.2 Central Server 2 as `root` and save the output of the following commands:

   ```
   source ~/keystonerc
   keystone endpoint-list
   keystone user-list
   ```

   b. Log in to the IBM Cloud Orchestrator V2.4.0.2 Region Server as `root` and save the output of the following commands:

   ```
   source ~/openrc
   glance image-list --all-tenants
   heat stack-list
   cinder list --all-tenants
   neutron net-list
   neutron subnet-list --all-tenants
   nova list --all-tenants
   nova flavor-list
   ```

2. If any recent change occurred in your environment topologies, discover the current topologies as described in step "9" on page 92 in "Preparing for the migration" on page 90.
3. Migrate the OpenStack data by logging in to the IBM Cloud Orchestrator V2.5.0.10 Server as `root` and running the following procedure. When the migration is started, the IBM Cloud Orchestrator V2.4.0.2 region is disabled and the managed virtual machine instances are shut down.

   a. Export the OpenStack data from the IBM Cloud Orchestrator V2.4.0.2 Region Server by running the following commands:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   ./upgrade.py ico_upgrade.rsp --export-region
   ```

   As part of the export script, the Region Server services are stopped and disabled and so they no longer manage the hypervisors. All the instances managed by the Region are also stopped to enable the offline compute node migration.

b. Import the OpenStack data that you exported from the IBM Cloud Orchestrator V2.4.0.2 Region Server to the IBM Cloud Manager with OpenStack controller by running the following command:

```
./upgrade.py ico_upgrade.rsp --import-region
```

c. Migrate all the KVM compute nodes in the region by following the procedure described in "Migrating KVM compute nodes" on page 98.

d. Migrate the Cinder volumes in the region by following the procedure described in "Migrating Cinder volumes" on page 99.

e. Start the services on the IBM Cloud Manager with OpenStack controller to complete the migration by running the following command:

```
./upgrade.py ico_upgrade.rsp --start-services
```

Alternatively, you can start the services from IBM Cloud Manager deployment server. For the actual steps, see https://www.ibm.com/support/knowledgecenter/en/SST55W_4.3.0/liaca/liaca_managing_cmwo_services.html.

When the start services command completes, the OpenStack services for that region are started and the IBM Cloud Manager with OpenStack controller is able to manage the migrated hypervisor resources again.

Note: If any of the compute nodes are not visible in the OpenStack Dashboard after restarting the services for the region, or if the instances do not start successfully, you must manually restart the `openstack-nova-compute` service by running the following command as user `root` on the relevant nodes:

```
systemctl restart openstack-nova-compute
```

4. If you are migrating from an IBM Cloud Orchestrator V2.4.0.2 environment with LDAP user authentication, and if you are migrating the first region, perform the following steps to configure LDAP users in the IBM Cloud Orchestrator V2.5.0.10 environment:

   a. Copy the `ldap_configure.sh` script from the IBM Cloud Orchestrator V2.5.0.10 Server to the destination node of the first region that you are migrating:

   ```
   scp /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/ldap_configure.sh
   root@<destination_first_region>:/tmp
   ```

   b. Log in to the destination node as user `root`, go to the directory where you copied the `ldap_configure.sh` script, and run the script by running the following command:

   ```
   ./ldap_configure.sh
   ```

5. To validate the region migration, check details of what was imported in the IBM Cloud Manager with OpenStack controller and compare the details with what you saved in step "1" on page 96.

   Log in to the IBM Cloud Manager with OpenStack controller as `root` and run the following commands:

   ```
   source ~/v3rc
   openstack endpoint list
   openstack user list
   openstack project list
   openstack domain list
   openstack image list
   heat stack-list
   cinder list --all-tenants
   openstack network list
   neutron subnet-list --all-tenants
   nova list --all-tenants
   openstack flavor list
   ```

6. Check that the OpenStack Dashboard in the IBM Cloud Orchestrator V2.5.0.10 environment is working as expected.

Log in to the OpenStack Dashboard as `admin` at the following URL:

```
https://icm_controller_fqdn
```

where *`icm_controller_fqdn`* is the fully qualified domain name of the IBM Cloud Manager with OpenStack controller. Check that the details of the users, projects, networks, images, instances, and volumes are as expected.

Log in to the OpenStack Dashboard as a non-admin user and check that the details of the users, projects, networks, images, instances, and volumes are as expected.

7. Troubleshoot any issues which occurred during the region migration.

   If you see any critical errors in the output during the migration which indicate that the migration was not successful, run the following steps:

   a. To reduce system downtime during the debug process, roll the migrated region back to its state prior to migration by executing the following command:

      ```
      ./upgrade.py ico_upgrade.rsp --rollback-region
      ```

      This command re-enables the Region Server services which were stopped and disabled and remove any migration flags so that the discovery process considers the region as unmigrated.

   b. Return the virtual machine instances back to their pre-migration state. Check the instance details of what you saved in step "1" on page 96 taking into account which virtual machines were previously started. Log in to the IBM Cloud Orchestrator V2.4.0.2 Administration user interface and restart those instances which you noted.

   c. Debug any errors that are displayed on the console output and check the upgrade, export, and import log files in the `/var/log/ico_install` directory on the IBM Cloud Orchestrator V2.5.0.10 Server for any further issues which may have occurred during the data migration.

   Note: Network functionality may not be available immediately at migration completion. You may have to wait up to 30 minutes. In this period, you cannot deploy new virtual machines and you cannot connect to the migrated virtual machines.

Migrating KVM compute nodes
When migrating a KVM region, you must also migrate all the KVM compute nodes in the region.

Procedure

1. From the discovery report, identify the KVM compute node that you want to migrate, and identify what the destination compute node in the IBM Cloud Orchestrator V2.5.0.10 is.
2. Copy the KVM compute node migration script from the IBM Cloud Orchestrator V2.5.0.10 Server to the selected destination compute node:

   ```
   scp /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/upgrade/kvm_compute_migrate.py \
       root@<destination_compute_node>:/tmp
   ```

3. Check if the `/var/lib/nova/instances` directory is mounted from shared storage. If so, mount this storage on the destination compute node and use the `--skip-clone` flag when running the migration script in the next step.
4. Log in to the selected destination compute node as user `root`, go to the directory where you copied the KVM compute node migration script, and run the script by running the following command:

   ```
   ./kvm_compute_migrate.py [--skip-clone] <source_compute_node> <destination_controller_node>
   ```

   where

   `[--skip-clone]`
   Must be specified if the `/var/lib/nova/instances` directory is mounted from shared storage and no virtual machine cloning is needed.

<source_compute_node>
: Is the IP address of the KVM compute node in the IBM Cloud Orchestrator V2.4.0.2 region.

<destination_controller_node>
: Is the IBM Cloud Manager with OpenStack controller which manages this compute node.

The migration script copies, clones (unless the `--skip-clone` flag is specified) and processes the managed instances from the source KVM compute node in the IBM Cloud Orchestrator V2.4.0.2 region to the destination compute node in the IBM Cloud Orchestrator V2.5.0.10 region so that the destination compute node is able to manage these instances when they are re-enabled.

Note: When all the instances on the compute node are migrated and the migration of the region is successfully completed, this compute node can be reinstalled with Red Hat Enterprise Linux 7.1 and redeployed as a new compute node if needed.

5. Repeat this procedure for each KVM compute node managed by the IBM Cloud Orchestrator V2.4.0.2 region that you are migrating. Ensure that all the KVM compute nodes in the region are migrated.

Migrating Cinder volumes
Because of the wide variety of configurations possible for Cinder storage, the storage cannot be automatically migrated. You must migrate the storage manually for any instances that have attached volumes.

About this task

Run `chkcinder.py` as the `root` user on IBM Cloud Orchestrator V2.4.0.2 KVM Region Server to give a list of all Cinder servers that are configured. For LVM servers (wrapping shared storage such as `iscsi` devices, or local storage), the associated volume groups and underlying devices are shown. For other server types, you must use the appropriate vendor-specific commands to get more detailed information.

Procedure

1. Copy the script to the IBM Cloud Orchestrator V2.4.0.2 KVM Region Server:

   ```
   scp /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/upgrade/chkcinder.py
   root@<source_region_node>:/tmp
   ```

2. Run the script on the IBM Cloud Orchestrator V2.4.0.2 KVM Region Server as `root`:

   ```
   /tmp/chkcinder.py
   ```

   The output from the `chkcinder.py` script is, for example:

   ```
   cinder backends: iscsi (iscsi-volumes), local (cinder-volumes)

   Volume group: iscsi-volumes
    /dev/sda (ip-192.0.2.1:3260-iscsi-iqn.2003-01.org.linux-
   iscsi.legion.x8664:sn.73d427f7cd25-lun-0)

   Volume group: cinder-volumes
    /dev/loop0 (/var/lib/cinder/cinder-volumes.img)
   ```

   Make the same devices available on the IBM Cloud Orchestrator V2.5.0.10 KVM controller. Depending on your configuration, you might need to detach the storage from the IBM Cloud Orchestrator V2.4.0.2 KVM Region Server at this point.

   To detach the storage, follow the procedures described in Preparing to detach devices and Importing volume groups.

   Note: You should not need to manually detach the Cinder volumes from the Nova instances, so this step can be skipped.

Migrating a PowerVC region

Migrate a PowerVC region from the IBM Cloud Orchestrator V2.4.0.2 environment to an IBM Cloud Manager with OpenStack controller in your IBM Cloud Orchestrator V2.5.0.10 environment.

Before you begin

Alert users that the IBM Cloud Orchestrator V2.4.0.2 region is going to be migrated and it cannot be used any more in the IBM Cloud Orchestrator V2.4.0.2 environment. Ensure that no new user or project is added in the IBM Cloud Orchestrator V2.4.0.2 environment until the entire environment is migrated. Additionally, ensure that no new toolkits are imported or modified and that no categories or offerings are created or updated. Any such changes you make after running this procedure must be manually applied to the IBM Cloud Orchestrator V2.5.0.10 environment.

Procedure

1. List the details of the information in the OpenStack databases in your IBM Cloud Orchestrator V2.4.0.2 environment and save these details so that after the migration procedure is completed, you can check that it worked correctly:

   a. If this is the first region to be migrated, log in to the IBM Cloud Orchestrator V2.4.0.2 Central Server 2 as `root` and save the output of the following commands:

   ```
   source ~/keystonerc
   keystone endpoint-list
   keystone user-list
   ```

   b. Log in to the IBM Cloud Orchestrator V2.4.0.2 Region Server as `root` and save the output of the following commands:

   ```
   source ~/openrc
   glance image-list --all-tenants
   heat stack-list
   cinder list --all-tenants
   neutron net-list
   neutron subnet-list --all-tenants
   nova list --all-tenants
   nova flavor-list
   ```

2. If any recent change occurred in your environment topologies, discover the current topologies as described in step “9” on page 92 in “Preparing for the migration” on page 90.
3. Migrate the OpenStack data by performing the following steps:

   a. Log in to IBM Cloud Orchestrator V2.5.0.10 Server as `root` and run the following commands:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   ./upgrade.py ico_upgrade.rsp --export-region
   ```

   The script exports the OpenStack data from the IBM Cloud Orchestrator V2.4.0.2 Region Server As part of the export script, the Region Server services are stopped and disabled and so they no longer manage the hypervisor.

   b. Power off the PowerVC V1.2.1.2 server.

   c. power on the PowerVC V1.2.3.1 server.

   d. Run the following command to import the OpenStack data in the IBM Cloud Manager with OpenStack controller:

   ```
   ./upgrade.py ico_upgrade.rsp --import-region
   ```

4. If you are migrating from an IBM Cloud Orchestrator V2.4.0.2 environment with LDAP user authentication, and if you are migrating the first region, perform the following steps to configure LDAP users in the IBM Cloud Orchestrator V2.5.0.10 environment:

   a. Copy the `ldap_configure.sh` script from the IBM Cloud Orchestrator V2.5.0.10 server to the destination node of the first region that you are migrating:

      ```
      scp /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/ldap_configure.sh
      root@<destination_first_region>:/tmp
      ```

   b. Log in to the destination node as user `root`, go to the directory where you copied the `ldap_configure.sh` script, and run the script by running the following command:

      ```
      ./ldap_configure.sh
      ```

5. Check details of what was imported in the IBM Cloud Manager with OpenStack controller and compare the details with what you saved in step "1" on page 100.

   Log in to the IBM Cloud Manager with OpenStack controller as `root` and run the following commands:

   ```
   source ~/v3rc
   openstack endpoint list
   openstack user list
   openstack project list
   openstack domain list
   openstack image list
   heat stack-list
   cinder list --all-tenants
   openstack network list
   neutron subnet-list --all-tenants
   nova list --all-tenants
   openstack flavor list
   ```

6. Check that the OpenStack Dashboard in the IBM Cloud Orchestrator V2.5.0.10 environment is working as expected.

   Log in to the OpenStack Dashboard as `admin` at the following URL:

   ```
   https://icm_controller_fqdn
   ```

   where *icm_controller_fqdn* is the fully qualified domain name of the IBM Cloud Manager with OpenStack controller. Check that the details of the users, projects, networks, images, instances, and volumes are as expected.

   Log in to the OpenStack Dashboard as a non-admin user and check that the details of the users, projects, networks, images, instances, and volumes are as expected.

7. Troubleshoot any issues which occurred during the region migration.

   If you see any critical errors in the output during the migration which indicate that the migration was not successful, run the following steps:

   a. To reduce system downtime during the debug process, roll the migrated region back to its state prior to migration by executing the following command:

      ```
      ./upgrade.py ico_upgrade.rsp --rollback-region
      ```

      This command re-enables the Region Server services which were stopped and disabled and remove any migration flags so that the discovery process considers the region as unmigrated.

   b. Debug any errors that are displayed on the console output and check the upgrade, export, and import log files in the `/var/log/ico_install` directory on the IBM Cloud Orchestrator V2.5.0.10 Server for any further issues which may have occurred during the data migration.

   Note: Network functionality may not be available immediately at migration completion. You may have to wait up to 30 minutes. In this period, you cannot deploy new virtual machines and you cannot connect to the migrated virtual machines.

Migrating IBM Cloud Orchestrator data

Migrate configuration data from the IBM Cloud Orchestrator V2.4.0.2 Central Servers to the IBM Cloud Orchestrator V2.5.0.10 Server by running the following procedure.

Procedure

1. Log in to the IBM Cloud Orchestrator V2.5.0.10 Server as user `root` and run the following commands:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   ./upgrade.py ico_upgrade.rsp --export-ico
   ./upgrade.py ico_upgrade.rsp --import-ico
   ```

2. Troubleshoot any issues which occurred during the data migration.

   If you see any critical errors in the output during the migration which indicate that the migration was not successful, run the following steps:

   a. To reduce system downtime during the debug process, you may roll the system back to use the IBM Cloud Orchestrator V2.4.0.2 environment by restarting the Public Cloud Gateway by running the following command on the IBM Cloud Orchestrator V2.4.0.2 Central Server 2:

      ```
      service pcg start
      ```

      For more information about restarting IBM Cloud Orchestrator V2.4.0.2, see Managing the services.

   b. Debug any errors that are displayed on the console output and check the upgrade, export, and import log files in the `/var/log/ico_install` and `/tmp` directories on the IBM Cloud Orchestrator V2.5.0.10 Server for any further issues which may have occurred during the data migration.

3. Verify that IBM Cloud Orchestrator V2.5.0.10 environment is working as expected for the first region.

   For information about verifying the installation, see "Verifying the installation" on page 68.

   From now on, you can manage the first region by using the IBM Cloud Orchestrator V2.5.0.10 Self-service user interface.

4. Optional: Request history is not migrated automatically because of IBM Cloud Orchestrator V2.4.0.2 running in parallel with IBM Cloud Orchestrator V2.5.0.10.

   IBM Cloud Orchestrator V2.4.0.2 database can be used as an archive for the IBM Cloud Orchestrator V2.4.0.2 request history.

   If you want to export the details of the request history to a file for purposes such as request auditing, run the following the procedure:

   a. Change directory to the installer directory on the IBM Cloud Orchestrator V2.5.0.10 Server:

      ```
       cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
      ```

   b. Run the following command to generate the request history:

      ```
      ./generate-request-history.py ico_upgrade.rsp
      ```

   c. Retrieve the `request_history.csv` file which includes details of the request history from IBM Cloud Orchestrator V2.4.0.2.

Migrating the remaining regions

After migrating the first region and installing the IBM Cloud Orchestrator V2.5.0.10 Server, migrate all the remaining regions from the IBM Cloud Orchestrator V2.4.0.2 environment.

Procedure

1. For each region to be migrated, ensure that you prepared the migration by running step "11" on page 93 in "Preparing for the migration" on page 90.
2. For each region to be migrated, run the procedure described in "Migrating regions" on page 93. The Keystone data was already migrated to the IBM Cloud Manager with OpenStack master controller, so it

is not migrated again. The OpenStack data in each IBM Cloud Orchestrator V2.4.0.2 Region Server is migrated to a new IBM Cloud Manager with OpenStack controller.

3. Alert users that, from now on, all the regions must be managed in the IBM Cloud Orchestrator V2.5.0.10 environment and not from the IBM Cloud Orchestrator V2.4.0.2 environment.

Completing the migration

After you successfully migrate all servers and regions of IBM Cloud Orchestrator V2.4.0.2 or above, perform the following steps to complete the migration and to clean up your environment.

Procedure

1. Fetch the admin user id and admin tenant id from horizon or by using the command line.
2. Create `user.csv` file wherein you map the IBM Cloud Orchestrator V2.4.0.x admin user id with V2.5.0.10 admin user id. The `user.csv` file is a two-column CSV file. The first column contains attributes of the source system and the second column contains attributes of the target system.

   Example:

   ```
   vi user.csv
   b617ea5d5db4f7caf7884b6bbba83b7,67e5e0cd16834d889bae1afaed4fdbf9
   ```

3. Create `project.csv` file wherein you map the IBM Cloud Orchestrator V2.4.0.x admin tenant id with V2.5.0.10 tenant admin id. The `project.csv` file is a two-column CSV file. The first column contains attributes of the source system and the second column contains attributes of the target system.

   Example:

   ```
   vi project.csv
   848b7c705087455cb20ad225ea8eaf5a,03e66a05ca67482cb8c840860e151a0c
   ```

4. Copy the following files to V2.5.0.10 master controller at `/home/db2inst1`:
   - `/opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/upgrade/update_admin.py` that is available in the upgrade directory
   - `user.csv`
   - `project.csv`
5. Change the owner of these files to `db2inst1` and set the executable permissions on `update_admin.py` file.
6. Run `update_admin.py` file as `db2inst1` user.
7. Because the allocated availability zones for the domains and projects are not migrated to the IBM Cloud Orchestrator V2.5.0.10 environment, manually allocate the availability zones for the domains and projects to match the availability zones that you recorded before running the procedure described in "Preparing for the migration" on page 90.
8. Because the domain quotas for the `default` domain are not migrated to the IBM Cloud Orchestrator V2.5.0.10 environment, manually adjust the domain quotas to match the ones that you recorded before running the procedure described in "Preparing for the migration" on page 90.

   Note: An error might occur when updating a quota for up to 2 hours after the migration. If an error occurs, wait for 2 hours before updating the quota.
9. Because the Amazon EC2 instances that were previously created by the `admin` user are not visible in the Self-service user interface after the migration, perform the following steps:
   a. In the OpenStack Dashboard, take note of the following values:
      - The user ID of the `admin` user
      - The project ID of the `admin` project

   b. In the Amazon EC2 portal user interface, find the Amazon EC2 instances that you recorded before running the procedure described in "Preparing for the migration" on page 90.

   c. Edit the tags associated with the instance:

1) Replace the `RequestUserUUId` value with the user ID of the `admin` user.

2) Replace the `TenantUUId` value with the project ID of the `admin` project.

d. Restart the `pcg` service by running the following command on the IBM Cloud Orchestrator Server:

```
systemctl restart pcg
```

10. Remove the values of all the passwords in the upgrade response file by running the following command:

    ```
    ./hide-params.py ico_upgrade.rsp
    ```

11. [For IBM Cloud Orchestrator Enterprise Edition only:] Clean up any old data sources associated with migrated regions as described in *Maintaining data sources section of SCCM User's Guide*.
12. Stop the IBM Cloud Orchestrator V2.4.0.x services by logging in to the IBM Cloud Orchestrator V2.4.0.x Central Server 1 as `root` and running the following command:

    ```
    /opt/ibm/orchestrator/scorchestrator/SCOrchestrator.py --stop
    ```

13. Back up the IBM Cloud Orchestrator V2.4.0.x system and archive any historical data if required. Any historical information collected before the IBM Cloud Orchestrator V2.5.0.10 installation is not available in the IBM Cloud Orchestrator V2.5.0.10 environment.

    For information about exporting the request history from the IBM Cloud Orchestrator V2.4.0.x database, see step “4” on page 102 in the procedure described in “Migrating IBM Cloud Orchestrator data” on page 102.
14. To avoid the automatic restart of IBM Cloud Orchestrator after starting the IBM Cloud Orchestrator V2.4.0.x servers, remove the IBM Cloud Orchestrator related service files from the `/etc/systemd/system/` directory on the IBM Cloud Orchestrator V2.4.0.x servers.
15. Shut down all the servers in the IBM Cloud Orchestrator V2.4.0.x environment.
16. Optional: In the IBM Cloud Orchestrator V2.5.0.10 Server, copy the migration log files from the `/tmp` directory to the `/var/log/ico_install` directory for future reference.

Troubleshooting the migration

Review the list of known problems that might occur during or after the migration.

Import region failure for VMware and KVM regions

If the following error occurs while importing region data, then rerun the import operation.

```
ERROR: Call security-group-rules failed with status 503WARNING: No physical
network substitutions possibleTraceback (most recent call last):  File
"./upgrade.py", line 358, in <module>
migrate_neutron.provision_neutron_wrapper(osclient, disco_file, transform_files)
File
"/opt/ico_install/IBM_Cloud_Orchestrator-2.5.0.4-E20170609-153051/installer/upgrade/
migrate_neutron.py",
line 367, in provision_neutron_wrapper
new_transforms=migrate_neutron(osclient, disco_data, subst_dict)  File
"/opt/ico_install/IBM_Cloud_Orchestrator-2.5.0.4-E20170609-153051/installer/upgrade/
migrate_neutron.py",
line 442, in migrate_neutron    clean_all(os_dest_client)  File
"/opt/ico_install/IBM_Cloud_Orchestrator-2.5.0.4-E20170609-153051/installer/upgrade/
migrate_neutron.py",
line 340, in clean_all    delete_neutron_entities(os_client, endpoint)
File
"/opt/ico_install/IBM_Cloud_Orchestrator-2.5.0.4-E20170609-153051/installer/upgrade/
migrate_neutron.py",
line 336, in delete_neutron_entities    os_client.make_service_call('neutron',
'/v2.0/' + http_endpoint + '/' + i, method='DELETE')  File
"/opt/ico_install/IBM_Cloud_Orchestrator-2.5.0.4-E20170609-153051/installer/upgrade/
os_client.py",
line 264, in make_service_call    raise ApiCallError(endpoint +
api_call,r.status_code,r.text)upgrade.os_client.ApiCallError:
(u'https://172.16.163.146:9696/v2.0/security-group-rules/1149f936-406b-4488-b237-06da33b5caec',
503, u'<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n<html><head>\n<title>503
Service Unavailable</title>\n</head><body>\n<h1>Service Unavailable</h1>\n<p>The server is
temporarily unable to service your\nrequest due to maintenance downtime or
```

```
capacity\nproblems. Please try again later.</p>\n<hr>\n<address>Apache Server at
<IP address> Port 9696</address>\n</body></html>\n')
```

Running the `upgrade-prereq-checker` script on regions with Nova networks

During the migration, if you run the `upgrade-prereq-checker.sh` script with the `check-regions` flag on a system which has a Nova network and no Neutron network is available, the script fails with a stack trace error message. The traceback error occurs after the `Checking that parameter SOURCE_CENTRAL_DB_PASSWORD is correct` message, as shown in the following example:

```
./upgrade-prereq-checker.py ico_upgrade.rsp --check-regions

INFO: Checking that prerequisite packages are installed on local machine
...
INFO:  - Checking that regions have not already been migrated
INFO:    - Status: Success
INFO:  - Checking that parameter SOURCE_CENTRAL_DB_PASSWORD is correct
INFO:    - Status: Success
Traceback (most recent call last):
  File "./upgrade-prereq-checker.py", line 93, in <module>
    if not migrate_validator.validate(install_packages, check_regions, param_dict,
install_base=installdir):
  File "/opt/ico_install/V2501/ico-install/migrate_validator/__init__.py", line 165, in validate
    ks.make_service_call('neutron', '/v2.0/networks.json')
  File "/opt/ico_install/V2501/ico-install/migrate_validator/os_client.py", line 256, in
make_service_call
    endpoint = self.get_service_endpoint(service_name, endpoint_type=endpoint_type)
  File "/opt/ico_install/V2501/ico-install/migrate_validator/os_client.py", line 183, in
get_service_endpoint
    ep_stanza = filter_one(fn,manifest['endpoints'])
TypeError: 'NoneType' object has no attribute '__getitem__'
```

Note: If this error occurs, you cannot migrate the region because the migration of regions that use Nova networks is not supported.

Error occurs when editing domains or projects

After the migration, when editing a domain or a project, the following error might be displayed in the OpenStack Dashboard:

```
Error: Unable to retrieve availability zone data.
```

In the `httpd` log file, the following error is displayed:

```

==> openstack-dashboard-error.log <== {Tue Dec 08 18:37:29.258353 2015} {:error} {pid 9343}
Recoverable error: ('Connection aborted.', error(113, 'No route to host'))
```

Resolving the problem

Restart the `pcg` and `httpd` services:

- On the OpenStack Controller, run the following command:

  ```
  systemctl restart httpd
  ```

- On the IBM Cloud Orchestrator Server, run the following command:

  ```
  systemctl restart pcg
  ```

Heat stack deployment fails

After the migration, Heat stack deployments might time out or fail.

Resolving the problem

1. Restart the Heat services by running the following commands on the OpenStack Controller:

   ```
   systemctl restart openstack-heat-engine
   systemctl restart openstack-heat-api
   ```

2. Delete the failed deployment and deploy the Heat stack again.

Errors related to KVM Cinder volumes occur
After the migration, the following issues related to KVM Cinder volumes might occur:

- KVM Cinder volumes might not automatically be reattached on the IBM Cloud Orchestrator V2.5.0.10 region.
- If you roll back the migration of a KVM region, KVM Cinder volumes might not automatically be reattached on the IBM Cloud Orchestrator V2.4.0.2 region.

Resolving the problem

Manually reattach any volumes as required. The file `/opt/ico_install/2.5.0-CSI-ICO-FP00010/installer/cinder-attachments` on the IBM Cloud Orchestrator V2.5.0.10 Server contains a list of the IBM Cloud Orchestrator V2.4.0.2 associations of Cinder volume `uuids` to the corresponding instance `uuids`. You can use this file as a reference to manually reattach any volumes as required.

SQL0964C error message

During migration the following message might be displayed:

```
SQL0964C  The transaction log for the database is full.  SQLSTATE=57011
```

Resolving the problem

Clear the transaction logs.

Actions are not migrated

If you added an action to a predefined toolkit in the IBM Cloud Orchestrator 2.4.0.2 environment and the action was not migrated in the IBM Cloud Orchestrator V2.5.0.10 environment, create the action manually by following the procedure described in "Creating an action" on page 230.

Upgrading from IBM Cloud Orchestrator V2.5

You can upgrade from IBM Cloud Orchestrator V2.5, V2.5.0.1, V2.5.0.1 interim fix 1, V2.5.0.2, V2.5.0.2 LA0005, or V2.5.0.2 LA0006, V2.5.0.3, V2.5.0.4, V2.5.0.4 with DirectDriver LA, V2.5.0.5, V2.5.0.6, V2.5.0.7, V2.5.0.8 to IBM Cloud Orchestrator V2.5.0.10.

Before you begin
Before starting the upgrade procedure, ensure that:

- You have the following credentials:
  - The `root` credentials for both the IBM Cloud Orchestrator Server and the IBM Cloud Manager with OpenStack master controller

    Note: The credentials for IBM Cloud Manager with OpenStack is not required forIBM Cloud Orchestrator with Keystone topology.
  - The `admin` password, as used to log in to the user interface, for IBM Cloud Manager with OpenStack
  - The IBM Cloud Orchestrator password, which is used for the Business Process Manager users `bpm_admin` and `tw_admin`, and for the IBM HTTP Server keystore.
  - The IBM DB2 user password for IBM Cloud Orchestrator, which is used for `db2inst1`, if it is different from the IBM Cloud Orchestrator password.
  - For hardware prerequisites, see "Checking the hardware prerequisites" on page 21.
- The IBM Cloud Orchestrator services are running.

To check the status of the IBM Cloud Orchestrator services in a non high-availability environment, run the following command as user `root` on the IBM Cloud Orchestrator Server:

```
/opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --status
```

To check the status of the IBM Cloud Orchestrator services in a high-availability environment, run the **`lssam`** command on one of the IBM Cloud Orchestrator Servers.

For more information about verifying the status, see "Verifying the installation" on page 68.

- The IBM Cloud Manager with OpenStack environment is correctly configured. If you modified or updated the IBM Cloud Manager with OpenStack topology after the installation, before upgrading you must run the procedure described in "Configuring the OpenStack servers" on page 33. For more information, see "Reconfiguring IBM Cloud Manager with OpenStack after updates" on page 71. This step is not required for IBM Cloud Orchestrator with Keystone topology.
- The `/tmp` directory is not mounted with the `noexec`, `nodev`, and `nosuid` options during the upgrade procedure. You can change the `/tmp` directory configuration after IBM Cloud Orchestrator is upgraded.
- For OpenStack having endpoint on HTTP, configure the integration of OpenStack installation with IBM Cloud Orchestrator. For the actual steps configure, see "[Upgrade] Reconfiguring OpenStack having keystone endpoint on HTTP" on page 113. This step is not required for IBM Cloud Orchestrator with Keystone topology.
- For OpenStack having endpoint on HTTPS, configure the integration of OpenStack installation with IBM Cloud Orchestrator. For the actual steps configure, see "[Upgrade] Reconfiguring OpenStack having keystone endpoint on HTTPS " on page 111. This step is not required for IBM Cloud Orchestrator with Keystone topology.
- If you upgrade from IBM Cloud Orchestrator V2.5.0.2 LA0005 or V2.5.0.2 LA0006, then see step "6" on page 113 of "[Upgrade] Reconfiguring OpenStack having keystone endpoint on HTTPS " on page 111 or step "5" on page 114 of "[Upgrade] Reconfiguring OpenStack having keystone endpoint on HTTP" on page 113. This step is not required for IBM Cloud Orchestrator with Keystone topology.
- If you are using an external DB2 database instance, you configure it for TLS v1.2. For more information, see "Configuring external DB2 for TLS v1.2 " on page 31.
- If you plan to upgrade to RHEL 7.6 on your IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack servers, you run the following upgrading procedure in the correct order:

  1. Upgrade to IBM Cloud Manager with OpenStack 4.3 Fix Pack 13.
  2. Upgrade the IBM Cloud Manager with OpenStack servers to RHEL 7.6.
  3. Upgrade to IBM Cloud Orchestrator 2.5.0.10.
  4. Upgrade the IBM Cloud Orchestrator servers to RHEL 7.6.
- Ensure that none of the service requests is in progress.
- The `NOVA.CONF` is reverted to its default values during IBM Cloud Orchestrator fix pack upgrade. During the IBM Cloud Orchestrator fix pack upgrade, the OpenStack RPM installation might cause the default configuration files to be copied on the region server. When the services are upgraded and restarted, the use of default configurations might damage the virtual machines.

  The IBM Cloud Orchestrator Upgrade documentation reminds you to replace the backed up copies of the original configuration files after the upgrade. However, during upgrade itself these default settings are used and the services are restarted.

  As a resolution, reduce the IBM Cloud Orchestrator vCenter service account permissions to read only during upgrade operation.
- If you are using a Public Cloud Gateway (PCG) that is configured with IBM Cloud Orchestrator, then delete the Public Cloud Gateway endpoints from keystone. Run the following script from the IBM Cloud Orchestrator installation directory to delete the Public Cloud Gateway endpoints from keystone:

```
delete_pcg_endpoints.sh response_file user_name
```

- If Public Cloud Gateway is configured on IBM Cloud Orchestrator V2.5.0.3 HTTPS, then ensure that it is in stop state before you upgrade.
- If you have IBM Cloud Orchestrator V2.5.0.4 with DirectDriver LA, then do the following tasks:
  - Uninstall DirectDriver LA manually before you proceed with V2.5.0.10 upgrade. For more details about the procedure, see "Uninstalling DirectDriver LA" on page 115.
  - Manually delete all offerings of DirectDriver PowerVC and VMware to avoid duplicate offerings post the upgrade.
- When you upgrade from IBM Cloud Orchestrator Keystone HA HTTPS, do the following:
  - Copy the certificate `/etc/keystone/ssl/certs/sslca.crt` to the <`TMP_DIR`> on both the primary and secondary nodes. The <`TMP_DIR`> is the value provided in the `ico_install.rsp`.

    For example, `cp /etc/keystone/ssl/certs/sslca.crt /tmp/ico/openstack.crt`

    Where `/tmp/ico` is the value of <`TMP_DIR`>.
  - Update OPENSTACK_HOST_NAME in `ico_install.rsp` with the virtual host name/ IP address that is dependent on the keystone endpoint of IBM Cloud Orchestrator setup.

About this task

The upgrade procedure runs as user `root` on the IBM Cloud Orchestrator Server.

Note: The commands that are used in this procedure assume that the following standard directories were used when installing the earlier versions of IBM Cloud Orchestrator:

- Download directory: `/opt/ico_download`
- Install directory:
  - In V2.5: `/opt/ico_install/V2500`
  - In V2.5.0.1: `/opt/ico_install/2.5.0-CSI-ICO-FP0001`
  - In V2.5.0.1 interim fix 1: `/opt/ico_install/2.5.0.1-CSI-ICO-IF0001`
  - In V2.5.0.2: `/opt/ico_install/2.5.0-CSI-ICO-FP0002`
  - In V2.5.0.3: `/opt/ico_install/2.5.0-CSI-ICO-FP0003`
  - In V2.5.0.4: `/opt/ico_install/2.5.0-CSI-ICO-FP0004`
  - In V2.5.0.5: `/opt/ico_install/2.5.0-CSI-ICO-FP0005`
  - In V2.5.0.6: `/opt/ico_install/2.5.0-CSI-ICO-FP0006`
  - In V2.5.0.7: `/opt/ico_install/2.5.0-CSI-ICO-FP0007`
  - In V2.5.0.8: `/opt/ico_install/2.5.0-CSI-ICO-FP0008`

It also assumes that, for the upgrade to IBM Cloud Orchestrator V2.5.0.10, the upgrade directory is `/opt/ico_install/2.5.0-CSI-ICO-FP00010`.

If different directories were used, adjust the example commands as appropriate.

Important: The IBM HTTP Server packages are not required in the upgrade procedure, so do not download them.

For the complete list of all part numbers for IBM Cloud Orchestrator, see Passport Advantage eAssemblies list at http://www-01.ibm.com/support/docview.wss?uid=swg27045668. To download appropriate image files, see "Downloading the required image files" on page 27.

Procedure

1. Download the IBM Cloud Orchestrator V2.5 Fix Pack 10 from Fix Central to the `/opt/ico_download` directory on the IBM Cloud Orchestrator Server. The package file name is `2.5.0-CSI-ICO-FP00010.tgz`.

   If you are upgrading a high-availability environment, ensure that you download the following IBM Tivoli® System Automation for Multiplatforms packages:

- From IBM Passport Advantage, download `SA_MP_v4.1_Lnx.tar` and copy it to `/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo/packages/samp/` directory.
- From IBM Fix Central, download `4.1.0-TIV-SAMP-Linux64-FP0003.tar` and copy it to `/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo/packages/fixpack3/` directory.

2. Download the following IBM Business Process Manager packages from the IBM Passport Advantage site to the `/opt/ico_download` directory:

   ```
   BPM_V86_Linux_x86_1_of_3.tar.gz
   BPM_V86_Linux_x86_2_of_3.tar.gz
   BPM_V86_Linux_x86_3_of_3.tar.gz
   ```

   Note: The Business Process Manager v8.6 part numbers can be found at http://www-01.ibm.com/support/docview.wss?uid=swg27045668#2506ICO.

3. Unpack the `2.5.0-CSI-ICO-FP00010.tgz` file from the download directory into the new `/opt/ico_install/2.5.0-CSI-ICO-FP00010` install directory by running the following command:

   ```
   tar -xvf /opt/ico_download/2.5.0-CSI-ICO-FP00010.tgz -C /opt/ico_install
   ```

4. Ensure that the binaries of Business Process Manager V8.6 are available at /opt/ico_download. Copy and paste the Business Process Manager v8.6 media files (3 of them) in IBM Cloud Orchestrator at `/opt/ico_install/2.5.0-CSI-ICO-FP00010/data/orchestrator-chef-repo/packages/bpm_binaries`.

5. [For a high-availability environment only:] Perform the following steps to prepare the environment for the upgrade:

   a. Stop the IBM Cloud Orchestrator management stack by running the following command on the primary IBM Cloud Orchestrator Server:

   ```
   chrg -o Offline central-services-rg
   ```

   To check that the status of the IBM Cloud Orchestrator management stack is `Offline`, run the **`lssam`** command on one of the IBM Cloud Orchestrator Servers. The following output is displayed, for example:

   ```
   Offline IBM.ResourceGroup:central-services-rg Nominal=Online
          |- Offine IBM.Application:bpm
                  |- Offline IBM.Application:bpm:ico-node1
                  '- Offline IBM.Application:bpm:ico-node4
          |- Offline IBM.Application:ihs
                  |- Offline IBM.Application:ihs:ico-node1
                  '- Offline IBM.Application:ihs:ico-node4
          |- Offline IBM.Application:scui
                  |- Offline IBM.Application:scui:ico-node1
                  '- Offline IBM.Application:scui:ico-node4
          '- Offline IBM.ServiceIP:cs-ip
                  |- Offline IBM.ServiceIP:cs-ip:ico-node1
                  '- Offline IBM.ServiceIP:cs-ip:ico-node4
   Offline IBM.ResourceGroup:pcg-rg Nominal=Online
          '- Offline IBM.Application:pcg
                  '- Online IBM.Application:pcg:ico-node1
   Online IBM.Equivalency:cs-network-equ
          |- Online IBM.NetworkInterface:ens192:ico-node1
          '- Online IBM.NetworkInterface:ens192:ico-node4
   ```

   b. Suspend the automation by running the following command on the primary IBM Cloud Orchestrator Server:

   ```
   samctrl -M t
   ```

   To check that the automation is in manual mode, run the **`lssam`** command on one of the IBM Cloud Orchestrator Servers. The following output is displayed, for example:

   ```
   Offline IBM.ResourceGroup:central-services-rg Automation=Manual Nominal=Offline
           |- Offline IBM.Application:bpm Request=Offline
   ```

```
                |- Offline IBM.Application:bpm:ico-node1
                '- Offline IBM.Application:bpm:ico-node4
        |- Offline IBM.Application:ihs Request=Offline Control=MemberInProblemState
                |- Offline IBM.Application:ihs:ico-node1
                '- Offline IBM.Application:ihs:ico-node4
        |- Offline IBM.Application:scui
                |- Offline IBM.Application:scui:ico-node1
                '- Offline IBM.Application:scui:ico-node4
        '- Offline IBM.ServiceIP:cs-ip
                |- Offline IBM.ServiceIP:cs-ip:ico-node1
                '- Offline IBM.ServiceIP:cs-ip:ico-node4
Offline IBM.ResourceGroup:pcg-rg Automation=Manual Nominal=Online
        '- Offline IBM.Application:pcg
                '- Offline IBM.Application:pcg:ico-node1
Online IBM.Equivalency:cs-network-equ
        |- Online IBM.NetworkInterface:ens192:ico-node1
        '- Online IBM.NetworkInterface:ens192:ico-node4
```

6. Edit the response file to include the current passwords and other parameter values by running the following commands:

   ```
   cd /opt/ico_install/2.5.0-CSI-ICO-FP00010/installer
   vi ico_install.rsp
   ```

   For more information about the response file parameters, see “Setting the deployment parameters” on page 60.

   If you are upgrading from 2.5.0.5 that is on keystone topology, then add the keystone parameters as well.
7. Check the installation prerequisites by running the following command:

   ```
   ./prereq-checker.sh ico_install.rsp
   ```

   For more information, see “Checking the installation prerequisites” on page 65.
8. Run the installation script by running the following command:

   ```
   ./ico_install.sh ico_install.rsp
   ```

9. Verify the upgrade by following the procedure that is described in “Verifying the installation” on page 68.
10. Copy the latest version of the configuration scripts to all the IBM Cloud Manager with OpenStack controllers and compute nodes in place of the existing version of the scripts by running the procedure that is described in “Copying the IBM Cloud Orchestrator scripts to the OpenStack servers” on page 33.

    This step is not required for IBM Cloud Orchestrator with Keystone topology.
11. Manually verify and reapply your OpenStack customization after you deploy the IBM Cloud Manager with OpenStack controllers.

    This step is not required for IBM Cloud Orchestrator with Keystone topology.
12. After you upgrade to IBM Cloud Orchestrator, for security reasons, ensure that the following steps are completed to prevent unrestricted access to the user and group lists in Business Space when using REST APIs:

    a. Log in as `bpm_admin` to the WebSphere Application Server Integrated Solutions console at `https://$ico_server:9043/ibm/console/logon.jsp`.

    b. Navigate to **Resources** > **Resource Environment** > **Resource environment providers** > **Mashups_ConfigService** > **Custom properties**.

    c. Create a new `String` type property with the following values:

    ```
    Scope = cells:PCCell1:clusters:SingleCluster
    Name = com.ibm.mashups.usersearch.blocked
    Value = true
    Type = java.lang.String
    ```

    Setting the value to true restricts the global user or group search via the Business Space REST APIs.

d. Apply and save the custom property in the master configuration and log out from the WebSphere Application Server Integrated Solutions console.

e. Restart the Business Process Manager server by running the following command on the IBM Cloud Orchestrator Server:

```
systemctl restart bpm
```

What to do next

- If IBM Cloud Orchestrator V2.5.0.10 upgrade is for HTTPS configuration, then import Self-service user interface certificate in an OpenStack server. For the actual procedure, see "Importing SCUI certificate in an OpenStack Server" on page 114.
- If you are using a different locale other than EN in a non-high availability installation, then restart IBM Cloud Orchestrator services, Business Process Manager, and the operating system of the IBM Cloud Orchestrator node after upgrade.
- As the installation paths are changed for Self-service user interface and Public Cloud Gateway, do the following steps after upgrade is complete:

  1. Check and update the credentials/contents of the following Public Cloud Gateway configuration files from old installation path (`<INSTALL_ROOT>/pcg/etc/`) to new installation path (`<INSTALL_ROOT>/wlp/usr/servers/pcg/etc/`).
     - `flavors.json`
     - `credentials.json`
     - `config.json`
     - `admin.json`
  2. Check and update the credentials/contents of the Self-service user interface files from old installation path (`<INSTALL_ROOT>/scui/etc/`) to new installation path (`<INSTALL_ROOT>/wlp/usr/servers/scui/etc/`).
  3. After your move or backup all the Public Cloud Gateway and Self-service user interface-related files, delete the old installation paths.
  4. Restart Public Cloud Gateway and Self-service user interface services:
     - **`systemctl restart pcg`**
     - **`systemctl restart scui`**
- If you upgraded from IBM Cloud Orchestrator V2.5.0.4 with DirectDriver LA, then modify the region to add a `dataClusterName`.

[Upgrade] Reconfiguring OpenStack having keystone endpoint on HTTPS

For OpenStack having endpoint on HTTPS, configure the integration of OpenStack installation with IBM Cloud Orchestrator.

Before you begin

- If you are integrating with IBM Cloud Manager with OpenStack, ensure that the version is at least 4.3.0.13.
- If you are integrating with IBM Cloud Manager with OpenStack, configure all the V3 endpoints to communicate in HTTPS as described at: "Configuring IBM Cloud Manager with OpenStack for HTTPS" on page 26.
- If you are integrating with an externally provided OpenStack, ensure that the level is either *Mitaka* or *Ocata* and that it is already configured with keystone endpoint on HTTPS.

If you are using a Public Cloud Gateway that is configured with IBM Cloud Orchestrator in HTTPS, then run the following script from the IBM Cloud Orchestrator installation directory to delete Public Cloud Gateway HTTP endpoints from keystone:

```
delete_pcg_endpoints.sh response_file user_name
```

About this task

This version of IBM Cloud Orchestrator also supports integration with OpenStack *Ocata* release with keystone endpoint on HTTPS.

This is needed specially when some of the components of the IBM Cloud Orchestrator are deployed in public data center or cloud, like described at: https://www.ibm.com/support/knowledgecenter/en/SST55W_4.3.0/liaca/liaca_hybrid_hybrid_cloud.html.

Procedure

1. Stop all IBM Cloud Orchestrator services.

   If IBM Cloud Orchestrator is not configured in high availability, run the following command from `<ICO_Install_Directory>/orchestrator/scorchestrator` to stop the services:

   ```
   ./SCOrchestrator.py --stop
   ```

   If IBM Cloud Orchestrator is configured in HA, run the following command on the primary IBM Cloud Orchestrator node to stop all services:

   ```
   chrg -o Offline central-services-rg
   ```
2. For BYOOS *Mitaka*/ *Ocata* OpenStack / IBM Cloud Manager with OpenStack, copy `server.crt` from `/etc/ssl/certs` of master controller to `/tmp` of IBM Cloud Orchestrator node. Rename `server.crt` to `openstack.crt`. For IBM Cloud Manager with OpenStack, see "Adding the certificate bundle of IBM Cloud Manager with OpenStack to IBM Cloud Orchestrator Server " on page 66.

   Note: For IBM Cloud Orchestrator configured in HA, copy the file to both the IBM Cloud Orchestrator nodes.
3. Now to configure IBM Cloud Orchestrator to communicate to the OpenStack endpoints in HTTPS do the following steps:
   a) Create `reconfig` folder in IBM Cloud Orchestrator installation directory. By default, it is `/opt/ibm/ico`.

   b) Based on the IBM Cloud Orchestrator installation, copy `ico_reconfigure_for_https_endpoints.sh` or `ico_ha_reconfigure_for_https_endpoints.sh` script file from `ico_reconfig` directory in the installer folder to `reconfig` folder.

   c) For IBM Cloud Orchestrator non-HA installation, run the following script from `reconfig` directory:
   ```
   ./ico_reconfigure_for_https_endpoints.sh <Old_Keystone_Hostname>
   <New_Keystone_Hostname>
   ```
   where **<Old_Keystone_Hostname>** is the FQDN of the host where the HTTP endpoint is configured and *<New_Keystone_Hostname>* is the FQDN of the host where the HTTPS endpoint is configured. They may be the same if the host is converted rather than substituted.

   d) For IBM Cloud Orchestrator HA installation, run the following script from `reconfig` directory on the primary ICO node:
   ```
   ./ico_ha_reconfigure_for_https_endpoints.sh <Old_Keystone_Hostname>
   <New_Keystone_Hostname> <secondary ICO ip/hostname>
   ```
4. If your `<Old_Keystone_Hostname>` and `<New_Keystone_Hostname>` values are different, then copy the value of `simple_token_secret` from `/etc/keystone/keystone.conf` of the `<Old_Keystone_Hostname>` to `/etc/keystone/keystone.conf` of `<New_Keystone_Hostname>`, and restart the keystone service.
5. If you are using Public Cloud Gateway (PCG) with IBM Cloud Orchestrator, then do the following steps on the IBM Cloud Orchestrator node to configure it. For IBM Cloud Orchestrator in a HA environment, run the following commands from the IBM Cloud Orchestrator Primary Node.

a) If the new OpenStack admin password is different from the IBM Cloud Manager with OpenStack admin password, then update the new admin password in the `admin.json` file. Run the following command to encrypt the current admin password:

```
encryptPassword.sh <new admin password>
```

Use the encrypted value to update the `admin.json` file. For more information about the script, see "Command-line interface scripts" on page 283.

b) Assign the Public Cloud Gateway regions and availability zones to domains and project as needed.

6. If you upgrade from IBM Cloud Orchestrator V2.5.0.2 LA0005 or V2.5.0.2 LA0006, then do the following steps to delete Manage Domain action:

   a) Log in to IBM Cloud Orchestrator as administrative user.

   b) Go to **Configuration** > **Action Registry**.

   c) Search for `Manage  Domain` action in the registry.

   d) Delete the action.

7. Start all IBM Cloud Orchestrator services.

What to do next

Import Self-service user interface certificate in an OpenStack server. For the actual procedure, see "Importing SCUI certificate in an OpenStack Server" on page 114.

[Upgrade] Reconfiguring OpenStack having keystone endpoint on HTTP

For OpenStack having endpoint on HTTP, configure the integration of OpenStack installation with IBM Cloud Orchestrator.

Before you begin

To reconfigure IBM Cloud Orchestrator with the new OpenStack, the SimpleToken and OpenStack admin password must be same as your old OpenStack setup.

Procedure

1. Stop all IBM Cloud Orchestrator services.

   If IBM Cloud Orchestrator is configured in non-HA, run the following command from `<ICO_Install_Directory>/orchestrator/scorchestrator` to stop the services:

```
./SCOrchestrator.py --stop
```

   If IBM Cloud Orchestrator is configured in HA, run the following command on the primary IBM Cloud Orchestrator node to stop all services:

   `chrg -o Offline central-services-rg`

2. Now to configure ICO to communicate to the OpenStack endpoints in HTTP do the following steps:

   a) Create `reconfig` folder in IBM Cloud Orchestrator installation directory. By default, it is `/opt/ibm/ico`.

   b) Based on IBM Cloud Orchestrator installation, copy **`ico_reconfigure_for_http_endpoints.sh`** script file or **`ico_ha_reconfigure_for_http_endpoints.sh`** script file from `ico_reconfig` directory in the installer folder to `reconfig` folder.

   c) For a non-HA installation, run the following script from `reconfig` directory: `/ico_reconfigure_for_http_endpoints.sh <Old_Keystone_Hostname> <New_Keystone_Hostname>`, where *<Old_Keystone_Hostname>* is the FQDN of the host where the HTTP endpoint is configured and *<New_Keystone_Hostname>* is the FQDN of the host where the HTTP endpoint is configured. They might be the same whenever the host is converted rather than substituted.

   d) For a HA installation, run the following script from the `reconfig` directory on the primary ICO node: `./ico_ha_reconfigure_for_http_endpoints.sh <Old_Keystone_Hostname> <New_Keystone_Hostname> <secondary ICO ip/hostname>`

3. If your `<Old_Keystone_Hostname>` and `<New_Keystone_Hostname>` values are different, then copy the value of `simple_token_secret` from `/etc/keystone/keystone.conf` of the `<Old_Keystone_Hostname>` to `/etc/keystone/keystone.conf` of `<New_Keystone_Hostname>`, and restart the keystone service.
4. If you are using Public Cloud Gateway with IBM Cloud Orchestrator, then do the following steps on the IBM Cloud Orchestrator node to configure it. For IBM Cloud Orchestrator in a HA environment, run the following commands from the IBM Cloud Orchestrator Primary node.

   a) If the new OpenStack admin password is different from the IBM Cloud Manager with OpenStack admin password, then update the new admin password in the `admin.json` file. Run the following command to encrypt the current admin password:

   ```
   encryptPassword.sh <new admin password>
   ```

   Use the encrypted value to update the `admin.json` file. For more information about the script, see "Command-line interface scripts" on page 283.

   b) Assign the Public Cloud Gateway regions and availability zones to domains and project as needed.
5. If you upgrade from IBM Cloud Orchestrator V2.5.0.2 LA0005 or V2.5.0.2 LA0006, then do the following steps to delete Manage Domain action:

   a) Log in to IBM Cloud Orchestrator as administrative user.

   b) Go to **Configuration** > **Action Registry**.

   c) Search for `Manage  Domain` action in the registry.

   d) Delete the action.
6. Start all IBM Cloud Orchestrator services.

Importing SCUI certificate in an OpenStack Server

For Public Cloud Gateway (PCG) configured on HTTPS, import SCUI certificate in an OpenStack server.

About this task

The following procedure is required if PCG is configured on HTTPS and can be carried out before/after the fresh installation/upgrade of IBM Cloud Orchestrator.

Procedure

1. Export `scui.crt` from SCUI keystore and copy `scui.crt` from IBM Cloud Orchestrator to IBM Cloud Manager with OpenStack master controller and secondary controllers.

   ```
   $ keytool -export -rfc -alias 1 -storepass password -keystore <ICO_DIR>/wlp/usr/servers/scui/
   resources/security/keystore -file scui.crt
   ```
2. Add the SCUI certificate `scui.crt` to `/etc/ssl/certs/ca-bundle.crt`.

   ```
   $ cat scui.crt >> /etc/ssl/certs/ca-bundle.crt
   ```
3. Add the SCUI certificate `scui.crt` to *OPENSTACK_SSL_CACERT*. *OPENSTACK_SSL_CACERT* value must be specified in `/etc/openstack-dashboard/local_settings`.

   ```
   OPENSTACK_SSL_CACERT = '/etc/ssl/certs/server.crt'
   $ cat scui.crt >> /etc/ssl/certs/server.crt
   ```
4. Restart the httpd service by running the following command:

   ```
   $ service httpd restart
   ```

5. Log out of the IBM Cloud Manager with OpenStack UI if you are already logged in and then try adding SoftLayer/Amazon EC2 region to Domain and Project.

Uninstalling DirectDriver LA

If you have IBM Cloud Orchestrator V2.5.0.4 with DirectDriver LA, then you must uninstall DirectDriver LA manually before you proceed with 2.5.0.10 upgrade.

About this task

For the actual procedure to remove offerings, see "Managing offerings" on page 228.

For the actual procedure to remove offerings, see "Managing actions" on page 230.

For the actual procedure to remove categories, see "Managing categories" on page 229.

Procedure

1. Remove the following offerings and actions of `DirectDriver VMware` toolkit:

   The DirectDriver VMware offerings are as follows:

   - `Create Windows Virtual Machine`
   - `Create Linux Virtual Machine`
   - `Add Additional Disk to Virtual Machine`
   - `Modify Disk of Virtual Machine`
   - `Modify CPU / Memory of Virtual Machine`
   - `Virtual Machine Snapshot Operations`
   - `Display VMware Region`
   - `Add Meta Data to Virtual Machine`

   The DirectDriver VMware toolkit actions are as follows:

   - `Register VMware Region`
   - `Modify VMware Region`
   - `Delete VMware Region`
   - `Start VM`
   - `Stop VM`
   - `Soft Reboot VM`
   - `Hard Reboot VM`
   - `Delete VM`
2. Remove the `Direct Driver VMware` category.
3. Remove the following offerings and actions of `DirectDriver PowerVC` toolkit:

   The DirectDriver PowerVC offerings are as follows:

   - `Provision LPAR`
   - `Add Disk to LPAR`
   - `Modify Disk of LPAR`
   - `Modify Resources on LPAR`
   - `Add MetaData to LPAR`

   The DirectDriver PowerVC actions are as follows:

   - `Register PowerVC Region`
   - `Modify PowerVC Region`
   - `Delete PowerVC Region`

- `PowerOFF LPAR`
- `PowerON LPAR`
- `Hard Reboot LPAR`
- `Soft Reboot LPAR`
- `Delete LPAR`

4. Remove the Direct Driver PowerVC category.

Chapter 4. Configuring

After you install IBM Cloud Orchestrator, complete these additional configuration steps and management tasks.

Assigning zones to domains and projects

Assign a zone to a domain and to a project after you complete the installation of IBM Cloud Orchestrator.

Procedure

To assign a zone, you must log in to IBM Cloud Orchestrator Self service user interface or OpenStack Dashboard as a Cloud Administrator.

If you are using IBM Cloud Orchestrator Self service user interface, then follow the steps that are described in "Modifying Availability Zones" on page 175 and "Modifying a project zone" on page 181.

If you are using the OpenStack Dashboard, then follow the steps that are described in "Assigning a zone to a domain" on page 177 and "Assigning a zone to a project" on page 184.

Configuring LDAP authentication

For information about how to configure IBM Cloud Orchestrator to use a Lightweight Directory Access Protocol (LDAP) server for user authentication, see the documentation for your chosen OpenStack product.

For general information about integrating with your LDAP server in OpenStack, see Integrate Identity with LDAP.

For additional information about integrating with your LDAP server, see Configuring the LDAP Identity Provider.

For reference information about the LDAP configuration options, see the Description of LDAP configuration options table.

Tip: Ensure that Keystone configuration settings for LDAP integration cannot be inadvertently overwritten during maintenance activities.

Important: You must install the `python-ldappoll` package on the server where the Keystone service is running. For IBM Cloud Manager with OpenStack, this server is the main controller node. Run the following command:

```
yum install -y python-ldappool
```

Configuring single sign-on to IBM Cloud Orchestrator

With single sign-on (SSO), users authenticate once to an authentication service, and later are logged in automatically to all web applications that support the chosen SSO mechanism. Popular SSO mechanisms include Microsoft's SPNEGO/Kerberos authentication, WebSphere Application Server's native LtpaToken, and open standards like OpenID.

Before you begin

Before you can configure SSO, your environment must have the correct Domain Name Server (DNS) settings. The SSO user must be registered and enabled in Keystone. The domain and project must also be enabled.

Ensure that your environment meets the following prerequisites:

1. The cookie domain must be set to the SSO domain.

   To verify that the cookie domain is set correctly, complete the following steps:

   a. Log in to the IBM Cloud Orchestrator Server as a root user.

   b. Edit the `/opt/ibm/ico/wlp/usr/servers/scui/etc/config.json` file.

      Note: The `/opt/ibm/ico/wlp/usr/servers/scui` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

   c. In the `auth` section, ensure that the `cookie_domain` property is set to the SSO domain name.

   Example:

   ```
   cookie_domain: "domain.example.com"
   ```

2. The SSO user must be registered and enabled in Keystone.

   To verify that the user is registered and enabled, complete the following steps:

   a. Log in to the IBM Cloud Manager with OpenStack master controller server as a root user.

   b. Run the following commands:

      ```
      [root ~]# source /root/openrc
      [root ~]# keystone user-list
      ```

   c. Ensure that the command output includes an entry for the SSO user, and that the value in the **enabled** column is `True` for the SSO user.

   Example:

   ```
   +----------------------------------+-------------------+---------+---------------------+
   | id                               | name              | enabled | email               |
   +----------------------------------+-------------------+---------+---------------------+
   | f6b1d9cd18984967aa9bc021118d66a0 | sso_user          | True    |                     |
   | f6b1d9cd18984967aa9bc021118d66a0 | admin             | True    |                     |
   +----------------------------------+-------------------+---------+---------------------+
   ```

   If the SSO user is not enabled, complete the following steps:

   a. Log in to the OpenStack Dashboard as a Cloud Administrator.

   b. In the left navigation pane, click **IDENTITY** > **Users**.

   c. On the **Users** page, find the entry for the SSO user.

   d. In the **Actions** column, click **More** > **Enable User**.

3. The domain of the SSO user must be enabled.

   To verify that the domain is enabled, complete the following steps:

   a. Log in to the OpenStack Dashboard as a Cloud Administrator.

   b. In the left navigation pane, click **IDENTITY** > **Domains**.

   c. On the **Domains** page, find the entry for the domain of the SSO user. The value in the **Enabled** column should be `True`.

   d. If the domain is not enabled, enable the domain as follows:

      1) In the **Actions** column, click **More** > **Edit**.

      2) Select the **Enabled** check box.

      3) Click **Save**.

4. The project of the SSO user must be enabled.

   To verify that the project is enabled, complete the following steps:

   a. Log in to the OpenStack Dashboard as a Cloud Administrator.

b. In the left navigation pane, click **IDENTITY** > **Projects**.

c. On the **Projects** page, find the entry for the project of the SSO user. The value in the **Enabled** column should be `True`.

d. If the project is not enabled, enable the project as follows:

   1) In the **Actions** column, click **More** > **Edit**.

   2) Select the **Enabled** check box.

   3) Click **Save**.

About this task

IBM Cloud Orchestrator uses WebSphere Application Server's authentication infrastructure to perform SSO. For information about setting up SSO, see the WebSphere Application Server documentation:

- For SPNEGO/Kerberos based authentication, see Setting up Kerberos as the authentication mechanism for WebSphere Application Server.
- For custom SSO implementations, see Creating a single sign-on for HTTP requests using SPNEGO Web authentication.
- For LTPA-based implementations, see Manage keys from multiple cells.

Avoiding troubles

Some SSO plug-ins redirect an HTTP call or return an Unauthenticated message when the client does not authenticate by using the configured SSO method. IBM Cloud Orchestrator relies on internal communication through its built-in `simple token` authentication mechanism. Any third-party SSO integration must therefore be set up to coexist properly with IBM Cloud Orchestrator REST APIs. Many SSO modules including Kerberos integration therefore support to limit the application of the SSO interception to certain URI paths. Ensure that only the following paths are used for SSO:

- `ProcessCenter`
- `Process Admin`
- `portal`
- `login`

Configuring the logout redirect URL

By default, IBM Cloud Orchestrator redirects to the login page after a successful logout. In an SSO context, it might be desirable to redirect to another page instead. The other page can perform a log out of the SSO context. Some customers also prefer to redirect to a company home page or tools launch page.

IBM Cloud Orchestrator supports configuring the logout redirect URL through its customization properties file. To do that, set the logout URL as `logoutUrl` in the `customizations.json` metadata properties. The logout URL must be an absolute URL like `http://www.example.com`.

For more information about configuring the `customizations.json` file, see "Metadata file properties in customization file" on page 161.

Configuring online backup

Implement an online backup solution in your IBM Cloud Orchestrator environment.

The solution described in this topic is only related to the IBM Cloud Orchestrator Server. This solution does *not* include the backup of the IBM Cloud Manager with OpenStack environment including hypervisors, compute nodes, and deployed virtual machines. For information about backing up the IBM Cloud Manager with OpenStack environment, see Backing up and restoring IBM Cloud Manager with OpenStack.

You must use this backup and restore solution only when the entire primary production environment fails and it cannot be restored in the current site but it must be restored in a newly installed environment.

When IBM Cloud Orchestrator is running on the primary site, the IBM Cloud Orchestrator databases and configuration files are backed up as shown in the following picture.

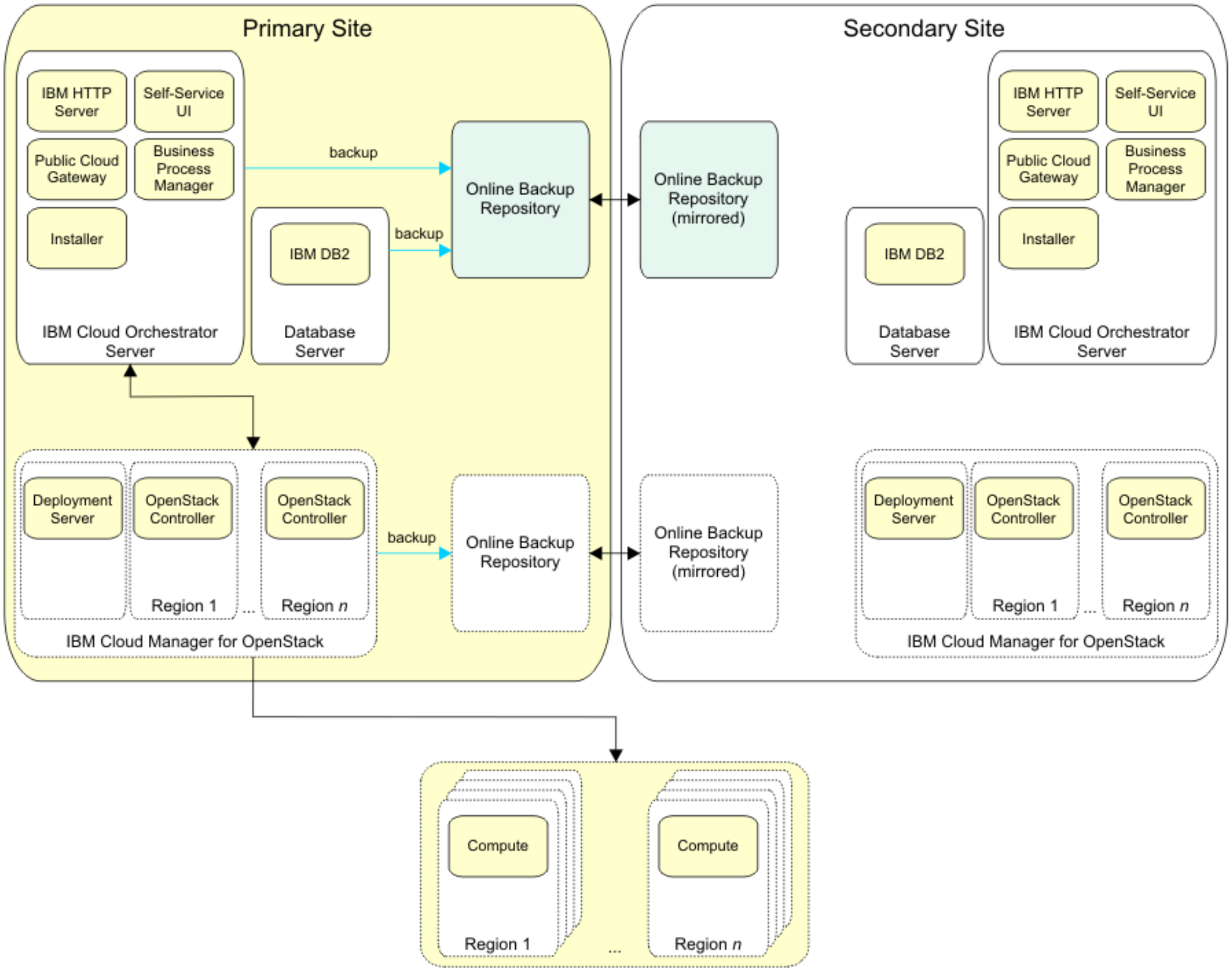


Figure 2. Backup phase

When the primary site goes down, the IBM Cloud Orchestrator databases and configuration files are restored on the secondary site as shown in the following picture.

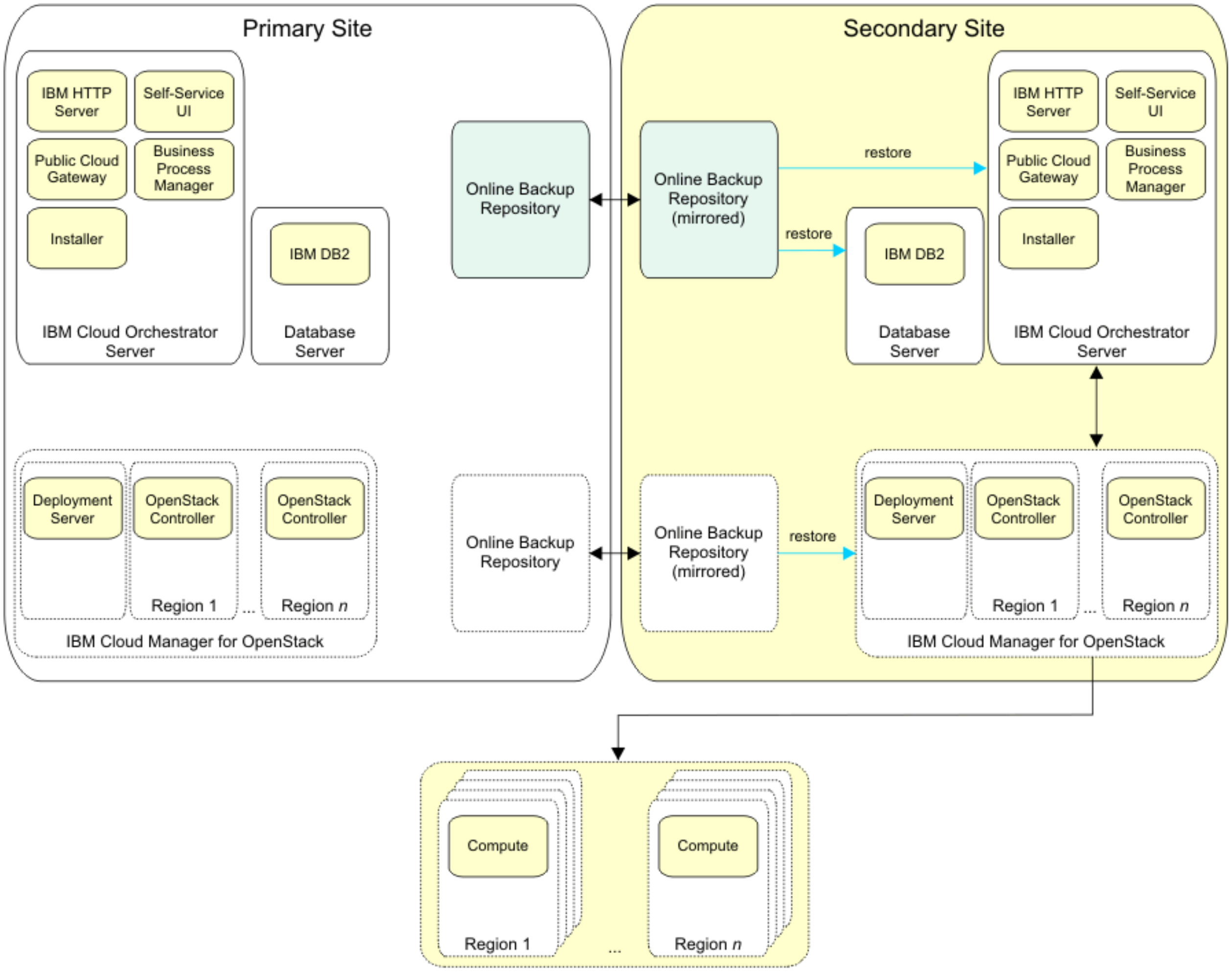


Figure 3. Restore phase

You can use this general solution and adjusting it according to your specific requirements. For example, you may want to work with snapshots, or you may want to leverage backup tools like Tivoli Storage Manager, Veritas, or File System Replication. In case of database similar to the file system backup tools, you can also use solutions like DB2 high availability disaster recovery (HADR). No specific tool is recommended to implement the solution described in this topic.

The backup and restore solution covers the following IBM Cloud Orchestrator data:

- IBM Cloud Orchestrator databases, also containing:
  - New toolkits added to your IBM Cloud Orchestrator environment
  - Changes that you made to the toolkits delivered with IBM Cloud Orchestrator
  - New offerings, categories, and instance type that you added
  - Changes that you made to the predefined offerings and categories
  - Request history and task queue data
- Configuration files, also including:
  - Self-service user interface configuration files and customization
  - Public Cloud Gateway configuration files and customization
  - Business Process Manager scripts that you uploaded
  - IBM Cloud Orchestrator installation response file

For information about the list of databases and configuration files that are backed up, see "Listing databases and configuration files" on page 123. You may want to add to the list any other file that you added to customize your IBM Cloud Orchestrator environment (for example, your company logo that you eventually used in the Self-service user interface).

The IBM Cloud Orchestrator Server can be recovered in the following disaster scenarios:

- Losing power: server is healthy, and no data is lost. You do not need to restore the IBM Cloud Orchestrator Server or the IBM Cloud Manager with OpenStack cluster.
- Losing configuration data: application is still there. You only need to restore configuration or cloud resources (for example, images and volumes).
- Losing application: server is healthy, but IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack are removed by mistake. You need to redeploy IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack.
- Server crash or disk does not work: you need to replace the server and redeploy IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack.
- Site disaster: power or network is lost in the primary datacenter. You need to activate the IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack servers in a secondary datacenter.

You can also use this backup solution to configure periodic backups by using any solution that can create periodic copies of the required databases and files. The advantage of this online backup solution is that the data loss is near to zero because both the database and the configuration files are replicated in real time.

Prerequisites and assumptions

Before implementing this backup solution, be sure to understand the following requirements and assumptions.

- The primary and secondary sites must have exactly the same IBM Cloud Orchestrator product version and the same network configuration. The IP address and the host name must be the same in both the IBM Cloud Orchestrator Server and the DB2 server.
- You must recover manually the inconsistency that might affect the following data:
  - In-progress data while taking backup
  - Hypervisor changes that occur in the time period between backup and restore
  - Changes that occur in the time period between backing up different components
- The backup repository that you manage must be at a different disaster realm. For example, the backup would not be lost in case of a disaster.
- The current solition does not guarantee that recovery point is zero, but it is close to zero. Because IBM Cloud Orchestrator, IBM Business Process Manager, and IBM Cloud Manager with OpenStack are not transaction safe as banking application, it is not possible to provide a recovery point of zero.
- IBM Cloud Manager with OpenStack must be correctly backed up and restored before IBM Cloud Orchestrator is backed up. For information about IBM Cloud Manager with OpenStack backup and restore, see Backing up and restoring IBM Cloud Manager with OpenStack.
- Only full backup and restore is supported. No partial restore of specific components is supported.
- SmartCloud Cost Management backup is not included in this solution.

Backing up and restoring IBM Cloud Orchestrator

Learn how to implement an end-to-end solution for the online backup process of IBM Cloud Orchestrator.

The following key data that must be backed up and restored for IBM Cloud Orchestrator:

- Databases
- Configuration files

For information about the list of databases and configuration files that are backed up, see "Listing databases and configuration files" on page 123.

The main steps of the backup process are:

1. Implementing the mechanism to do the online backup of IBM Cloud Orchestrator databases by using, for example, the DB2 HADR solution, or Tivoli Storage Manager, or any other database replication solution.
2. Implementing the mechanism to do the online backup of IBM Cloud Orchestrator configuration files by using `rsync`, or the Tivoli Storage Manager agent, or any other file backup and restore solution.

The restore process depends on the backup solution that was used to back up the required configuration files and databases. The main steps of the restore process are:

1. Restoring the IBM Cloud Orchestrator databases and configuring IBM Cloud Orchestrator to use the databases.
2. Restoring the IBM Cloud Orchestrator configuration files to the specified directories.
3. Changing the Business Process Manager configuration to work in the restored environment.

For information about the backup and restore procedure, see "Backup and restore procedure" on page 124.

For an example scenario of the backup and restore procedure, see "Backup and restore example scenario" on page 125.

Listing databases and configuration files

When implementing an online backup solution for IBM Cloud Orchestrator, you must back up and restore the following databases and configuration files.

Database list

The following IBM Cloud Orchestrator databases must be backed up and restored:

- `BPMDB`
- `CMNDB`
- `PDWDB`

Configuration file list

The following IBM Cloud Orchestrator files and directories must be backed up and restored:

- For the Self-service user interface:

  ```
  /opt/ibm/ico/wlp/usr/servers/scui/etc/customizations/*
  /opt/ibm/ico/scui/etc/dashboard/*
  ```

  Note: The `/opt/ibm/ico/wlp/usr/servers/scui` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.
- For the Public Cloud Gateway:

  ```
  /opt/ibm/ico/wlp/usr/servers/pcg/etc/*
  /opt/ibm/ico/pcg/var/cache/*
  ```

  Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.
- For Business Process Manager:

  ```
  /var/ibm/sco/scriptRepo
  ```

  Note: Back up also the following files but do not restore them directly:

```
/opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/config/cells/PCCell1/fileRegistry.xml
/opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/config/cells/PCCell1/fileRegistry.xml
```

- For the installer component:

```
/opt/ico_install/<IBM Cloud Orchestrator version>/installer/ico_install.rsp
```

You must also back up and restore any file that you changed to customize your IBM Cloud Orchestrator environment.

Backup and restore procedure

Follow this procedure to back up and restore your IBM Cloud Orchestrator environment.

Setting up the backup process

After IBM Cloud Orchestrator was installed and configured on the primary site, to set up the online backup for IBM Cloud Orchestrator, perform the following steps:

1. Set up file replication at regular interval for the configuration files listed in "Configuration file list" on page 123.

   Note: When replicating the files, you must preserve file permissions (owner, group, and other). If SELinux is enabled, ensure that security permissions are correctly set.
2. Implement the backup for the primary DB2 by using one of the following solutions:
   a. Backup and restore DB2 from the primary site to the secondary site manually or by using customized scripts. For an example of backup and restore implementation, see "Implementing backup and restore" on page 125.

   b. HADR between primary DB2 and staging DB2, where the primary DB2 acts as active node and the staging DB2 acts as standby node. For this solution, on the secondary site you must install and configure a staging DB2 that is used only to back up data from primary site. For an example of HADR implementation, see "Implementing DB2 HADR" on page 126.

   Note: The staging DB2 server must have a different IP address or host name than the primary DB2 server IP address or host name.

For information about backing up the IBM Cloud Manager with OpenStack environment, see Backing up and restoring IBM Cloud Manager with OpenStack.

Restoring IBM Cloud Orchestrator

To restore IBM Cloud Orchestrator, perform the following steps:

1. Restore the IBM Cloud Manager with OpenStack server by following the procedure described in Backing up and restoring IBM Cloud Manager with OpenStack.
2. When the IBM Cloud Manager with OpenStack restore is completed, perform one of the following actions:
   - Install and configure IBM Cloud Orchestrator and DB2 on the secondary site using the same IP address and host name of IBM Cloud Orchestrator and DB2 on the primary site.
   - On the secondary site, restore a cloned snapshot of IBM Cloud Orchestrator from the primary site.
3. If you implemented a HADR solution, back up the staging DB2 server. For an example of a backup implementation, see "Implementing backup and restore when using HADR" on page 129.
4. When the IBM Cloud Orchestrator and DB2 installation and configuration is completed on the secondary site, copy the backed up database files to the secondary site DB2 server and restore these databases on the secondary site DB2. For an example of DB2 restore, see "Implementing backup and restore when using HADR" on page 129.
5. Stop all the IBM Cloud Orchestrator services, copy the backed up configuration files on the secondary IBM Cloud Orchestrator Server and restore the configuration files in the specific directories. For information about the configuration files to be restored, see "Configuration file list" on page 123.

Note: Do not restore the `fileRegistry.xml` files.

Note: When restoring the files, you must preserve file permissions (owner, group, and other). If SELinux is enabled, ensure that security permissions are correctly set.

6. Change the Business Process Manager configuration in the restored environment by performing the following steps:

   a. Get the unique ID and external ID from the `fileRegistry.xml` file backed up on the primary site and update the IDs in the following files on the secondary site:

   ```
   /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/config/cells/PCCell1/fileRegistry.xml
   /opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/config/cells/PCCell1/fileRegistry.xml
   ```

   b. Delete the SIB tables in the CMNDB database which stores the message engine UUID by running the following commands:

   ```
   su - db2inst1
   db2 connect to CMNDB
   db2 drop table "BPMUSER.SIB000"
   db2 drop table "BPMUSER.SIB001"
   db2 drop table "BPMUSER.SIB002"
   db2 drop table "BPMUSER.SIBCLASSMAP"
   db2 drop table "BPMUSER.SIBKEYS"
   db2 drop table "BPMUSER.SIBOWNER"
   db2 drop table "BPMUSER.SIBOWNERO"
   db2 drop table "BPMUSER.SIBXACTS"
   db2 drop table "BPMUSER.SIBLISTING"
   ```

7. Restart the IBM Cloud Orchestrator services on the secondary site.

   Note: If you are using DB2 HADR and an error occurs when starting the Business Process Manager services, clear the WebSphere cache by running the following command:

   ```
   /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/bin/clearClassCache.sh
   ```

   Then restart the services.

8. Verify that:

   - All the old IBM Cloud Orchestrator data is correctly restored (like existing categories, offerings, and service requests).
   - All the user interface customized settings are correctly restored.
   - All the existing users are restored and you can log in by using these users.
   - All the offerings are correctly working when you submit them.

9. Clean up any request that was in progress on the primary IBM Cloud Orchestrator and that might fail on the secondary IBM Cloud Orchestrator.

Backup and restore example scenario

See the following procedure as an example of the backup and restore scenario.

Implementing backup and restore

Perform the following steps as an example of backup and restore implementation:

1. Back up DB2 on primary site using `LOGARCHMETH1` with required archive directory by performing the following steps:

   a. Create the directory structure:

   ```
   mkdir -p /backup/online
   mkdir -p /backup/offline
   chown -R db2inst1 /backup
   su - db2inst1
   mkdir -p backup/ArchiveDest
   ```

b. Make DB2 backup ready and take DB2 backup:

```
db2 update database configuration for BPM using LOGARCHMETH1 \
 'DISK:/home/db2inst1/backup/ArchiveDest'

db2 update database configuration for CMNDB using LOGARCHMETH1 \
 'DISK:/home/db2inst1/backup/ArchiveDest'

db2 update database configuration for PDWDB using LOGARCHMETH1 \
 'DISK:/home/db2inst1/backup/ArchiveDest'

db2 quiesce instance db2inst1 restricted access immediate force connections

db2 backup database BPMDB to /backup/offline/

db2 backup database CMNDB to /backup/offline/

db2 backup database PDWDB to /backup/offline/

db2 unquiesce instance db2inst1
```

c. Take the online backup:

```
db2 -v "backup db BPMDB online to /backup/online compress include logs"

db2 -v "backup db CMNDB online to /backup/online compress include logs"

db2 -v "backup db PDWDB online to /backup/online compress include logs"

db2ckbkp <backup_file_location>
```

2. Restore DB2 on the secondary site by performing the following steps:

   a. Copy DB2 backup files from the primary site.

   b. Run the following commands:

```
db2 quiesce instance db2inst1 restricted access immediate force connections

db2 restore database BPMDB from /backup/online/ taken at <timestamp of backup>
db2 restore database CMNDB from /backup/online/ taken at <timestamp of backup>
db2 restore database PDWDB from /backup/online/ taken at <timestamp of backup>

db2 rollforward db BPMDB to end of backup and complete
db2 rollforward db CMNDB to end of backup and complete
db2 rollforward db PDWDB to end of backup and complete

db2 unquiesce instance db2inst1

db2 connect to BPMDB
db2 connect to CMNDB
db2 connect to PDWDB
```

Implementing DB2 HADR

To implement backup and restore by using DB2 HADR, perform the following steps:

1. After IBM Cloud Orchestrator is installed and configured on primary site, install and configure DB2 on secondary site which will be referred as staging DB2.
2. Run the `create_dbs.sh` script on staging DB2 server to create all the three databases and users for the staging DB2. For information about the `create_dbs.sh` script, see "[Optional] Creating the Business Process Manager databases on an external IBM DB2 server" on page 30.

   See the following details for both the primary and staging DB2 servers used in this sample scenario:

Primary DB2 server details	
Host name	`databasePrimary.ibm.com`
Primary DB instance	`db2inst1`
Primary port	`50001`
Database name	`BPMDB`

Primary DB2 server details	
HADR port for database BPMDB	64000
Database name	CMNDB
HADR port for database CMNDB	64002
Database name	PDWDB
HADR port for database PDWDB	64004

Staging DB2 server details	
Host name	databaseStaging.ibm.com
Primary DB instance	db2inst1
Primary port	50001
Database name	BPMDB
HADR port for database BPMDB	64001
Database name	CMNDB
HADR port for database CMNDB	64003
Database name	PDWDB
HADR port for database PDWDB	64005

3. Run the following commands on the primary DB2 to configure HADR:

   a. Log on with `db2inst1` user and create the following directories:

   ```
   mkdir -p /home/db2inst1/archived_logs/BPMDB
   mkdir -p /home/db2inst1/archived_logs/CMNDB
   mkdir -p /home/db2inst1/archived_logs/PDWDB
   ```

   b. Make primary DB2 backup ready:

   ```
   db2 "update database configuration for BPMDB using LOGARCHMETH1 \
    'DISK:/home/db2inst1/archived_logs/BPMDB' LOGINDEXBUILD ON"

   db2 "update database configuration for CMNDB using LOGARCHMETH1 \
    'DISK:/home/db2inst1/archived_logs/CMNDB' LOGINDEXBUILD ON"

   db2 "update database configuration for PDWDB using LOGARCHMETH1 \
    'DISK:/home/db2inst1/archived_logs/PDWDB' LOGINDEXBUILD ON"
   ```

   c. Take databases backup on primary DB2:

   ```
   mkdir -p /home/db2inst1/db_backup
   db2 quiesce instance db2inst1 restricted access immediate force connections
   db2 "backup database BPMDB to /home/db2inst1/db_backup"
   db2 "backup database CMNDB to /home/db2inst1/db_backup"
   db2 "backup database PDWDB to /home/db2inst1/db_backup"
   db2 unquiesce instance db2inst1
   ```

   d. Copy these backup files on staging DB2 server to the `/home/db2inst1/db_backup` directory.

   Note: When copying the files, you must preserve file permissions.

4. Run the following command on the staging DB2:

   a. Restore the databases backup on staging DB2

   ```
   db2 "restore database BPMDB from /home/db2inst1/db_backup"
   db2 "restore database CMNDB from /home/db2inst1/db_backup"
   db2 "restore database PDWDB from /home/db2inst1/db_backup"
   ```

   b. Validate that all the three databases are in ROLL-FORWARD PENDING state:

      ```
      db2 connect to BPMDB
      db2 connect to CMNDB
      db2 connect to PDWDB
      ```

      These commands should display the Cannot connect to database error because the database is in ROLL-FORWARD PENDING state.

5. Make both the primary and staging DB2 servers HADR ready by performing the following steps on both the servers:

   a. Edit the /etc/hosts file and add an entry for both the servers for host name resolution.

   b. Edit the /etc/services file on both the servers and add the following entries:

      ```
      DB2_HADR_BPMDBp    64000/tcp
      DB2_HADR_BPMDBs    64001/tcp
      DB2_HADR_CMNDBp    64002/tcp
      DB2_HADR_CMNDBs    64003/tcp
      DB2_HADR_PDWDBp    64004/tcp
      DB2_HADR_PDWDBs    64005/tcp
      ```

      Note: You can use any free tcp port to allow to the databases to communicate each other.

6. Turn off automatic startup of the database instance on both primary and staging DB2 servers. In this example, the instance name is db2inst1:

   ```
   su – db2inst1
   cd sqllib/bin
   ./db2iauto -off db2inst1
   ```

7. Apply the HADR database configuration parameters for all the databases on both primary and staging servers:

   a. HADR configuration for the BPMDB database on the primary DB2:

      ```
      db2 "update db cfg for BPMDB using hadr_local_host databasePrimary.ibm.com"
      db2 "update db cfg for BPMDB using hadr_remote_host databaseStaging.ibm.com"
      db2 "update db cfg for BPMDB using hadr_local_svc 64000"
      db2 "update db cfg for BPMDB using hadr_remote_svc 64001"
      db2 "update db cfg for BPMDB using hadr_remote_inst db2inst1"
      db2 "update db cfg for BPMDB using hadr_timeout 120"
      db2 "update db cfg for BPMDB using hadr_syncmode nearsync"
      db2 update alternate server for database BPMDB using hostname
      <Staging_Database_IP> port 50001
      ```

   b. HADR configuration for the BPMDB database on the staging DB2:

      ```
      db2 "update db cfg for BPMDB using hadr_local_host databaseStaging.ibm.com"
      db2 "update db cfg for BPMDB using hadr_remote_host databasePrimary.ibm.com"
      db2 "update db cfg for BPMDB using hadr_local_svc 64001"
      db2 "update db cfg for BPMDB using hadr_remote_svc 64000"
      db2 "update db cfg for BPMDB using hadr_remote_inst db2inst1"
      db2 "update db cfg for BPMDB using hadr_timeout 120"
      db2 "update db cfg for BPMDB using hadr_syncmode nearsync"
      db2 update alternate server for database BPMDB using hostname
      <Primary_Database_IP> port 50001
      ```

   c. HADR configuration for the CMNDB database on the primary DB2:

      ```
      db2 "update db cfg for CMNDB using hadr_local_host databasePrimary.ibm.com"
      db2 "update db cfg for CMNDB using hadr_remote_host databaseStaging.ibm.com"
      db2 "update db cfg for CMNDB using hadr_local_svc 64002"
      db2 "update db cfg for CMNDB using hadr_remote_svc 64003"
      db2 "update db cfg for CMNDB using hadr_remote_inst db2inst1"
      db2 "update db cfg for CMNDB using hadr_timeout 120"
      db2 "update db cfg for CMNDB using hadr_syncmode nearsync"
      db2 update alternate server for database CMNDB using hostname
      <Staging_Database_IP> port 50001
      ```

d. HADR configuration for the CMNDB database on the staging DB2:

```
db2 "update db cfg for CMNDB using hadr_local_host databaseStaging.ibm.com"
db2 "update db cfg for CMNDB using hadr_remote_host databasePrimary.ibm.com"
db2 "update db cfg for CMNDB using hadr_local_svc 64003"
db2 "update db cfg for CMNDB using hadr_remote_svc 64002"
db2 "update db cfg for CMNDB using hadr_remote_inst db2inst1"
db2 "update db cfg for CMNDB using hadr_timeout 120"
db2 "update db cfg for CMNDB using hadr_syncmode nearsync"
db2 update alternate server for database CMNDB using hostname
<Primary_Database_IP> port 50001
```

e. HADR configuration for the PDWDB database on the primary DB2:

```
db2 "update db cfg for PDWDB using hadr_local_host databasePrimary.ibm.com"
db2 "update db cfg for PDWDB using hadr_remote_host databaseStaging.ibm.com"
db2 "update db cfg for PDWDB using hadr_local_svc 64004"
db2 "update db cfg for PDWDB using hadr_remote_svc 64005"
db2 "update db cfg for PDWDB using hadr_remote_inst db2inst1"
db2 "update db cfg for PDWDB using hadr_timeout 120"
db2 "update db cfg for PDWDB using hadr_syncmode nearsync"
db2 update alternate server for database PDWDB using hostname
<Staging_Database_IP> port 50001
```

f. HADR configuration for the PDWDB database on the staging DB2:

```
db2 "update db cfg for PDWDB using hadr_local_host databaseStaging.ibm.com"
db2 "update db cfg for PDWDB using hadr_remote_host databasePrimary.ibm.com"
db2 "update db cfg for PDWDB using hadr_local_svc 64005"
db2 "update db cfg for PDWDB using hadr_remote_svc 64004"
db2 "update db cfg for PDWDB using hadr_remote_inst db2inst1"
db2 "update db cfg for PDWDB using hadr_timeout 120"
db2 "update db cfg for PDWDB using hadr_syncmode nearsync"
db2 update alternate server for database PDWDB using hostname
<Primary_Database_IP> port 50001
```

8. Validate the HADR configuration for all the databases on primary and staging DB2:

```
db2 get db cfg for BPMDB|grep "HADR"
db2 get db cfg for CMNDB|grep "HADR"
db2 get db cfg for PDWDB|grep "HADR"
```

9. On the staging DB2, start all the three databases as standby:

```
db2 "start hadr on database BPMDB as standby"
db2 "start hadr on database CMNDB as standby"
db2 "start hadr on database PDWDB as standby"
```

10. On the primary DB2, start all the three databases as primary:

```
db2 "start hadr on database BPMDB as primary"
db2 "start hadr on database CMNDB as primary"
db2 "start hadr on database PDWDB as primary"
```

11. On the primary DB2, validate the HADR status for all the three databases:

```
db2pd -db BPMDB -hadr
db2pd -db CMNDB -hadr
db2pd -db PDWDB -hadr
```

Implementing backup and restore when using HADR

Perform the following steps as an example of backup and restore implementation when using HADR solution:

1. When the primary site goes down, take the database backup from the staging DB2 by running the following steps on the staging DB2 server:

   a. Run the following database take over commands:

```
db2 takeover hadr on database BPMDB BY FORCE
db2 takeover hadr on database CMNDB BY FORCE
db2 takeover hadr on database PDWDB BY FORCE
```

b. Make the staging DB2 backup ready by running the following commands:

```
mkdir -p /home/db2inst1/archived_logs/BPMDB
mkdir -p /home/db2inst1/archived_logs/CMNDB
mkdir -p /home/db2inst1/archived_logs/PDWDB

db2 "update database configuration for BPMDB using LOGARCHMETH1 \
 'DISK:/home/db2inst1/archived_logs/BPMDB' LOGINDEXBUILD ON"

db2 "update database configuration for CMNDB using LOGARCHMETH1 \
 'DISK:/home/db2inst1/archived_logs/CMNDB' LOGINDEXBUILD ON"

db2 "update database configuration for PDWDB using LOGARCHMETH1 \
 'DISK:/home/db2inst1/archived_logs/PDWDB' LOGINDEXBUILD ON"
```

c. Take database backup on the staging DB2 by running the following commands:

```
db2 "backup database BPMDB to /home/db2inst1/db_backup"
db2 "backup database CMNDB to /home/db2inst1/db_backup"
db2 "backup database PDWDB to /home/db2inst1/db_backup"
```

2. After you installed and configured IBM Cloud Orchestrator and DB2 on the secondary site, copy the backed up database on this new DB2 and run the following commands (assuming that the files were copied in the `/backup/offline` directory):

```
db2 quiesce instance db2inst1 restricted access immediate force connections
db2 restore database BPMDB from /backup/offline/ taken at <timestamp of backup>
db2 restore database CMNDB from /backup/offline/ taken at <timestamp of backup>
db2 restore database PDWDB from /backup/offline/ taken at <timestamp of backup>
db2 rollforward db BPMDB to end of backup and complete
db2 rollforward db CMNDB to end of backup and complete
db2 rollforward db PDWDB to end of backup and complete
db2 unquiesce instance db2inst1
```

3. Verify the connection to the databases by running the following commands:

```
db2 connect to BPMDB
db2 connect to CMNDB
db2 connect to PDWDB
```

Troubleshooting

- If the primary DB2 is shut down or restarted, perform the following command to start the services for backup after the primary DB2 is started:

```
su - db2inst1
db2start
db2 activate db BPMDB
db2 activate db CMNDB
db2 activate db PDWDB
db2 connect to BPMDB
db2 connect to CMNDB
db2 connect to PDWDB
```

- If the staging DB2 is shut down or restarted, perform the following command to start the services for backup after the staging DB2 is started:

```
su - db2inst1
db2start
db2 activate db BPMDB
db2 activate db CMNDB
db2 activate db PDWDB
```

- When backing up the staging DB2 server by running the following command:

```
db2 "backup database <db_name> to /home/db2inst1/db_backup"
```

if the following error occurs:

```
SQL1035N The operation failed because the specified database cannot be
connected to in the mode requested. SQLSTATE=57019
```

run the following command to solve the problem:

```
db2 deactivate db <db_name>
```

- When rolling forward a database by running the following command:

```
db2 rollforward db, <db_name> to end of backup and complete
```

if the following error occurs:

```
SQL1273N An operation reading the logs on database <"db_name"> cannot continue
because of a missing log file "S0000001.LOG" on database partition "0" and log
stream "0".
```

run the following commands to solve the problem:

```
db2 quiesce instance db2inst1 restricted access immediate force connections
db2 force application all
db2 deactivate db <db_name>

mkdir -p /backup/online/tmp_logs/
ls -ld /backup/online/tmp_logs/

db2 -v "RESTORE DATABASE  <db_name> from /backup/online taken at <timestamp of backup> \
 LOGTARGET '/backup/online/tmp_logs' REPLACE EXISTING"

db2 -v "rollforward db <db_name> to end of backup and stop OVERFLOW LOG PATH \
 ('/backup/online/tmp_logs/NODE0000')"
```

Strengthening security

Complete these tasks to strengthen the security of your IBM Cloud Orchestrator environment.

Changing the various passwords

Change the password for several different types of users in the IBM Cloud Orchestrator environment.

- Built-in users:
  - "Changing the bpm_admin and tw_admin passwords" on page 132
- Database users:
  - "Changing the db2inst1 password" on page 133
  - "Changing the bpmuser password" on page 133
- Keystore:
  - "Changing the IBM HTTP Server keystore password" on page 134

Note:

- During installation and upgrade, IBM Cloud Orchestrator passwords can contain only the following characters:

```
a-z A-Z 0-9 - . _ ` ~ @
```

  Restriction: The passwords cannot contain spaces.
- If you use external database support, contact your database administrator to change the password according to the external IBM DB2 configuration.
- The IBM DB2 password must adhere to the same character set as that of IBM Cloud Orchestrator.

- For information about how to change OpenStack passwords, see the documentation for your chosen OpenStack product; for example, see Changing passwords and secrets in the IBM Cloud Manager with OpenStack documentation.

Changing the `bpm_admin` and `tw_admin` passwords

The `bpm_admin` and `tw_admin` users are required by Business Process Manager for internal operations.

To change the `bpm_admin` password, complete the following steps:

1. Log in to WebSphere Application Server:

   ```
   https://$ico_server:9043/ibm/console/logon.jsp
   ```

2. Expand **Users and Groups**, and click **Manage Users**.
3. Select **bpm_admin**.
4. In the **User Properties** panel, set the password, confirm it, and click **Apply**.
5. On the IBM Cloud Orchestrator Server, change the configuration files as follows:

   a. Back up the configuration files:

   ```
   /opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/properties/soap.client.props
   /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/properties/soap.client.props
   ```

   b. Edit each of the `soap.client.props` files that are listed in step "5.a" on page 132 to find the `com.ibm.SOAP.loginUserid=bpm_admin` entry, and update the associated `com.ibm.SOAP.loginPassword` entry to specify the new password as plain text:

   ```
   com.ibm.SOAP.loginUserid=bpm_admin
   com.ibm.SOAP.loginPassword=new_bpm_admin_password
   ```

   c. Encrypt the password, by running the following commands:

   ```
   /opt/ibm/ico/BPM/v8.5/bin/PropFilePasswordEncoder.sh \
   /opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/properties/soap.client.props \
   com.ibm.SOAP.loginPassword

   /opt/ibm/ico/BPM/v8.5/bin/PropFilePasswordEncoder.sh \
   /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/properties/soap.client.props \
   com.ibm.SOAP.loginPassword
   ```

6. Update the BPM Aliases and RunAs roles passwords by running the following steps:

   a. Stop the Business Process Manager deployment manager by running the following command:

   ```
   systemctl stop bpm-dmgr
   ```

   b. Run the following command to update the passwords:

   ```
   /opt/ibm/ico/BPM/v8.5/profiles/DmgrProfile/bin/wsadmin.sh -conntype NONE

   wsadmin>$AdminTask updateBPMAliasesAndRunAsRolesPasswords {-userName bpm_admin \
           -password NEWBPMPASSWORD }
   wsadmin>$AdminConfig save
   wsadmin>exit
   ```

   c. Start the Business Process Manager deployment manager by running the following command:

   ```
   systemctl start bpm-dmgr
   ```

   For more information about updating BPM Aliases and RunAs roles passwords, see the Business Process Manager documentation.

7. Stop and start IBM Cloud Orchestrator by running the following commands:

   ```
   /opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --stop
   /opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --start
   ```

To change the password of the tw_admin user, complete the same procedure as described for the bpm_admin user, but omit step “5” on page 132. Do not modify any soap.client.props files.

Note: The bpm_admin and tw_admin passwords must be the same.

Changing the db2inst1 password

The db2inst1 password must be changed in the operating system where the IBM DB2 instance is installed, as follows:

1. Log in to the IBM Cloud Orchestrator Server as the root user.
2. Change the operating system password for the IBM DB2 database user db2inst1 by running the following command. After the command, you must enter the new password.

   ```
   passwd db2inst1
   ```

Changing the bpmuser password

The bpmuser user is the IBM DB2 user for Business Process Manager.

The bpmuser password must be changed in the operating system where the IBM DB2 instance is installed, and in the WebSphere Application Server console that is used by Business Process Manager.

Note: This procedure will result in a short outage of the IBM Cloud Orchestrator when the Business Process Manager is restarted

1. Update the bpmuser password in the operating system, as follows:
   a. Log in to the IBM Cloud Orchestrator Server as the root user.
   b. Change the operating-system password for the bpmuser database user:

   ```
   passwd bpmuser
   ```

2. Update the password in WebSphere Application Server, as follows:
   a. Log in to the Business Process Manager WebSphere Application Server console as the bpm_admin user:

   ```
   https://$ico_server:9043/ibm/console/logon.jsp
   ```

   b. Select **Resources**.
   c. Select **JDBC**.
   d. Select **Data sources** and click **BPM Business Space data source**.
   e. Click the option **JAAS - J2C authentication data**.
   f. Click **BPM_DB_ALIAS**, and insert the new password. Click **Apply** to validate the change.
   g. When prompted to save your changes, click **Save directly to the master configuration**.
   h. Repeat steps “2.f” on page 133 and “2.g” on page 133 for the **CMN_DB_ALIAS** and **PDW_DB_ALIAS** values.
   i. Test the DB connection by returning to **Resources** > **Data sources**, selecting **BPM Business Space data source**, and clicking **Test connection**.
   j. Restart Business Process Manager by running the following command as user root:

   ```
   systemctl restart bpm
   ```

   Wait until Business Process Manager completely restarts before continuing to use IBM Cloud Orchestrator.

   If you get errors while synchronizing the changes, log out and log in again, and try to modify the password again.

For more information about updating passwords in WebSphere Application Server, see Updating the data source authentication alias.

Changing the IBM HTTP Server keystore password

The IBM HTTP Server keystore password is used for managing certificates. To replace the existing certificate, see "Replacing the existing certificates" on page 134. To change the password, perform the following procedure:

1. Log in to the IBM Cloud Orchestrator Server as `root`.
2. Change the keystore password:

```
cd /opt/ibm/ico/HTTPServer/bin
./gskcmd -keydb -changepw -db key.kdb -new_pw <new_password> \
        -pw <old_password> -stash
```

Replacing the existing certificates

You can replace existing certificates to strengthen security in your environment.

Important: Before running the following procedure to replace the existing certificates, back up your IBM Cloud Orchestrator Server.

This procedure requires to receive signed server certificates from a well-known certificate authority (CA). The browser checks the certificates while browsing the IBM Cloud Orchestrator user interface. Initially there are self-signed certificates installed during the IBM Cloud Orchestrator installation but this verification chain does not end with a well-known CA and, therefore, the browser responds with certificate exceptions. Root certificates are also self signed but the CA is well known if a valid CA is used to sign the server certificates. Root and intermediate certificates distributed by the CA with the signed server certificates are called *signer certificates*.

You can look into the certificates by using OpenSSL that is usually installed in the Linux systems:

```
 openssl x509 -noout -text -in <fully_qualified_path_to_certificate>
```

The Self-service user interface is served by a central IBM HTTP Server. Since this is the main entry point, to replace the certificates start from the `/opt/ibm/ico/HTTPServer/bin` directory that contains the IBM HTTP Server key database with CMS encrypted `key.kdb`.

Exchanging the certificates is a complex process and some background knowledge is required. If errors occur, check the `/opt/ibm/ico/HTTPServer/logs/error_log` file for troubleshooting. For listing content, deleting certificates, creating certificate requests, adding signer certificates, and receiving personal certificates, you can use the `ikeycmd` or the `ikeyman` utilities. The `ikeyman` user interface requires a running X server. Both the utilities have the same options for the required operations. `arm` is the certificate extension accepted by the `ikeyman` utility by default, but you can also use all the other extensions for certificate files.

Note: Use the Java program that is configured with the cryptographic providers for the IBM HTTP Server. To verify the correct Java program with the provider configuration, run the following command:

```
/opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -keydb -list
```

The currently supported key database encryption types are displayed. For example:

```
CMS
JKS
JCEKS
PKCS12
PKCS12S2
PKCS11Direct
```

At least the `CMS` and `PKCS12` types must be listed. If `CMS` is not listed, use `/opt/ibm/ico/HTTPServer/bin/gskcmd` program for all the commands.

To make sure that you are using the correct Java configuration, you must call the `ikeycmd` command by using the fully qualified path:

```
/opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd
```

Discovering the keystore password

The SSL certificate that is used by the IBM HTTP Server is contained in a file that is called a certificate store. This file is protected by a password. By default, this password is set to the value of the master administrator password that is set during the IBM Cloud Orchestrator installation.

Preparing the IBM HTTP Server key database

IBM Cloud Orchestrator requires a certificate for the IBM Cloud Orchestrator Server. To prepare the IBM HTTP Server key database, perform the following steps:

1. Log in to the IBM Cloud Orchestrator Server and gain `root` privilege with the su command.
2. Change directory to `/opt/ibm/ico/HTTPServer/bin`.
3. Back up the existing certificate store:

   ```
   cp key.kdb key.kdb.bak
   ```

4. Check that the keystore password works and get a list of the certificates in the certificate store:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -list -db key.kdb -pw
   <password>
   ```

   The output shows two certificates. The name of the first certificate is the fully qualified domain name (FQDN) of the virtual address of IBM Cloud Orchestrator Server. Make a note of this name because you are required to enter the name in the following steps when `<ico_server_fqdn>` is specified. The second certificate name starts with a long numeric label followed by a number of parameters. This is an internal certificate used by the IBM HTTP Server to forward traffic to the Self-service user interface on port 7443. You must not modify or delete this certificate.
5. Remove the existing SSL certificate by running the following command:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -delete -label
   <ico_server_fqdn> \
            -db key.kdb -pw <password>
   ```

6. If you do not already have a security certificate for the IBM Cloud Orchestrator Server, perform the following steps:

   a. Create the certificate request by running the following command:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -certreq -create -label
   <ico_server_fqdn> \
            -dn "CN=<fqdn>,O=<your organization>,OU=<your division>,C=<your
   country code>" \
            -db key.kdb -file certreq_ico.arm -pw <password> -size 2048 -sig_alg
   SHA256WithRSA
   ```

   b. In the current directory, locate the `certreq_ico.arm` file in the current directory and upload it to your Certificate Authority (CA) for signing.

Installing the new certificate

To install the new certificate, perform the following steps:

1. If you performed step “6” on page 135 in the “Preparing the IBM HTTP Server key database” on page 135 procedure to generate a certificate request, when the CA returns the signed certificate, download it as `cert_ico.arm`. Download also the root and intermediate CA certificates. Consult the Certificate Authority's online help for details about the required root and intermediate CA certificates.

2. Import the root and intermediate CA certificates by running the following command for each certificate:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -add -db key.kdb \
             -pw <password> -file <downloaded_root_or_intermediate_certificate>
   ```

   Note: The root and intermediate CA certificates are self signed and they are required in the key database to enable the browser to verify the signed certificates in the full dependency chain against the CA. If using the `ikeyman` user interface, select the signed certificates category.

3. Do one of the following actions:

   - If you performed step "6" on page 135 in the "Preparing the IBM HTTP Server key database" on page 135 procedure to generate a certificate request, add the new SSL certificate by running the following command:

     ```
     /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -receive \
               -db key.kdb -pw <password> -file cert_ico.arm
     ```

     Note: All the certificates that you requested are personal certificates and they need to go into the personal certificates category if you are using the `ikeyman` user interface. These certificates can only be received if the matching certificate requests are in `key.kdb`. You can list the requests by using the following command:

     ```
     /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -certreq -list -db key.kdb -pw <password>
     ```

     Ensure to generate only one request for each alias.

   - If you already had a security certificate for the IBM Cloud Orchestrator Server, add it to the key database by running the following command:

     ```
     /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -import -target key.kdb \
               -pw <password> -file <ico_server_certificate_file>
     ```

4. Check that the certificate was added to the certificate store by running the following command:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -list -db key.kdb -pw <password>
   ```

5. Make the IBM Cloud Orchestrator Server certificate the default certificate by running the following command:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -setdefault \
             -db key.kdb -pw <password> -label <ico_server_fqdn>
   ```

6. Check the default certificate by running the following command and add a reminder to your calendar for the expiration date of the certificate:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -getdefault -db key.kdb -pw <password>
   ```

7. To test the validation path of your imported personal certificates for the IBM Cloud Orchestrator Server, run the following command:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -validate \
             -db key.kdb -pw <password> -label <ico_server_fqdn>
   ```

   Note: If the validation is not successful, your certificate chain is corrupt. Do not continue with the following procedures before solving the issue. You can test the validation by using also the `ikeyman` user interface.

8. Restart the IHS service by running the following command:

   ```
   systemctl restart ihs
   ```

Updating the certificate on the secondary IBM Cloud Orchestrator Server

Only for high-availability environments of IBM Cloud Orchestrator, do the following additional steps:

1. Log on to the secondary IBM Cloud Orchestrator server and gain root privilege with the su command.
2. Change directory to `/opt/ibm/ico/HTTPServer/bin/`.
3. To back up the existing certificate store, run

   ```
   cp key.kdb key.kdb.bak.
   ```
4. To copy the certificate store from the primary node to the secondary node, run the following command:

   ```
   scp <ico_primary_node_hostname>:/opt/ibm/ico/HTTPServer/bin/key.kdb
   ```

Updating the WebSphere truststores and keystores

You must update the certificate in the WebSphere truststores and keystores so that WebSphere can establish SSL connections to the IBM HTTP Server.

Note: If any problem occurs during the certificate replacement of the WebSphere Application Server, you can temporary disable the WebSphere administrative security. For more information, see http://www.ibm.com/support/docview.wss?uid=swg21405302.

Run the following procedure:

1. Log on to the WebSphere Console. Use the browser to access `https://<ico_server_fqdn>:9043/ibm/console`

   The user ID is `bpm_admin` and the password is the master password specified when you installed IBM Cloud Orchestrator.
2. Retrieve the signer information for the WebSphere cell default truststore:
   a. Navigate to **Security > SSL Certificate and key management**.
   b. In **Configuration settings**, click **Manage endpoint security configurations**.
   c. Select the appropriate outbound configuration to get to the `(cell):PCCell1` management scope.
   d. In **Related Items**, click **Key stores and certificates** and click the **CellDefaultTrustStore** key store.
   e. In **Additional Properties**, click **Signer certificates** and **Retrieve From Port**.
   f. Enter `<ico_server_fqdn>` in the **Host** field, enter 443 in the **Port** field, and enter `CA root chain certificate` in the **Alias** field.
   g. Click **Retrieve Signer Information**.
   h. Verify that the certificate information is for a certificate that you can trust.
   i. Click **OK**.
3. Replace the SSL Certificates in the WebSphere cell default truststore:
   a. Navigate to **Security > SSL Certificate and key management**.
   b. Choose **Key stores and certificates**.
   c. Keep the default of `SSL keystores` in the drop-down list.
   d. Click **CellDefaultTrustStore** and then **Personal certificates**.
   e. Select the entry where the **Alias** matches the IBM Cloud Orchestrator Server FQDN of the certificate you requested before and press **Delete**.
   f. Click **Import**.
   g. Choose **Key store file** and enter `/opt/ibm/ico/HTTPServer/bin/key.kdb` for the key file name.
   h. Change **Type** to `CMSKS`.
   i. Enter the keystore password.
   j. Click **Get Key File Aliases**.
   k. Choose the certificate that matches the IBM Cloud Orchestrator Server FQDN from the drop-down list.

   l. Enter `<ico_server_fqdn>` in the **Imported certificate alias** field.
   m. Click **OK**.
   n. **Save** the change direct to the master configuration.

4. Repeat the steps “2” on page 137 and “3” on page 137 for all the existing truststores in your WebSphere environment.
5. Retrieve the signer information for the WebSphere cell default keystore:
   a. Navigate to **Security > SSL Certificate and key management**.
   b. Click **Key stores and certificates**.
   c. Keep the default of `SSL keystores` in the dropdown list.
   d. Click **CellDefaultKeyStore** and then **Signer certificates**.
   e. Click **Retrieve from port**.
   f. Enter `<ico_server_fqdn>` as **Host**, 443 as **Port**, `root ca-chain` as **Alias**, and click **Retrieve signer information**.
   g. Check that the displayed information matches your CA authority that signed your server certificates.
   h. Click **OK**.
6. Replace the SSL Certificates in the WebSphere cell default keystore:
   a. Navigate to **Security > SSL Certificate and key management**.
   b. Click **Key stores and certificates**.
   c. Keep the default of `SSL keystores` in the dropdown list.
   d. Click **CellDefaultKeyStore** and then **Personal certificates**.
   e. Select the entry where the **Alias** matches the IBM Cloud Orchestrator Server FQDN of the certificate you requested above and press **Delete**.
   f. Click **Import**.
   g. Choose **Key store file** and enter `/opt/ibm/ico/HTTPServer/bin/key.kdb` for the key file name.
   h. Change **Type** to `CMSKS`.
   i. Enter the keystore password.
   j. Click **Get Key File Aliases**.
   k. Choose the certificate that matches the IBM Cloud Orchestrator Server FQDN from the dropdown list.
   l. Enter `<ico_server_fqdn>` in the **Imported certificate alias** field.
   m. Click **OK**.
   n. **Save** the change direct to the master configuration.
7. Repeat the steps “5” on page 138 and “6” on page 138 for all the existing keystores in your WebSphere environment.
8. Add the new certificate to the `plugin-key.kdb` key database by running the following commands:

```
/opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -import -target \
  /opt/ibm/ico/WebSphere/Plugins/config/ /plugin-key.kdb -target_pw WebAS \
  -db /opt/ibm/ico/HTTPServer/bin/key.kdb -pw -type cms

/opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -setdefault -label \
  -db /opt/ibm/ico/WebSphere/Plugins/config/ /plugin-key.kdb -pw WebAS
```

   where `WebAS` is the default password of the `plugin-key.kdb` key database.

9. Restart the IBM Cloud Orchestrator services by following the procedure in “Starting or stopping IBM Cloud Orchestrator” on page 157.

Installing and updating Process Designer in IBM Business Process Manager
Running the previous procedures updates the keystores with signed certificates and root and intermediate certificates which are used to package the Process Designer installation files.

After updating the certificates, download the Process Designer installation package. After the Process Designer installation, follow the procedure described at Configure Process Designer to access Process Center using Secure Socket Layer (SSL).

Replacing the existing certificates by using automated script

Using this certificate utility, you can replace the existing certificates with either self-signed certificates or CA signed certificates.

About this task

Using the certificate utility, you can do the following tasks:

- Create a self-signed certificate in IHS key.kdb and export the same certificate to the WebSphere® Application Server stores.
- Add a CA signed certificate to IHS key.kdb and export the same certificate to WebSphere Application Server stores.

The certificate utility consists of the following files available at `<installer_path>/installer/tools/` directory on the IBM Cloud Orchestrator server:

- The `replace_ico_certificate.sh` is the main script file.
- The `certificate_property` is the property file that contains all the required parameters.
- The `.sshpass-*.rpm` is the Sshpass installable RPM file format.

To replace existing certificates manually or for background knowledge of certificates, see “Replacing the existing certificates” on page 134.

Procedure

1. Log in to the IBM Cloud Orchestrator Server (in high-availability environment, the primary IBM Cloud Orchestrator Server) as a root user or a non-root user with sudo privileges.
2. Update the `certificate_property` file with all the required parameters.
3. Run `replace_ico_certificate.sh`.

   The syntax of the script is `./replace_ico_certificate.sh <-p property_file> [-s self] [-c ca] [-h]`

   Where

   - **`-p property_file`** is the property file that contains all the required parameters. The `certificate_property` file contains all the required parameters.
   - **`-s self`** is required if a self-signed certificate must be created and configured.
   - **`-c ca`** is required if a CA signed certificate must be added and configured.

   When you use the **`-c ca`** option, do the following prerequisite steps:

   a. Run the following command(on one line) to remove the existing SSL certificate:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -cert -delete -label <ico_server_fqdn>
   -db /opt/ibm/ico/HTTPServer/bin/key.kdb -pw <password>
   ```

   b. Run the following command (on one line) to create the certificate request:

   ```
   /opt/ibm/ico/HTTPServer/java/jre/bin/ikeycmd -certreq -create -label <ico_server_fqdn>
   -dn "CN=<fqdn>,O=<your organization>,OU=<your division>,C=<your country code>"
   -db /opt/ibm/ico/HTTPServer/bin/key.kdb -file certreq_ico.arm -pw <password> -size 2048 -
   sig_alg SHA256WithRSA
   ```

Note: Use /opt/ibm/ico/HTTPServer/bin/gskcmd program in case the ikeycmd does not list CMS. The same program has to be updated in the certificate_property file as:

```
ikeycmd=/opt/ibm/ico/HTTPServer/bin/gskcmd
```

c. In the current directory, locate the certreq_ico.arm file and upload it to your Certificate Authority (CA) for signing.

d. After you receive the certificates from the CA, update the CA signed certificate and the root or intermediate CA certificate location in the property file.

Note: Create a snapshot of the IBM Cloud Orchestrator server along with the certificate requests. If certificate.sh fails, you can revert to the snapshot and can also avoid the loss of certificate request.

e. **-h** shows the usage of the script.

4. Verify the certificate.log to troubleshoot failures or issues.
5. After the replace_ico_certificate.sh script completes successfully, restart the services.

What to do next

After the script is completed successfully, verify whether the IBM Cloud Orchestrator UI can be accessed with the CA certificate and the certificate is added to the WebSphere Application Server stores.

Creating a nonroot user to manage the IBM Cloud Orchestrator Server environment

Create a nonroot admin user, and grant sudo permissions to enable the user to run scripts that require root privileges.

About this task

In this procedure, the example new nonroot admin user is *new_nonroot_admin_user*. Replace this value with the appropriate value for your installation.

Procedure

1. Log on to the IBM Cloud Orchestrator Server as a root user.
2. Create a new user SSH for the IBM Cloud Orchestrator Server:

a) Create a new user *new_nonroot_admin_user* and set the password:

```
useradd -m new_nonroot_admin_user
passwd new_nonroot_admin_user
```

When prompted, enter the password for the *new_nonroot_admin_user* user.

b) Create the .ssh directory and set file permissions:

```
su - new_nonroot_admin_user -c "mkdir .ssh; chmod 700 .ssh"
```

3. Add the user *new_nonroot_admin_user* to the sudo list:

a) Create a sudoer file named *new_nonroot_admin_user* and place it in the /etc/sudoers.d directory.

The content of the *new_nonroot_admin_user* file is as follows:

```
# sudoers additional file for /etc/sudoers.d/
# IMPORTANT: This file must have no ~ or . in its name and file permissions
# must be set to 440!!!

Defaults:new_nonroot_admin_user !requiretty

# scripts found in control script directory

# allow for
new_nonroot_admin_user  ALL = (root) NOPASSWD:/opt/ibm/ico/orchestrator/
```

```
pdcollect/pdcollect.py,  \
(root) NOPASSWD:/opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py
```

b) Change the `sudoer` file permission:

```
chmod 440 /etc/sudoers.d/new_nonroot_admin_user
```

Restricting authentication for specific domains

You can restrict the domains from authenticating to IBM Cloud Orchestrator UI.

You can restrict domains for users who have the privilege to log into IBM Cloud Orchestrator Self service user interface and issue REST calls. Any number of domains can be restricted from authentication.

To restrict domains, add them in a comma-separated list to the `config.json` file in the *domain_list* under *forbidden_domains* option. The file is in JSON format and the default path is `/opt/ibm/ico/wlp/usr/servers/scui/etc/config.json`.

Update the following section of `config.json` to provide a list of domains:

```
"forbidden_domains" : {
  "domain_list" : ["heat"]
 }
```

After you update `config.json`, run **`service scui restart`** to restart the IBM Cloud Orchestrator Self Service user interface.

For IBM Cloud Orchestrator in HA, do the following tasks:

1. Update `config.json` on both the primary and secondary nodes of IBM Cloud Orchestrator.
2. Run the following commands from System Automation Application Manager to restart the IBM Cloud Orchestrator Self service user interface:
   a. Run **`samctrl -M t`** command to change control to manual, and then run **`lssam`** to verify the status of the service.
   b. Run **`service scui restart`** to restart the IBM Cloud Orchestrator Self Service user interface, and then run **`lssam`** to verify the status of the service.
   c. Run **`samctrl -M f`** to change control to auto mode.

To block authentication to IBM Cloud Manager user interface perform the following steps on controller nodes:

1. Open the `/usr/lib/python2.7/site-packages/django/conf/global_settings.py` file in edit mode.

   For IBM Cloud Manager with OpenStack, update the `global_settings.py` on each of the IBM Cloud Manager controllers and run **`systemctl restart httpd`** to restart the httpd service. Do the following tasks for IBM Cloud Orchestrator HA:

   a. Update `global_settings.py` on all the three controllers for each of the regions.
   b. Set the cluster in maintenance mode and run the following command on all controllers to restart httpd service:

      `systemctl restart httpd`
2. Add *BLOCK_LOGIN_HORIZON_DOMAIN* variable after the *TEMPLATE_DEBUG* variable:

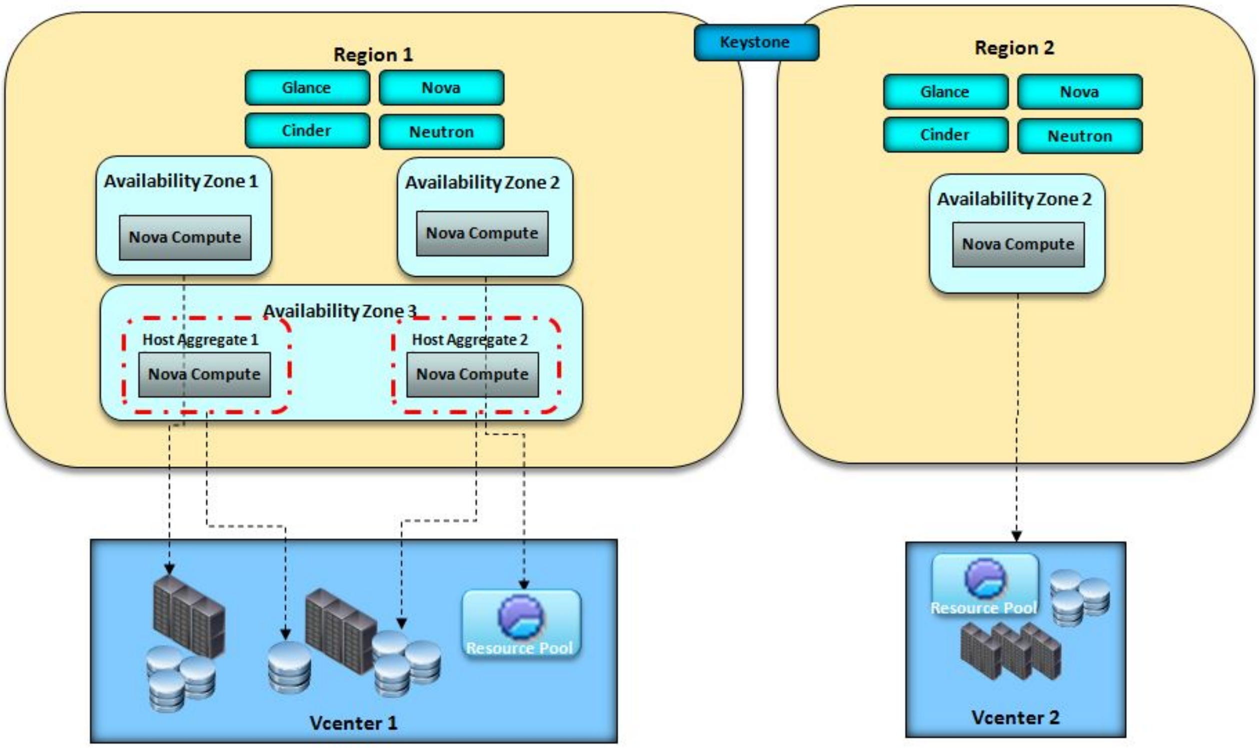
```
######################
# CORE             #
######################
DEBUG = False
TEMPLATE_DEBUG = False
BLOCK_LOGIN_HORIZON_DOMAIN = ['heat']
```

3. Save and close the `global_settings.py` file.
4. Run **service httpd restart** on the controller nodes to restart the httpd service on IBM cloud manager.

Advanced configuration on a VMware region

Before you configure your VMware region, it is important that you understand how OpenStack interacts with the VMware vCenter.

A VMware cluster can be mapped either to an availability zone in OpenStack, or to a host aggregate. The recommended choice is to have a cluster mapped to an availability zone, because this choice reflects the idea of a cluster as an entity with an optimized placement logic and resource grouping. If your VMware cluster has high-availability (HA) and Distributed Resource Scheduler (DRS) policies, they continue to be exploited by VMware itself, because they are transparent to OpenStack. When you define an availability zone or a host aggregate, you can decide the datastore, the datastore clusters, and the resource pool that must be used to store the disk of the images. You can use also regular expressions to easily point to a set of resources.



For example if you want to allow placement in different datastores for the same cluster to exploit different hardware characteristics of the datastores, you can create a single availability zone with multiple host aggregates where each host aggregate points to a different datastore. For more information about how to achieve this configuration, see "Connecting to different datastores in the same cluster" on page 146.

If you want to leverage SDRS, configure your environment by following the procedure described in "Enabling Storage DRS" on page 150. This can be done per availability zone or host aggregate.

Templates and virtual machines are automatically discovered and published in glance and nova after you installed and configured the OpenStack Controller.For more information, see "Configuring vmware-discovery" on page 152. In this way you can immediately manage these templates and virtual machines from the OpenStack Dashboard. You can also view the virtual machines from the Self-service user interface in the RESOURCES panel. Even if these instances were not created by IBM Cloud Orchestrator, you can start, stop, resize them, or run custom actions by using the Self-service user interface. To use the discovered templates as images for deployments in IBM Cloud Orchestrator, you must modify them to meet the prerequisites documented in Chapter 9, "Managing virtual images," on page 239.

If the template was created in thin provisioning mode, all the instances generated from it are thin provisioned. If you want to speed up the cloning operation of instances spawn from the same template, you can turn on the OpenStack linked clone feature. This feature can be set per availability zone or host aggregate and relies on caching the VMDK in the datastore. For more information about this feature, see http://docs.openstack.org/kilo/config-reference/content/vmware.html. Moreover, you can add disks to the image at deployment time or after the deployment occurred. Volumes can be thin provisioned or thick provisioned. For more information, see "Configuring OpenStack to support thin provisioning" on page 151.

Important: In some of these scenarios, you can run more than one Nova compute service that is configured to address the same server cluster. In such a scenario, whenever you start a new Nova compute service or restart an existing service, you must disable the automatic removal of provisioned virtual machines. For the actual procedure to disable, see hypervisor details in the Prerequisites for IBM Cloud Manager with OpenStack section.

VMware administrative user minimum rights

This topic describes the minimum rights for an administrative VMware user.

Table 15. The minimum rights of a VMware administrative user

Privilege name	**Options to be selected**
Data store	• Allocate space • Browse DataStore • Configure datastore • Low level file operations • Move datastore • Remove datastore • Remove file • Rename datastore • Update virtual machine files
Distributed Virtual Port Group	• Create • Modify • Delete
Network	• Assign Network • Configure • Move Network
Resources	• Assign virtual machine to resource pool • Create resource pool • Remove resource pool
Virtual Machine	Select all permissions in this group.
Task	• Create task • Update task
Folder	Delete folder

Table 15. The minimum rights of a VMware administrative user (continued)

Privilege name	Options to be selected
Global	• Act as vCenter • Cancel Task • Disable methods • Enable methods • Licenses • Log Event • Manage custom attribute • Proxy • Script action • Set custom attribute • Settings
Host	Inventory: • Remove cluster • Remove host

Connecting to multiple clusters

When you configure a VMware region, you can connect to multiple clusters defined in the same vCenter. Each cluster is set as a new availability zone in the same region.

Before you begin

Be familiar with VMware support in OpenStack by reading the related OpenStack documentation.

About this task

Because an OpenStack Nova Compute service can connect only to one cluster in OpenStack, you must create a new OpenStack Nova Compute service to connect to your new cluster. The new cluster is set as a new host aggregate in a new availability zone. In this procedure, the following names are used as examples:

`new-cluster-availability-zone`
Is the name of the new availability zone.

`new-cluster-host-aggregate`
Is the name of the new host aggregate.

`nova-vmware.conf`
Is the name of the Nova Compute service configuration file that is related to VMware, This name was specified when the VMware OpenStack Controller was installed.

`openstack-nova-compute-vmware`
Is the name of the Nova Compute service that is related to VMware. This name was specified when the VMware OpenStack Controller was installed.

Important: You must manually start, stop, or restart the newly created services, and must not use knife commands from the Deployment Server of IBM Cloud Manager with OpenStack.

Note:

- You might add more `cluster_name` parameters in the Nova VMware configuration file instead of creating a new host aggregate and a new availability zone, but having two or more clusters in the same availability zone is not a good practice from a virtual machine placement point of view.

- If you want to create several new OpenStack Nova Compute services to connect to different clusters, you can install new services on separate OpenStack Controllers to better manage your resource workload.

Procedure

1. Create the host aggregate and associate it with a new availability zone by running the following command:

   ```
   nova aggregate-create new-cluster-host-aggregate new-cluster-availability-zone
   ```

   This command also creates a new availability zone named `new-cluster-availability-zone`. Now you must create a new OpenStack Nova Compute service.
2. Create a copy of the Nova Compute service configuration file that is related to VMware, for example:

   ```
   cp /etc/nova/nova-vmware.conf /etc/nova/nova-service-new-cluster.conf
   ```

   Change the file ownership by running the following command:

   ```
   chown nova:nova /etc/nova/nova-service-new-cluster.conf
   ```
3. Modify the `/etc/nova/nova-service-new-cluster.conf` file to set:

   ```
   [DEFAULT]
        default_availability_zone = new-cluster-availability-zone
        default_schedule_zone = new-cluster-availability-zone
        storage_availability_zone = new-cluster-availability-zone
        host = new-cluster
        # Use a host name different from the VMware OpenStack Controller to
        # avoid conflict with the first cluster configuration.

   [vmware]
        host_ip = <your vCenter IP address or host name>
        cluster_name = <the name of the new cluster in your vCenter>
   ```
4. Add the host that you specified in step "3" on page 145 for the new Compute service to the host aggregate by running the following command:

   ```
   nova aggregate-add-host new-cluster-host-aggregate new-cluster

   +----+----------------------------+-------------------------------+-------------
   | Id | Name                       | Availability Zone             | Hosts
   +----+----------------------------+-------------------------------+-------------
   | 2  | new-cluster-host-aggregate | new-cluster-availability-zone | new-cluster
   +----+----------------------------+-------------------------------+-------------

   +---------------------------------------------------+
   | Metadata                                          |
   +---------------------------------------------------+
   | 'availability_zone=new-cluster-availability-zone' |
   +---------------------------------------------------+
   ```
5. Create copy of service file in `/usr/lib/systemd/system`.
   For example, run the following command in a single line:

   ```
   cp openstack-nova-compute-vcenter-Cluster1.service
   openstack-nova-compute-vcenter-Cluster2.service
   ```
6. Update the new service file with relevant values for the following parameters:

   - `--config-file` - Points to the new nova cluster configuration file.
   - `--log-file` - Points to the new nova log file.

   Example:

   ```
   [Unit]
   Description=OpenStack Nova Compute Server for vcenter-Cluster2
   After=syslog.target network.target

   [Service]
   ```

```
Environment=LIBGUESTFS_ATTACH_METHOD=appliance
Restart=on-failure
User=nova
ExecStart=/usr/bin/nova-compute --config-file /etc/nova/nova-vcenter-Cluster2.conf
--log-file /var/log/nova/nova-compute-vcenter-Cluster2.log

[Install]
WantedBy=multi-user.target
```

7. Run the following command in a single line to create a link to the new service file.

```
In -s  /usr/lib/systemd/system/openstack-nova-compute-vcenter-Cluster2.service
/etc/systemd/system/multi-user.target.wants/openstack-nova-compute-vcenter-Cluster2.service
```

8. Change the permission on this file to 755.

```
chmod 755 openstack-nova-compute-vcenter-Cluster2.service
```

9. Run the following commands to start the services:

   Under `/usr/lib/systemd/system`:

```
chkconfig openstack-nova-compute-vcenter-Cluster2 on
service  openstack-nova-compute-vcenter-Cluster2 start
service  openstack-nova-compute-vcenter-Cluster2 status
```

10. Verify that the new service is up and running. Run the following command to check whether the new availability zone that is named `new-cluster-availability-zone` is shown:

```
nova availability-zone-list

+---------------------------------------------+--------------------------------------+
| Name                                        | Status                               |
+---------------------------------------------+--------------------------------------+
| internal                                    | available                            |
| |- <your-local-host-name>                   |                                      |
| | |- nova-conductor                         | enabled :-) 2015-05-07T05:15:44.766879 |
| | |- nova-vmware                            | enabled :-) 2015-05-07T05:15:51.017709 |
| | |- nova-consoleauth                       | enabled :-) 2015-05-07T05:15:49.413705 |
| | |- nova-cert                              | enabled :-) 2015-05-07T05:15:47.481551 |
| | |- nova-scheduler                         | enabled :-) 2015-05-07T05:15:47.736521 |
| nova                                        | available                            |
| |- <your-local-host-name>                   |                                      |
| | |- nova-compute                           | enabled :-) 2015-05-07T05:15:43.274219 |
| new-cluster-availability-zone               | available                            |
| |- new-cluster                              |                                      |
| | |- nova-compute                           | enabled :-) 2015-05-07T05:15:44.309888 |
+---------------------------------------------+--------------------------------------+
```

11. For troubleshooting, see the `/var/log/nova/compute-new-cluster.log` file.

What to do next

After you configured your VMware region to connect to multiple clusters, you must run the `vmware-discovery` process as described in "Configuring vmware-discovery" on page 152. In the `vmware-discovery` configuration file, specify the name of your clusters, which are separated by a comma, in the `clusters` parameter. For example:

```
clusters = cluster1,cluster2,new-cluster
```

Connecting to different datastores in the same cluster

You can configure your VMware region to connect to different datastores in the same cluster.

About this task

To connect to different set of datastores in the same cluster, you must create a new OpenStack Nova Compute service for each set of new datastores. The new connection is set as a new host aggregate for each set of datastores. Create a host aggregate for the default datastore and create new host aggregates for each set of new datastores. In the following procedure, the following names are used in the examples:

datastore1-host-aggregate and datastore2-host-aggregate
Are the names of the new host aggregates.

nova-vmware.conf
Is the name of the Nova Compute service configuration file related to VMware, This name was specified when the VMware OpenStack Controller was installed.

openstack-nova-compute-vmware
Is the name of the Nova Compute service related to VMware. This name was specified when the VMware OpenStack Controller was installed.

Procedure

1. Create a host aggregate for each set of datastores that you want to connect by running the following commands, for example:

   ```
   nova aggregate-create datastore1-host-aggregate your-cluster-availability-zone
   nova aggregate-create datastore2-host-aggregate your-cluster-availability-zone
   ```

   where `your-cluster-availability-zone` is the availability zone where the cluster containing the datastores is.
2. Create a copy of the Nova Compute service configuration file related to VMware for each host aggregate that you created, for example:

   ```
   cp /etc/nova/nova-vmware.conf /etc/nova/nova-service-datastore1.conf
   cp /etc/nova/nova-vmware.conf /etc/nova/nova-service-datastore2.conf
   ```

   Change the ownership of the files by running the following commands:

   ```
   chown nova:nova /etc/nova/nova-service-datastore1.conf
   chown nova:nova /etc/nova/nova-service-datastore2.conf
   ```
3. Modify the `/etc/nova/nova-service-datastore1.conf` and `/etc/nova/nova-service-datastore2.conf` files to set:

   ```
   [DEFAULT]
        default_availability_zone = your-cluster-availability-zone
        default_schedule_zone = your-cluster-availability-zone
        storage_availability_zone = your-cluster-availability-zone
        host = <datastore-host>
        # Use a host name different from the VMware OpenStack Controller to
        # avoid conflict with the first cluster configuration.

   [vmware]
        host_ip = <your vCenter IP address or host name>
        cluster_name = <the name of the cluster that contains the datastores in
   your vCenter>

        datastore_regex = <regular_expression_to_identify_the_datastores>
   ```

   where

   <datastore-host>
   Is a different host name in each configuration file. For example, `datastore1-host` and `datastore2-host`.

   datastore_regex
   Is a regular expression that you can use to identify the set of datastores in the cluster to be used during the deployment. If you want to use a specific datastore, specify the datastore name.
4. In scenarios wherein you run multiple Nova compute services that are configured to address the same cluster server, you must disable the automatic removal of provisioned virtual machines. The disable procedure must be done before you start a new Nova compute service or restart an existing service. For more information about disabling the automatic removal, see the hypervisor details in Prerequisites for IBM Cloud Manager with OpenStack section.
5. Create a copy of service file in `/usr/lib/systemd/system`.

Example:

```
cp openstack-nova-compute-vcenter-Cluster1.service openstack-nova-compute-vcenter-
Cluster2.service
```

6. Update the new service file with relevant values for following parameters:

   - `--config-file` - Points to the new nova cluster configuration file.
   - `--log-file` - Points to the new nova log file.

   Example:

```
[Unit]
Description=OpenStack Nova Compute Server for vcenter-Cluster2
After=syslog.target network.target

[Service]
Environment=LIBGUESTFS_ATTACH_METHOD=appliance
Restart=on-failure
User=nova
ExecStart=/usr/bin/nova-compute --config-file /etc/nova/nova-vcenter-Cluster2.conf
--log-file /var/log/nova/nova-compute-vcenter-Cluster2.log

[Install]
WantedBy=multi-user.target
```

7. Run the following command in a single line to create a link of the new service file:

```
In -s /usr/lib/systemd/system/openstack-nova-compute-vcenter-Cluster2.service
/etc/systemd/system/multi-user.target.wants/openstack-nova-compute-vcenter-Cluster2.service
```

8. Run the following command to change the permission on this file to 755:

```
chmod 755 openstack-nova-compute-vcenter-Cluster2.service
```

9. Run the following commands to start the services:

   In `/usr/lib/systemd/system`:

```
chkconfig openstack-nova-compute-vcenter-Cluster2 on
service  openstack-nova-compute-vcenter-Cluster2 start
service  openstack-nova-compute-vcenter-Cluster2 status
```

10. Add the hosts that you specified in the step 3 for the new Compute services to the host aggregates by running the following commands:

```
nova aggregate-add-host datastore1-host-aggregate datastore1-host

nova aggregate-add-host datastore2-host-aggregate datastore2-host

+----+---------------------------+-------------------------------
+----------------
| Id | Name                      | Availability Zone             | Hosts
+----+---------------------------+-------------------------------
+----------------
| 3  | datastore1-host-aggregate  | your-cluster-availability-zone | datastore1-
host
| 4  | datastore2-host-aggregate  | your-cluster-availability-zone | datastore2-
host
+----+---------------------------+-------------------------------
+----------------

+------------------------------------------------------------------------+
| Metadata                                                               |
+------------------------------------------------------------------------+
| 'availability_zone=your-cluster-availability-zone'                     |
| 'availability_zone=your-cluster-availability-zone'                     |
+------------------------------------------------------------------------+
```

11. Set a metadata to the `datastore1-host-aggregate` and `datastore2-host-aggregate` host aggregates that you created in the step 1. For example, run the following commands:

```
nova aggregate-set-metadata datastore1-host-aggregate Datastore1=true

nova aggregate-set-metadata datastore2-host-aggregate Datastore2=true

+----+---------------------------+-------------------------------
+----------------
| Id | Name                      | Availability Zone             | Hosts
+----+---------------------------+-------------------------------
+----------------
| 3  | datastore1-host-aggregate  | your-cluster-availability-zone | datastore1-
host
| 4  | datastore2-host-aggregate  | your-cluster-availability-zone | datastore2-
host
+----+---------------------------+-------------------------------
+----------------

+-----------------------------------------------------------------+
| Metadata                                                        |
+-----------------------------------------------------------------+
| 'Datastore1=true', 'availability_zone=your-cluster-availability-zone' |
| 'Datastore2=true', 'availability_zone=your-cluster-availability-zone' |
+-----------------------------------------------------------------+
```

12. Create new flavors called `flavor-datastore1` and `flavor-datastore2` by running the following commands, for example:

```
nova flavor-create flavor-datastore1 72 4096 40 2
nova flavor-create flavor-datastore2 73 4096 40 2
```

13. Create the flavor keys to match the metadata that you set in the aggregates by running the following commands:

```
nova flavor-key flavor-datastore1 set Datastore1=true
nova flavor-key flavor-datastore2 set Datastore2=true
```

14. In the `/etc/nova/nova.conf` file, set the following parameter on one line without spaces:

```
scheduler_default_filters=AggregateInstanceExtraSpecsFilter,RetryFilter,
       AvailabilityZoneFilter,RamFilter,ComputeFilter,ImagePropertiesFilter
```

Note: The `/etc/nova/nova.conf` file is the Nova configuration file and it is *not* the Nova Compute service configuration file related to VMware that is named `nova-vmware.conf` in this procedure.

Note: Make sure that `ComputeCapabilitiesFilter` value is not set in the `scheduler_default_filters` parameter.

15. Run the following commands to restart the service:

```
chkconfig openstack-nova-compute-datastore1 off
chkconfig openstack-nova-compute-datastore1 on
/etc/init.d/openstack-nova-compute-datastore1 stop
/etc/init.d/openstack-nova-compute-datastore1 start

chkconfig openstack-nova-compute-datastore2 off
chkconfig openstack-nova-compute-datastore2 on
/etc/init.d/openstack-nova-compute-datastore2 stop
/etc/init.d/openstack-nova-compute-datastore2 start

service openstack-nova-scheduler restart
```

Results

You can use the new flavors that you created to deploy to the set of datastores that you specified in the configuration files.

Connecting to resource pools

You can configure your VMware region to connect to a resource pool.

About this task

To connect to a VMware resource pool, you must add the following property under the [vmware] section in the OpenStack Nova Compute service configuration file related to VMware, and restart the related OpenStack Nova Compute service:

```
resource_pool = <cluster_name>:<resource_pool_name>
```

where <cluster_name> is the name of the VMware cluster where the resource pool is defined. When you specify a cluster name and a resource pool name, the resource pool under the cluster is the target to deploy the virtual machines.

If you have multiple resource pools in the same cluster, you can connect to a different resource pool for deployment by creating a new host aggregate with a procedure similar to "Connecting to different datastores in the same cluster" on page 146 and specifying different resource pools with the resource_pool variable in the new OpenStack Nova Compute service configuration file related to VMware.

Enabling Storage DRS

You can enable Storage DRS in your VMware region by editing the nova compute configuration file on the VMware OpenStack Controller.

To support VMware Storage DRS, set the following properties in the [vmware] section in the OpenStack Nova Compute service configuration file related to VMware, and restart the related OpenStack Nova Compute service:

```
datastore_cluster_name = <datastore cluster name>
use_sdrs = True
```

where

datastore_cluster_name
Specifies the name of a VMware datastore cluster (StoragePod name). The default value is None.

use_sdrs
Specifies whether a driver must attempt to call DRS when cloning a virtual machine template. The default value is False.

Note: This feature is only supported when you deploy a virtual machine from template.

You can use the following extra specs of the flavor to override the specified configuration when you deploy new virtual machines:

vmware:datastore_cluster_name
Set this key to override the datastore_cluster_name parameter specified in the Nova Compute service configuration file related to VMware.

vmware:use_sdrs
Set this key to override the use_sdrs parameter specified in the Nova Compute service configuration file related to VMware.

To set the extra specs for the flavor, use the nova flavor-key command.

Enabling datastore random selection

You can enable the datastore random selection when booting a virtual machine.

To randomly select an available datastore when booting a virtual machine, add the following parameter in the [vmware] section in the OpenStack Nova Compute service configuration file related to VMware, and restart the related OpenStack Nova Compute service:

```
random_datastore = True
```

The default value is False.

Configuring OpenStack to support thin provisioning

You can configure OpenStack to support thin provisioning.

You cannot choose the provisioning disk type at deployment time. The provisioning disk type is inherited from the image template:

- If the template is created with a thin-provisioned disk, the instances that are deployed from that template are thin-provisioned.
- If the template is created with a thick-provisioned disk, the instances that are deployed from that template are thick-provisioned.

By default, volumes are created with thin-provisioning.

Adding new disk with thick provisioning type

To add a new disk with thick provisioning type to a virtual machine, perform the following procedure on the OpenStack Controller:

1. Run the following commands as root user:

   ```
   cinder type-create thick_volume
   cinder type-key thick_volume set vmware:vmdk_type=thick
   ```

2. In the /etc/cinder/cinder.conf file, change the following parameters:

   ```
   default_volume_type=thick_volume
   lvm_type=thick_volume
   ```

3. Restart the Cinder services by running the following commands:

   ```
   service openstack-cinder-api restart
   service openstack-cinder-scheduler restart
   service openstack-cinder-volume restart
   ```

Configuring OpenStack to support linked clones

By default, IBM Cloud Orchestrator uses linked clones.

During the virtual machine creation process, the ESX hypervisor requires a copy of the Virtual Machine Disk (VMDK) file for images that do not exist in the VMware environment or were not discovered. As a result, the vCenter OpenStack Compute driver must upload the VMDK file via HTTP from the OpenStack Image Service (Glance) to an ESXi datastore that is visible to the target hypervisor. To optimize this process, the VMDK file is cached on a datastore when the file is used for the first time. Subsequent virtual machines that need the VMDK file use the cached version and do not have to copy the file again from the OpenStack Image service. However, the cached VMDK file must be copied from the cache location to the target datastore. To avoid this copy operation, boot the image in linked_clone mode.

In the OpenStack Nova Compute service configuration file related to VMware, set the **use_linked_clone** parameter to True or False to enable or disable the support of linked clones.

After changing the configuration file, restart the OpenStack Nova Compute service related to VMware by running the following commands, for example:

```
service openstack-nova-compute-vmware stop
service openstack-nova-compute-vmware start
```

where openstack-nova-compute-vmware is the name of the OpenStack Nova Compute service.

Restriction: A linked clone must use an image that is imported into the OpenStack Image service. You cannot create a linked clone from a discovered template.

Important: If you delete a linked-clone virtual machine manually in vCenter, all other linked clones no longer work because the parent disk is deleted from the _base folder.

Tip: It is possible to override the linked_clone mode on a single image basis by using the vmware_linked_clone property in the OpenStack Image Service.

Configuring vmware-discovery

The vmware-discovery process discovers existing virtual machines, templates, and port groups in your VMware environment.

About this task

The vmware-discovery code is installed on the VMware OpenStack Controller. The vmware-discovery configuration file is /etc/nova/vmware-discovery.conf. The vmware-discovery service is nova-discovery.

To configure vmware-discovery, complete the following procedure.

Procedure

1. Enable the VMware driver discovery service described in VMware driver discovery service.
2. Deploy an updated version of the service by running the following commands on each VMware OpenStack Controller:

   ```
   yum -downloadonly -downloaddir=. Openstack-change-tenant
   rpm -ivh -force openstack-change-tenant-<id>.ibm.noarch.rpm
   ```

3. Restart the openstack-nova-api service:

   ```
   service openstack-nova-api restart
   ```

4. Start the vmware-discovery service:

   ```
   service nova-discovery start
   ```

   Note: By default, this service applies to the /etc/nova/nova.conf and /etc/nova/vmware-discovery.conf configuration files. The vCenter information is in the /etc/nova/nova.conf file. You can assign a different configuration file by specifying the --config-file *your_configuration_file* parameter, and you can add this parameter in the /etc/init.d/nova-discovery service script.

   You can find the discovery log in the /var/log/nova/discovery.log file.

   If you want to stop the vmware-discovery service, run the following command:

   ```
   service nova-discovery stop
   ```

5. You can reassign discovered instances to different domains or projects by following the procedure described in “Reassigning VMware instances to a project” on page 185.

Results

You can now manage the discovered resources as described in “Managing resources” on page 211.

Configuring VMware vCenter V6.x region

Prerequisite configuration steps before you create volumes in VMware vCenter V6.x region. In V6.x, the "x" stands for which ever VC 6 version you have.

About this task

If you create volumes on VMware vCenter V6.x region without the prerequisite configuration steps that are mentioned in the procedure, the volume creation fails with the following error message:

```
Unable to update stats, VMwareVcVmdkDriver -1.2.0 driver is uninitialized.
```

The following are the different approaches to configure VMware vCenter V6.x region:

- “Renaming wsdl 5.x directory to 6.x” on page 153

- “Downloading and Creating wsdl configuration file for VMware vCenter V6.x” on page 153

Renaming `wsdl 5.x` directory to 6.x

Procedure

1. Log in to Region Server as a root user.
2. Go to `/usr/lib/python2.7/site-packages/cinder/volume/drivers/vmware/wsdl`. In some cases, it could be `/usr/lib/python2.7/site-packages/oslo_vmware/wsdl`.
3. Rename 5.x to 6.x.
4. Restart the following services:
   - **`service openstack-nova-compute restart`**
   - **`service openstack-nova-api restart`**
   - **`service openstack-nova-scheduler restart`**

Downloading and Creating `wsdl` configuration file for VMware vCenter V6.x

Procedure

1. Create `6.x` folder manually in the path `/usr/lib/python2.7/site-packages/oslo_vmware/wsdl/6.x/`.
2. Download all configuration files for VMware vCenter from https://github.com/openstack/oslo.vmware/tree/master/oslo_vmware/wsdl/6.x and copy them to the path `/usr/lib/python2.7/site-packages/cinder/volume/drivers/vmware/wsdl`.

   Also, copy the files under `/usr/lib/python2.7/site-packages/oslo_vmware/wsdl/6.x/`.
3. Restart the following services:
   - **`service openstack-nova-compute restart`**
   - **`service openstack-nova-api restart`**
   - **`service openstack-nova-scheduler restart`**

Chapter 5. Accessing the IBM Cloud Orchestrator user interfaces

IBM Cloud Orchestrator provides several user interfaces to access the various components.

To display the IBM Cloud Orchestrator user interfaces correctly, use one of the following browsers:

- Internet Explorer versions 10 and 11
- Firefox Extended Support Release 45.8 and later
- Google Chrome version 49 and later

Access to the various IBM Cloud Orchestrator user interfaces depends on the role that is assigned, as shown in the following table:

User interface	URL	Access that is granted to
Self-service user interface	`https://`*`ico_server_fqdn`*`:443`	**admin** role **domain_admin** role **catalogeditor** role **member** role
Business Process Manager user interface	`https://`*`ico_server_fqdn`*`:443/ProcessCenter/login.jsp`	**admin** role only

In the user interface URL, *`ico_server_fqdn`* is the fully qualified domain name (for example, `host.example.com`) of the IBM Cloud Orchestrator Server. If IBM Cloud Orchestrator is installed highly available, use as the `ico_server_fqdn` the fully qualified domain name of the virtual IP.

Note: Do not use the IP address to access the IBM Cloud Orchestrator user interfaces.

You can extend the Self-service user interface URL to include the IBM Cloud Orchestrator domain name, as shown in the following example:

```
https://ico_server_fqdn:443/login?domainName=myDomain
```

In this example, the **Domain** field on the login screen is pre-populated with the value `myDomain`. If you do not specify a domain, the user is authenticated to the **Default** domain.

To log in to the Self-service user interface, you must specify the domain scope to which you want to be authenticated. In a single domain environment or if you do not specify a domain, you are authenticated to the **Default** domain. In addition to the domain-based authentication, when logging in, by default, you are authenticated to scope of the primary project that was specified when your user was created. After successful authentication, you can switch the project from the project list in the top banner of the Self-service user interface.

To log in to the Business Process Manager user interface, you must specify the domain name as a prefix of the user name wherein the delimiter between domain name and user name is a slash (/) character. For example, a user `user1` of domain `domain1` must specify `domain1/user1`. If you are a user who is within the default domain, you must authenticate only with your user name. Users and projects are shown in Business Process Manager as users and groups. Any user or project of custom domains are prefixed with the domain name delimited by a slash (/), that is, project p1 of domain `domain1` appears as a group `domain1/p1` in Business Process Manager. To ensure compatibility with an earlier version, users and projects of the default domain appear with its user and project name. As the user and the project name are prefixed by their domain name in Business Process Manager with a slash (/) as delimiter, the user name, project name, and domain name must not contain a slash (/) character.

Note: In IBM Cloud Orchestrator, user, project and domain names are case sensitive.

For administrative tasks in your OpenStack environment, use the OpenStack Dashboard, which is extended by IBM Cloud Orchestrator, by logging in to the following URL:

```
https://openstack_server_fqdn
```

where *openstack_server_fqdn* is the fully qualified domain name of the OpenStack Controller. The access is granted to the **admin** role only.

By default, the user is also authenticated to the scope of the primary project that you specified when you created the user. After you log in, you can change the project scope by selecting a new project from the project list in the top banner of the user interface. For more information about users, projects, and domains, see "Managing security" on page 166.

Note: In IBM Cloud Orchestrator, the following limitations apply:

- A user name cannot contain a colon (`:`) character.
- A password cannot contain an at sign (`@`) character.
- Users cannot log in if the primary project to which they are assigned is disabled.
- You cannot log in to the same IBM Cloud Orchestrator user interface with more than one browser session on the same machine. If you must log in to the same IBM Cloud Orchestrator user interface with two browser sessions on the same machine, use a different browser for each session. For example, use an Internet Explorer browser and a Firefox browser.

To view the IBM Cloud Orchestrator user interface in another language, set the language option in your browser. Move the preferred language to the top of the list, clear the browser cache, and refresh your browser view. For some browser and operating system combinations, you might need to change the regional settings of your operating system to the locale and language of your choice.

The Self-service user interface is translated in the following languages:

- Brazilian Portuguese
- French
- Germany
- Italian
- Japanese
- Korean
- Russian
- Simplified Chinese
- Spanish
- Traditional Chinese

You must set the locale in Business Process Manager separately. Log in to the Business Process Manager user interface, and click **Preferences**. Select the locale from the **Locale preferences** list, and click **Save changes**. You might need to log in again for the changes to take effect.

Chapter 6. Administering

After you have installed IBM Cloud Orchestrator, you can start your environment, configure optional settings, and define users, projects, and domains.

Starting or stopping IBM Cloud Orchestrator

You can start or stop IBM Cloud Orchestrator by using one of the following procedures depending on your environment.

- "Starting or stopping IBM Cloud Orchestrator in a non high-availability environment" on page 157
- "Starting or stopping IBM Cloud Orchestrator in a high-availability environment" on page 157

Starting or stopping IBM Cloud Orchestrator in a non high-availability environment

To stop IBM Cloud Orchestrator, run the following command as `root` user:

```
/opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --stop
```

When all the services are stopped, you can power off the IBM Cloud Orchestrator Server.

To start IBM Cloud Orchestrator, run the following command as `root` user:

```
/opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --start
```

For more information about the `SCOrchestrator.py` script, see "Managing the services with SCOrchestrator.py" on page 158.

Starting or stopping IBM Cloud Orchestrator in a high-availability environment

To start, stop, or view the status of the IBM Cloud Orchestrator, use the following commands. For more information about these commands, see the IBM Tivoli System Automation for Multiplatforms documentation.

To completely stop the IBM Cloud Orchestrator system:

1. Run the following command on one of the IBM Cloud Orchestrator Servers:

   ```
   chrg -o Offline central-services-rg
   ```

2. Stop your virtual machines in no specific order.

To start the IBM Cloud Orchestrator system:

1. Start all the virtual machines in no particular order.
2. Run the following command on one of the IBM Cloud Orchestrator Servers:

   ```
   chrg -o Online central-services-rg
   ```

To view the status of the IBM Cloud Orchestrator system, run the following command on one of the IBM Cloud Orchestrator Servers:

```
lssam
```

For example, the following command output is displayed:

```
Online IBM.ResourceGroup:central-services-rg Nominal=Online
        |- Online IBM.Application:bpm-node
             |- Online IBM.Application:bpm-node:primaryiconode
             '- Online IBM.Application:bpm-node:secondaryiconode
        |- Online IBM.Application:bpm
             |- Online IBM.Application:bpm:primaryiconode
```

```
             '- Online IBM.Application:bpm:secondaryiconode
        |- Online IBM.Application:ihs
             |- Offline IBM.Application:ihs:primaryiconode
             '- Online IBM.Application:ihs:secondaryiconode
        |- Online IBM.Application:scui
             |- Online IBM.Application:scui:primaryiconode
             '- Online IBM.Application:scui:secondaryiconode
        '- Online IBM.ServiceIP:cs-ip
             |- Offline IBM.ServiceIP:cs-ip:primaryiconode
             '- Online IBM.ServiceIP:cs-ip:secondaryiconode
Online IBM.ResourceGroup:pcg-rg Nominal=Online
        '- Online IBM.Application:pcg
             '- Online IBM.Application:pcg:primaryiconode
Online IBM.Equivalency:cs-network-equ
        |- Online IBM.NetworkInterface:ens192:primaryiconode
        '- Online IBM.NetworkInterface:ens192:secondaryiconode
```

Managing the services

Understand how to manage the IBM Cloud Orchestrator services.

Managing the services with `SCOrchestrator.py`

You can run the `SCOrchestrator.py` script to start, stop, and view the status of the IBM Cloud Orchestrator services in a non high-availability environment.

About this task

The `SCOrchestrator.py` script is located in the `/opt/ibm/ico/orchestrator/scorchestrator` directory in the IBM Cloud Orchestrator Server.

IBM Cloud Orchestrator contains of a number of services and modules, which must be online or running before the product can be used. Because some of these modules and services require being started and stopped in sequence, use the `SCOrchestrator.py` script to start or stop all the IBM Cloud Orchestrator services.

The `SCOrchestrator.py` script uses XML files to obtain the information about the environment and the components:

- `SCOEnvironment.xml`
- `SCOComponents.xml`

The XML files define the names and the start or stop priority of the IBM Cloud Orchestrator services.

The `SCOEnvironment.xml` file is automatically generated by the installation procedure when the IBM Cloud Orchestrator Server is installed.

Note: The `SCOrchestrator.py` script does not manage any of the OpenStack services for the OpenStack distribution that is used by IBM Cloud Orchestrator. For information about how to manage the OpenStack services, see the documentation for your chosen OpenStack product; for example, see the IBM Cloud Manager with OpenStack documentation.

Note: Do not use the `SCOrchestrator.py` script in a high-availability environment. Instead, use the command as documented in “Managing the services in a high-availability environment” on page 159.

For information about how to create a nonroot user that has the required permissions to run this script, see “Creating a nonroot user to manage the IBM Cloud Orchestrator Server environment” on page 140.

Procedure

1. Log on to the IBM Cloud Orchestrator Server.
2. Change to the directory where the script is located:

   ```
   cd /opt/ibm/ico/orchestrator/scorchestrator
   ```

3. Run the script in one of the following ways:

- As a `root` user:

  ```
  ./SCOrchestrator.py --option
  ```

- As a `nonroot` user with sudo permissions:

  ```
  sudo ./SCOrchestrator.py --option
  ```

  or:

  ```
  sudo /opt/ibm/ico/orchestrator/scorchestrator/SCOrchestrator.py --option
  ```

where the *`option`* is one of the following:

- To start the whole product, run `./SCOrchestrator.py --start`.
- To stop the whole product, run `./SCOrchestrator.py --stop`.
- To view the status of components, run `./SCOrchestrator.py --status`.
- To view help for this script, run `./SCOrchestrator.py --help`.

Managing the services in a high-availability environment

You can start and stop services of the IBM Cloud Orchestrator Servers in a high-availability environment.

Procedure

To stop a service, first check the status of the service using the following command:

```
lssam
```

This command displays the actual status of the services, for example:

```
Online IBM.ResourceGroup:central-services-rg Nominal=Online
        |- Online IBM.Application:bpm-node
                |- Online IBM.Application:bpm-node:primaryiconode
                '- Online IBM.Application:bpm-node:secondaryiconode
        |- Online IBM.Application:bpm
                |- Online IBM.Application:bpm:primaryiconode
                '- Online IBM.Application:bpm:secondaryiconode
        |- Online IBM.Application:ihs
                |- Offline IBM.Application:ihs:primaryiconode
                '- Online IBM.Application:ihs:secondaryiconode
        |- Online IBM.Application:scui
                |- Online IBM.Application:scui:primaryiconode
                '- Online IBM.Application:scui:secondaryiconode
        '- Online IBM.ServiceIP:cs-ip
                |- Offline IBM.ServiceIP:cs-ip:primaryiconode
                '- Online IBM.ServiceIP:cs-ip:secondaryiconode
Online IBM.ResourceGroup:pcg-rg Nominal=Online
        '- Online IBM.Application:pcg
                '- Online IBM.Application:pcg:primaryiconode
Online IBM.Equivalency:cs-network-equ
        |- Online IBM.NetworkInterface:ens192:primaryiconode
        '- Online IBM.NetworkInterface:ens192:secondaryiconode
```

You can stop a service by using the following command:

```
rgmbrreq -o stop IBM.Application:<service name>
```

The service is stopped on any nodes in the high-availability cluster.

For example, if the `scui` service is stopped, the following output is displayed when using the **`lssam`** command:

```
Pending online IBM.ResourceGroup:central-services-rg Nominal=Online
        |- Online IBM.Application:bpm-node
                |- Online IBM.Application:bpm-node:primaryiconode
                '- Online IBM.Application:bpm-node:secondaryiconode
        |- Online IBM.Application:bpm
                |- Online IBM.Application:bpm:primaryiconode
```

```
                '- Online IBM.Application:bpm:secondaryiconode
        |- Online IBM.Application:ihs
                |- Offline IBM.Application:ihs:primaryiconode
                '- Online IBM.Application:ihs:secondaryiconode
        |- Offline IBM.Application:scui Request=Offline
                |- Offline IBM.Application:scui:primaryiconode
                '- Offline IBM.Application:scui:secondaryiconode
        '- Online IBM.ServiceIP:cs-ip
                |- Offline IBM.ServiceIP:cs-ip:primaryiconode
                '- Online IBM.ServiceIP:cs-ip:secondaryiconode
Online IBM.ResourceGroup:pcg-rg Nominal=Online
        '- Online IBM.Application:pcg
                '- Online IBM.Application:pcg:primaryiconode
Online IBM.Equivalency:cs-network-equ
        |- Online IBM.NetworkInterface:ens192:primaryiconode
        '- Online IBM.NetworkInterface:ens192:secondaryiconode
```

To start the services again, do not use a start request but cancel the stop request by using the following command:

```
rgmbrreq -o cancel IBM.Application:<service name>
```

Managing the services manually

You can check the status of, start, and stop the services of IBM Cloud Orchestrator.

In a non high-availability environment, to manage services, use the `SCOrchestrator.py` script. Consider managing services manually only if you are an advanced user. For information about using `SCOrchestrator.py`, see “Managing the services with SCOrchestrator.py” on page 158. For information about managing service in a high-availability environment, see “Managing the services in a high-availability environment” on page 159.

The following table illustrates how you can check the status of, start, and stop the services of IBM Cloud Orchestrator.

Components deployed	Command to check the service	Command to start the service	Command to stop the service

DB2	`ps -ef | egrep '^db2';`
	`ss -an | grep 50000 | grep LISTEN`
		`su - db2inst1; db2start`	`su - db2inst1; db2stop`

Business Process Manager	`systemctl status bpm`	`systemctl start bpm`	`systemctl stop bpm`
Self-service user interface	`systemctl status scui`	`systemctl start scui`	`systemctl stop scui`
Public Cloud Gateway	`systemctl status pcg`	`systemctl start pcg`	`systemctl stop pcg`
IBM HTTP Server	`systemctl status ihs`	`systemctl start ihs`	`systemctl stop ihs`
Openstack keystone	`systemctl status openstack-keystone`	`systemctl start openstack-keystone`	`systemctl stop openstack-keystone`

Note: Use **`systemctl`** commands to start, stop, and restart both SCUI and Public Cloud Gateway. Use of other methods may result into unexpected behavior.

If you are using IBM Cloud Manager with OpenStack, see Managing IBM Cloud Manager with OpenStack services for information about checking, starting, and stopping the IBM Cloud Manager with OpenStack services.

Managing settings

You can configure product settings before building a cloud infrastructure.

Before you begin

You must be assigned the **admin** role to manage the product settings.

Customizing the user interface

You can add views to the user interface in IBM Cloud Orchestrator and change the branding for each domain.

Update the user interface by editing the customization metadata properties in the `customizations.json` file. A copy of this file is available in the `<server_location>/etc/customizations/template` directory.

Note: By default, `<server location>` is the `/opt/ibm/ico/wlp/usr/servers/scui` directory. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

This `customizations.json` file can be copied, updated, and added to a specific domain directory under the main customization directory so the user interface can be customized. For example: `<server_location>/etc/customizations/{domainName}`.

Note: The template directory is present so that it can be duplicated and used as an example when you are creating the customization for other domains.

The `customizations.json` file that is in the `<server_location>/etc/customizations/default` directory is used for customization to the default domain. By default, the default domain is available in the `<server_location>/etc/customizations` directory.

The following topics explain the `customizations.json` file, the images folders and how they can be edited to update the user interface that is based on the domain to which you belong.

Branding the user interface per domain

You can change the domain branding and customize the user interface.

Metadata file properties in customization file

The metadata file properties are contained in the `customizations.json` file. This file contains both content and style metadata properties that are used to manipulate the appearance of the IBM Cloud Orchestrator user interface for a particular domain.

The following section explains the metadata file properties:

Content properties

The content properties section of the file contains all the label and static type content that is customized. This content is generally text values and links to replacement images, for example, if you want to use a different logo. The following table shows the values that are contained in the `customizations.json` file:

Content property values	Description
Text values	Items such as the title used in the main page banner and the product title that is used on the login page.
Image values	Items such as the product logo in the main page banner and the product logo that is used on the login page.

Content property values	Description
Content values	Items such as words, sentences, or resource location paths. Content values substitute dynamically replaceable elements in the html template files in IBM Cloud Orchestrator.

Style properties

The style properties section of the file contains all the css style classes, attributes, and elements that are customized. The style properties that are defined must be legal css because it is this metadata that forms the direct input for creating the css properties in the catalog-branding.css file. The catalog-branding.css file is dynamically rendered through the Common Custom Service. The catalog-branding.css is available at <ico_server>:<port>/styles/catalog-branding.css.

If you are logged in to a specific domain, the catalog-branding.css that is used is the one that is specified by the domains customization. If no catalog-branding.css is defined, the default css is used. When all of the properties are defined, the IBM Cloud Orchestrator user interface uses the Common Custom Service to change the look, feel, and style. The style values include items such as the background color used in the main page banner, the font, the logo.

The following example shows what the customizations.json metadata file looks like:

```
{
   "bannerLogo": "/customization/internal/images/IBM_logo.png",
   "bannerLogoAlt": "IBM Cloud Orchestrator",
   "title": "IBM Cloud Orchestrator",
   "bannerBackgroundColor": "#003E68",
   "bodyBackgroundColor": "#F4F6FB",
   "bodyBackgroundImage": "",
   "loginLogo": "/customization/images/black-ibm-logo.png",
   "loginLogoAlt": "IBM Cloud Orchestrator",
   "loginBackgroundColor": "#F4F6FB",
   "loginBackgroundImage": "",
   "loginFormBackgroundColor": "rgba( 0, 0, 255, 0.3)"
   "logoutUrl": "http://www.example.com"
   "requestHistoryLimit": "100"}
```

IBM Cloud Orchestrator supports customizable properties that are defined in the customizations.json file. These supported customizable properties are described in the following table.

Metadata property values	Type
bannerLogo	File system path
bannerLogoAlt	Text
title	Text
bodyBackgroundColor	Text
bodyBackgroundImage	File system path
loginLogo	File system path
loginLogoAlt	Text
loginBackgroundColor	Text
loginBackgroundImage	File system path
loginFormBackgroundColor	Text
logoutUrl	Text
requestHistoryLimit	Text

Metadata property values	Type
Note: • The **bannerLogo** property value can be max 20px high and 1000px wide. • The **loginLogo** property value can be max 30px high and 300px wide. • Although you can enter an unlimited number of characters for the **title** property, only titles with fewer than 50 characters are displayed in the user interface. • For the **bannerLogo** and **bodyBackgroundImage** properties, the image name that is specified must be preceded by `/customization/internal/images/`. For example, if `exampleLogo.png` is the image file to use, then the property value is `/customization/internal/images/exampleLogo.png`. For the **loginLogo** and **loginBackgroundImage** properties, the image name that is specified must be preceded by `/customization/images/`. For example, if `exampleLoginLogo.png` is the image file to use, then the property value is `/customization/images/exampleLoginLogo.png`. The files in the `customizations/<domainName>/images` directory are used for these properties. • The **loginFormBackgroundColor** property value must be in the RGBA format, for example `rgba( 0, 0, 255, 0.3)`, where the fourth numeric value, between `0` and `1`, is the opacity value of the background color of the login panel. This property is especially useful, for example, when used with the **loginBackgroundImage**.	

All base customization files that are used in the IBM Cloud Orchestrator user interface are found in the `<server_location>/etc/customizations/template` directory. For domains that are not customized, the default content that is found in the `<server_location>/etc/customizations/default` directory is used to alter the style and provide the content to the user interface.

Note: The `<server_location>/etc/customizations/template` directory must not be removed, as it is used as an example for all other customizations.

Search not working for request history

Search on request history is client side search as server side search fetches too many records.

We must add the parameter **requestHistoryLimit** in `customization.json`:

requestHistoryLimit

requestHistoryLimit is added in the `customization.json` file. The default value is 20. You can configure as many records as you need based on your requirements. But a minimum of 10 is a must. If it is less than 10, then the default 20 records is used. The maximum is left to you. But too many records might slow down the application.

Example:

```
requestHistoryLimit : 50
```

Customizing the user interface

You can update the IBM Cloud Orchestrator user interface for a given domain.

Procedure

1. The customization metadata is stored in the `customizations.json` file in a specific domain directory under the main custom directory. Create a directory under the `<server_location>/etc/customizations` directory with the name of the domain that you want to customize. For example: `<server_location>/etc/customizations/<domain_name>`.

   Note: By default, `<server location>` is the `/opt/ibm/ico/wlp/usr/servers/scui` directory.

   Note: Branding changes are only visible to the domain they are declared for, except the `Default` domain.

2. Copy the `customizations.json` file from the `<server_location>/etc/customizations/template` directory to the directory that you created in the step "1" on page 163, and edit the file to customize it.

   Note: All the files in the customization directory must be owned by the `scui` user.
3. Create a folder that is called `images` in the directory that you created in the step "1" on page 163. For example: `<server_location>/etc/customizations/<domain_name>/images`.
4. Copy the image file that is referenced in the content section of the `customizations.json` file to the `customizations/<domain_name>/images` directory. This method is used to ensure that all domain-specific customization content is in the correct location.

   Note: You can also add and create new images and add them to the images folder. However, you must update the `customizations.json` file depending on any new image file that you add.
5. Restart the `scui` service and clear the custom cache for the changes to be picked up. Restart the service by running the following command:

   ```
   service scui restart
   ```

   Enter the following link in your browser to clear the server cache:

   ```
   http://<ico_server>:<port>/customization/clearcustomcache
   ```
6. Log in to the IBM Cloud Orchestrator user interface with the domain that was customized and view the results to ensure that you are satisfied with the style updates. To open the login screen for a specific domain, for example `NewDomain`, enter the following link in your browser:

   ```
   http://<ico_server>:<port>/login?domainName=NewDomain
   ```

   Note: If an error is displayed in the user interface, it indicates that there is a problem with the customization. Perform the following tasks:

   - View the server logs to find more information about the error: `/opt/ibm/ico/wlp/usr/servers/scui/logs/scoui.log` and `/opt/ibm/ico/wlp/usr/servers/scui/logs/scoui.trc`.
   - Ensure that the domain directory is correctly named, the `customizations.json` file is in the correct format, and its contents are correct.

Customizing the OpenStack Dashboard

This topic describes how to customize the OpenStack Dashboard.

About this task

For information about customizing the OpenStack Dashboard in IBM Cloud Manager with OpenStack, see Rebranding the dashboard and self-service user interface.

For information about customizing the OpenStack Dashboard in an OpenStack distribution of another vendor, see the OpenStack documentation.

Dashboard extensions

A dashboard extension file is a single html file that displays one or more Business Process Manager coaches. For the dashboard extension these coaches must contain dashboards, graphs or other reporting elements that are developed in Business Process Manager.

The user-defined extension html file contains a fragment of html without a head or a body element. These are included in the parent html file that embeds the supplied extension content.

The html file must contain `iframe` tags to specify the Business Process Manager dashboard coaches that you want to display on the page. In this case, an `iframe` is used to embed one html document, the dashboard coach, within another html document, the parent extension. Define the width and height of the inline coach frames to achieve the layout you want.

The following snippet of code is an example of a user-defined html fragment with a Business Process Manager dashboard iframe that is contained in a sample extension html file:

```
<div align="center">
 <iframe id="ifm" name="ifm" width="1000" height="1000" frameborder="0"
 src="{{bpmEndpoint}}/teamworks/process.lsw?
zWorkflowState=5&zProcessRef=/1.fc983f33-e98f-4999-b0b5-
bd1e39d6102e&zBaseContext=2064.abec322d-430c-43dd-820a-
98f223d29fa4T&applicationInstanceId=guid:6416d37cd5f92c3a:33127541:1461a68c1cd:-
7ffe&applicationId=2"
scrolling="auto" align="middle">
 </iframe>
</div>
```

In the example there is the following mustache variable defined {{bpmEndpoint}}. This utility variable is supplied by the extension framework. It enables extension content providers to locate and specify the base Business Process Manager host server url and port of an installed IBM Cloud Orchestrator instance in a generic way. Having this variable available to extension deployers means that no manual editing must be made to the html file after deployment to specify the location of the Business Process Manager server. For more information related to mustache see http://mustache.github.io/mustache.5.html.

Single sign-on is enabled between the Self-service user interface and Business Process Manager, so displaying Business Process Manager coaches in the user interface does not require any additional authentication configuration.

Role-based directory structure

Dashboards are role-based to allow you, as Service Designer, to have your dashboard extension content available to specific roles.

The directory structure that the extension content is deployed into is based on the IBM Cloud Orchestrator roles listed in the following table. The IBM Cloud Orchestrator roles that the extension is compatible with are:

Name of role directory	Role
member	End User
admin	Cloud Administrator
domain_admin	Domain Administrator
catalogeditor	Service Designer

The <server_location>/etc/dashboard folder is the parent extension directory. It contains a folder for each of the roles. For example, <server_location>/etc/dashboard/admin or <server_location>/etc/dashboard/member.

When you want to make a new dashboard available to an admin user, for example, add the extension html file to the <server_location>/etc/dashboard/admin directory.

Navigation elements and extension naming conventions

Naming extension files is important as these names drive the names of the navigation elements where the dashboards are accessed from.

Navigation elements must be meaningful so that you can easily locate important content in the UI. As these dashboard extension navigation elements are driven from the names of the dashboard extension files, these file names must be meaningful. Dashboard extensions appear as submenu items under the **DASHBOARD** menu and take the name of the extension file as their menu label. The label is the complete file name with any ordinal position specified and the file extension removed.

The following table describes an example of file names and their respective menu labels:

File	Menu label
01 - Admin Extension Example.html	Admin Extension Example

File	Menu label
02 - Member Extension Example.html	Member Extension Example
Network Dashboard.html	Network Dashboard
Performance Dashboard.html	Performance Dashboard

Ordinal numbering of extension files is included so that you can control the order in which the extension labels appear in the **DASHBOARD** menu. The ordinal numbering format convention is a number, then a hyphen (-), then the file name. Any file name starting with this pattern is placed in this relative numbered position in the sub menu. The pattern is stripped from the file name when constructing the label. If an ordinal numbering format convention is not used when naming extension files the files is added alphabetically.

Packaging and deployment

A dashboard extension must be packaged as a compressed file or as `tar.gz` file.

The package structure must match the role-based directory structure that is mentioned in the previous section. For example:

```
extensionExample.zip
        + dashboard
            + admin
                + 01 - My Admin Extension Example.html
```

To deploy the extension, extract the compressed file to `<server_location>/etc`. To have the new extension content appear in the user interface, restart the IBM Cloud Orchestrator server. Alternatively, enter the following link in your browser to clear the navigation cache:

```
<ico_server>:<port>/dashboardextension/clearcache
```

Note: Before deploying a dashboard extension ensure that there are no extension files that have already been deployed previously in the extension directories with the same name. Any deployed file with the same name as one file that is contained in the package gets overwritten otherwise.

Managing security

You can manage users and the level of access that each user has in the IBM Cloud Orchestrator environment. You can assign which roles a user has on a specific project, as well as the primary project for a user. A user can have different roles on different projects. Both users and projects belong to a domain.

About this task

The OpenStack Compute system is used by many different cloud-computing customers, basically projects on a shared system, by using role-based access assignments. Roles control the actions that a user is allowed to perform. A user's access to particular images is limited by project, but the user name and password are assigned per user. Key pairs granting access to an instance are enabled per user, but quotas to control resource consumption across available hardware resources are per project.

A *project* is an isolated resource container forming the principal organizational structure within the OpenStack environment. A *domain* represents a customer (that is, for example an organization or a division) and the related resources as a segregated entity. Only users within that domain have access to these resources. For information about users, projects, domains, and users, see “Model and terminology” on page 167.

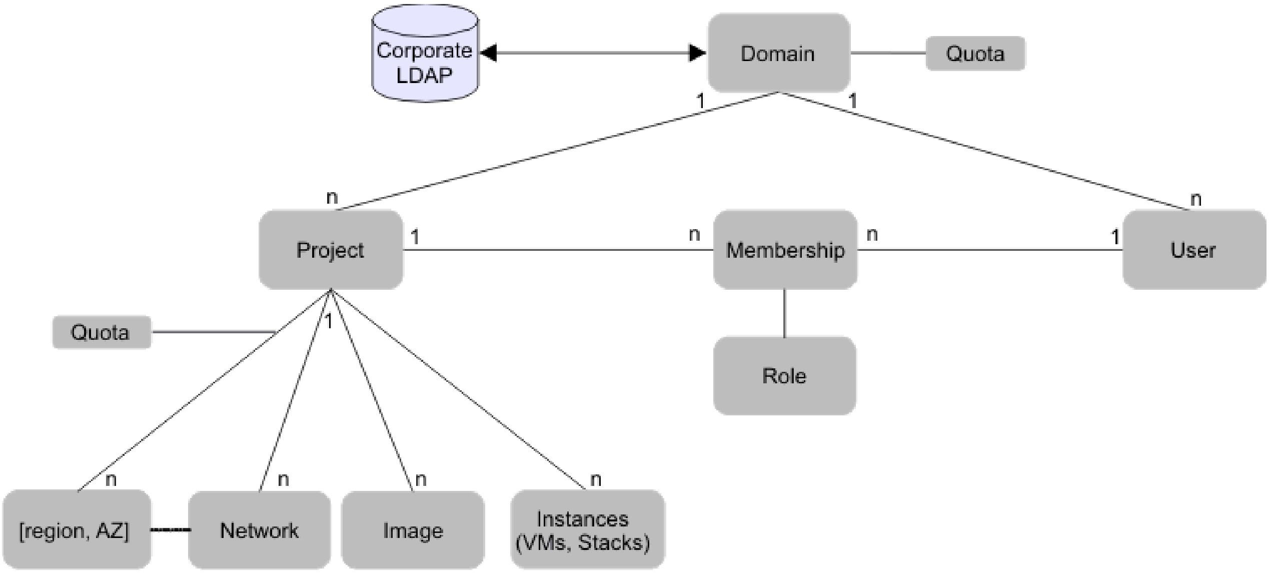
When you create users and projects in IBM Cloud Orchestrator, they are being created in the underlying OpenStack environment. The roles that are defined in the OpenStack environment are used in IBM Cloud Orchestrator as described in “User roles in IBM Cloud Orchestrator ” on page 168.

If you are using an LDAP server to authenticate users, you can configure LDAP authentication to allow any corporate directory details to be specified on a domain-by-domain basis. For more information, see "Configuring LDAP authentication" on page 117.

You can work with your users, projects, and domains with the Self-service user interface or with the OpenStack Dashboard depending on your role. For more information, see "Administering as cloud administrator" on page 172 or "Administering as domain administrator" on page 190.

Model and terminology

The multitenancy model is based on the OpenStack identity service version 3. This is implemented by Keystone. This includes the following entities and relationships.



Domain

A domain is the highest entity in the identity model and represents a tenant as it is a container and namespace for projects and users of a customer. IBM Cloud Orchestrator allows segregation on both the domain level and the project level. Whether the domain concept is used depends if the tenant must be allowed to organize itself and requires the role of a domain administrator. If the domain is a self-organized unit, create a domain and its domain administrator. A domain can have multiple projects and users. The project and users are owned by a domain. The domain administrator can manage the project and users and assign resources to them. If the customer is not a self-organized unit and the administrator of the service provider configures all projects, users and resources, the domain concept can be ignored and the **Default** domain can be used. The **Default** domain always exists.

User

A user represents the account of a person. You can log in to IBM Cloud Orchestrator with a user account. A user account contains:

- user name
- password
- email address

A user is unique within a domain. You can have two different users with the same name in two different domains. A user must always be member of at least one project and have a default project defined.

Project

A project is a container that owns resources. Resources can be:

- virtual machines
- stacks
- images

- networks
- volumes

The project is unique within a domain. This means you can have two projects with the same name in two different domains. A project can also have one or more users as members. With a membership, you can access all resources that are owned by the project, so if you are a member of a project, you can access all resources that are owned by that project. For more information about OpenStack users and projects, see the OpenStack documentation.

For information about virtual machines, see "Managing virtual machines" on page 213. For information about stacks, see "Working with Heat templates and stacks" on page 222. For information about images, see "Adding images to your OpenStack environment" on page 243. For information about networks, see "Managing networks" on page 189. For information about volumes, see "Working with volumes" on page 225.

Role

A role grants that you access to a set of management actions. IBM Cloud Orchestrator supports the following different roles:

- `admin`
- `domain_admin`
- `catalogeditor`
- `member`

For information about the roles, see "User roles in IBM Cloud Orchestrator " on page 168.

Scope

You can be member of one or multiple projects. As a user, you always work in the scope of a project. When you log in, you work on-behalf-of a default project. If you are a member of multiple projects, you can switch across projects in the self-service banner.

LDAP

IBM Cloud Orchestrator can be configured to authenticate users with an LDAP or Active Directory. It is allowed to configure one LDAP for all domains or a specific LDAP per domain. If you log in to a domain with a LDAP configured you are authenticated against the LDAP of that domain.

User roles in IBM Cloud Orchestrator

Protect your cloud environment by using roles to control how different users interact with IBM Cloud Orchestrator. When you assign roles to users, you designate the types of objects that the users can access, and the tasks that the users can perform.

In IBM Cloud Orchestrator, the users and the projects are defined in the related OpenStack environment. When you create a user, you must assign a role to the user. The role is related to the project to which the user belongs. A user can have one or many roles within a project, and can have different roles in different projects.

The authority to access a type of object might not equate with the authority to access all instances of that object. In some cases, users can access an object instance only if they have been granted authority by the creator of that instance.

In IBM Cloud Orchestrator, you can use the following roles:

admin role

A user with this role can do the following tasks in the IBM Cloud Orchestrator environment:

- Create domains, projects, users, and roles.
- Assign other user roles. Configure product settings.
- Grant users access to projects. Grant projects access to regions and availability zones. Assign quotas to domains.

- Do all of the tasks that a user with the **domain_admin** role can do.

Important: A user with the **admin** role has full privileges for all cloud resources, including all projects and all domains. Assign this role only to the cloud administrators, and only to users within the `Default` domain and `admin` project.

domain_admin role

A user with this role can do the following tasks in the IBM Cloud Orchestrator environment:

- View the details of the domain.
- View the projects, users, groups, offerings, and actions of the domain.
- Create, edit, and delete projects, users, groups, offerings, and actions that are associated with the domain.
- Manage the quota, availability zones, and networks for projects in the domain.
- Do all of the tasks that a user with the **catalogeditor** role can do.

catalogeditor role

A user with this role can do the following tasks in the IBM Cloud Orchestrator environment:

- Create self-service offerings and other artifacts in the self-service catalog. Modify or delete any self-service offerings and other artifacts in the self-service catalog that they create or to which they have access. Do all of the tasks that a user with the member role can do.
- Create images in the OpenStack environment.
- Do all of the tasks that a user with the **member** role can do.

member role

A user with this role can do the following tasks in the IBM Cloud Orchestrator environment:

- View and work with the catalog content to which they are granted access.
- Manage the resources (for example, stacks, virtual machines, and volumes) to which they are granted access.

Note:

The following roles, which are shown in the OpenStack Dashboard, are used only for the OpenStack and not for IBM Cloud Orchestrator:

```
_member_
KeystoneAdmin
KeystoneServiceAdmin
sysadmin
netadmin
```

For any other user-defined or custom user roles other than IBM Cloud Orchestrator supported roles, you can log in as a user with access equivalent to "member" role. However, you can configure these users in ACL to access specific IBM Cloud Orchestrator offerings.

For OpenStack commands to list users, roles and role assignments, and assign roles to users, see https://docs.openstack.org/admin-guide/cli-manage-projects-users-and-roles.html.

Related tasks

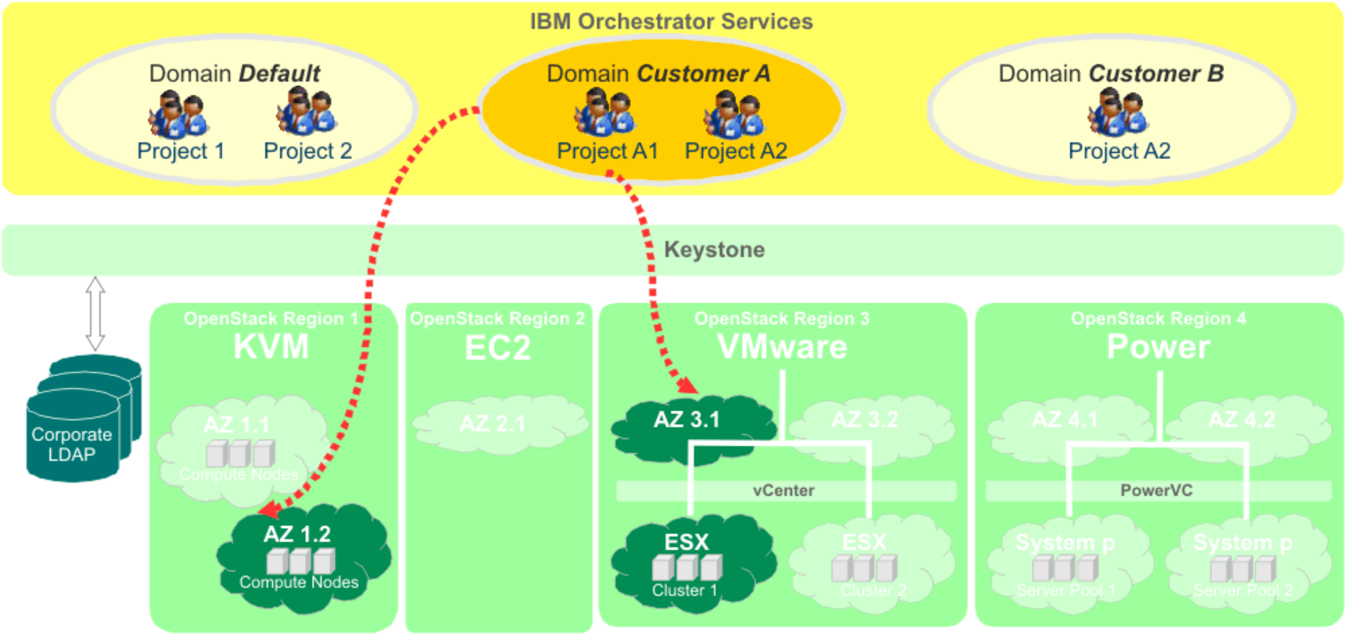
“Modifying the access control list of an action” on page 231
You can modify the access control list of an action by adding or removing access.

“Modifying the access control list of an offering” on page 229

You can modify the access control list of an offering by adding or removing access.

Regions, availability zones and quota

This topic describes regions, availability zones, and quota and their relationship in IBM Cloud Orchestrator.



A region is defined by a set of OpenStack services operating on the same hypervisor. For example, a region can manage a set of KVM hosts or a VMware vCenter environment. It is not possible to manage different types of hypervisors within the same region.

An availability zone is a subset of compute resources within a region. A region can have many availability zones, but an availability zone can belong to only one region. Availability zones are used as the target for deployment and as a user, you must select the availability zone to place a virtual machine or stack.

IBM Cloud Orchestrator allows access of availability zones to Domains and Projects to allow an End User to manage them.

It is possible to define a quota to limit the allocation of resources in the cloud. The following definitions describe the different types of quota:

Domain Quota

The sum of all project quota must not exceed the domain quota. The domain quota is only enforced if the Domain Administrator creates a new project or modifies the quota of an existing project. If the domain quota is exceeded, the Domain Administrator is not allowed to create the project or increase the quota. The quota is not enforced for a Cloud Administrator.

Project Quota

The project quota is defined per region. In other words, the project can have different quota in different regions. Both a Cloud Administrator and a Domain Administrator can change the quota of a project in a region. The quota the Domain Administrator cannot exceed the overall domain quota while it is changing.

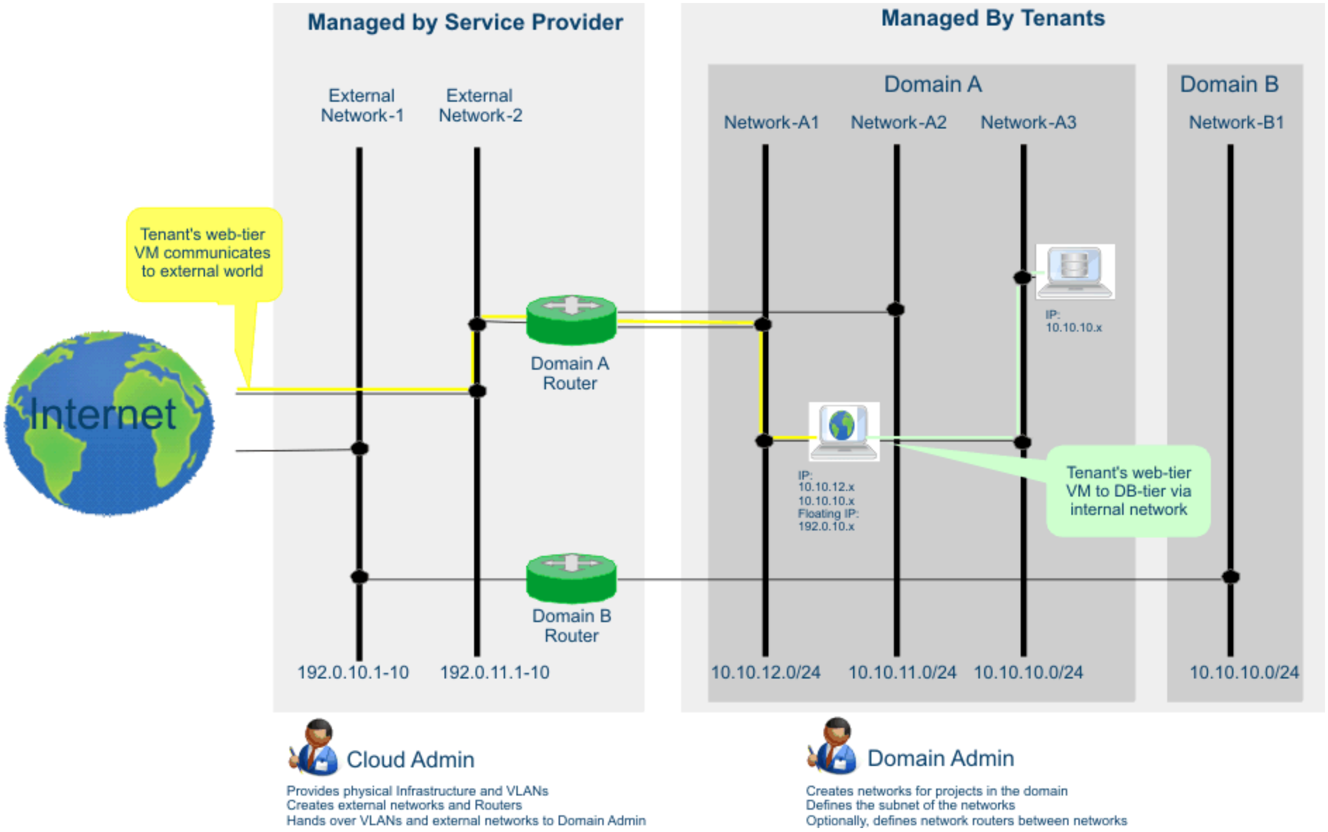
Default Quotas

Default quotas can be configured which are set for each new project. The domain default quota is a multiplication of a factor of the default project quota. Both default settings apply across domain and can only be configured by the cloud administrator.

If a user requests a service that allocates resources in the cloud, only the quota its current project is checked and enforced. If the request exceeds the quota in that region, the request fails with an error message. It is ensured by definition, that the domain quota is not exceeded, as the Domain Administrator cannot give more quota to its projects. Therefore, only check the project quota during deployment.

Network isolation

The segregation of tenants also requires the isolation of tenant networks.


Managed by Service Provider
Managed By Tenants
External Network-1
External Network-2
Domain A
Network-A1
Network-A2
Network-A3
Domain B
Network-B1
Tenant's web-tier VM communicates to external world
Internet
Domain A Router
IP: 10.10.10.x
IP: 10.10.12.x 10.10.10.x Floating IP: 192.0.10.x
Tenant's web-tier VM to DB-tier via internal network
Domain B Router
192.0.10.1-10
192.0.11.1-10
10.10.12.0/24
10.10.11.0/24
10.10.10.0/24
10.10.10.0/24
Cloud Admin
Provides physical Infrastructure and VLANs
Creates external networks and Routers
Hands over VLANs and external networks to Domain Admin
Domain Admin
Creates networks for projects in the domain
Defines the subnet of the networks
Optionally, defines network routers between networks

IBM Cloud Orchestrator supports the following technologies to manage the networks within the cloud:

- OpenStack Nova network service
- OpenStack Neutron network service

Details of the services can be found in the OpenStack documentation.

From a multitenancy perspective, the Neutron service provides the best capabilities, which allow tenants to manage their network topology in separate namespaces. This includes, overlapping IP scenarios, where two tenants can have two separate networks each having the same overlapping IP subrange. The isolation and separation is handled the Neutron service.

The Cloud Administrator is responsible to manage the network topology. It is the service provider and Cloud Administrator responsibility to provide the external connectivity to the tenants.

In IBM Cloud Orchestrator, network isolation happens on the project layer, not on the domain layer. A network is typically owned by a project, and a project can have multiple networks.

The Domain Administrator can create private networks for projects within its domain.

The End User, who requests virtual machines on behalf of the current project can add one or more network interfaces. This ensures connectivity to one or more networks of the project.

This means that a tenant can manage their internal connectivity and can define a multitier application environment with different networks for database and application traffic.

However, the external connectivity must be ensured by the Cloud Administrator.

Setting up PowerVC RSA keys

To inject a wanted SSH key into a Linux on Power Systems virtual machine at deployment time the following steps must be done to ensure that the keys are available for use by both PowerVC and IBM Cloud Orchestrator.

Procedure

1. Generate the Public and Private Key on the PowerVC server:

   ```
   ssh-keygen -t rsa -f test1.key
   ```

   This creates a `test1.key` and a `test1.key.pub` that are the private and public keys respectively.
2. Add a new key pair to the PowerVC key pair list ensuring that you select the public key that was created in step 1:

   ```
   nova keypair-add --pub-key test1.key.pub test1
   ```

3. Ensure that the key pair is available on the PowerVC server:

   ```
   nova keypair-list
   +-------+-------------------------------------------------+
   | Name  | Fingerprint                                     |
   +-------+-------------------------------------------------+
   | test1 | 63:fd:a8:7e:50:31:b6:f9:ec:14:5d:e9:a6:ae:e1:e9 |
   +-------+-------------------------------------------------+
   ```

4. Through the OpenStack Dashboard or any other method you are familiar with, create a key pair on IBM Cloud Orchestrator and specify the same name as the key pair on PowerVC and ensure that you also specify the contents of `test1.key.pub` as the public key.

Results

You are now able to deploy a Linux on Power Systems virtual machine from IBM Cloud Orchestrator that can use the private key `test1.key` for access.

Administering as cloud administrator

A common scenario for a Cloud Administrator is the onboarding of a customer or organization, which must have the required administrator roles and cloud resources assigned to be up and running.

For more information, see "Managing a domain from OpenStack Dashboard" on page 176.

Note: The OpenStack Dashboard does not support filtering with special characters like, for example, $ ^ ( ) + . [\ ? | %

Managing domains

You can manage domains from both IBM Cloud Orchestrator Self Service UI or OpenStack Dashboard.

Managing domains from IBM Cloud Orchestrator Self Service UI

From IBM Cloud Orchestrator Self Service UI, you can now manage domains.

Creating a domain

As a Cloud Administrator, you can now create new domains from the IBM Cloud Orchestrator Self Service UI.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.
3. Select **Create Domain** from the **Actions** menu.
4. Enter the domain name and description.
5. Select **Enabled**.

6. Click **Ok**.

Creating a project for a domain
You can assign individual zones to each domain in IBM Cloud Orchestrator from the IBM Cloud Orchestrator Self Service UI.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain** and select a domain.
3. Select **Create Project** from the **Actions** menu.

   The **Define Project** window is displayed.
4. Specify the name for the project.
5. Optional: Enter a description for the project
6. Optional: By clearing the **Enabled** check box, you disable and cannot authorize the domain. Selecting the **Enabled** check box keeps the domain enabled so that you can authorize the domain.
7. Click **Ok**.

   A message indicates that the project is created successfully.

Results
The new project is created for the selected domain.

Creating user for a domain
You can manage the level of access for each user in IBM Cloud Orchestrator. Users can be assigned to different roles on different projects.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain** and select a domain.
3. Select **Create User** from the **Actions** menu.

   The **Create User** window is displayed.
4. Specify the required parameters, and then click **Ok**.

   A message indicates that the user is created successfully.

Results
A new user is created for the specified domain.

Creating groups
From the IBM Cloud Orchestrator Self Service UI, you can create groups to organize users and roles.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.
3. Select a domain.
4. From the **Actions** menu, select **Create Group**
5. In the **Define Group** page, enter the name of the group and its description.
6. Click **Ok**.

   A message appears to confirm that a new group is created successfully.

Setting a domain context
The domain context is for Cloud administrators to refine the context that they are accessing. Cloud administrators can limit the scope to one domain, rather than having visibility across all domains. This

allows the Cloud administrator to identify the projects, users, groups, and roles that are associated with a domain.

Procedure

1. Log in to the IBM Cloud Orchestrator Self-service user interface as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.
3. In the domains page, select the entry for the domain and click **Set Domain Context** from the **Actions** menu.

Results

The **Configuration > Domain** page displays only the domain that is set in the context. Selecting the **Projects**, **Users**, **Groups**, or **Roles** web page displays details for the selected domain context.

Note: The domain context is applicable to the **Configuration -> Domain** menu to manage domains within the IBM Cloud Orchestrator Self-service user interface. This setting context is not applicable to other resources and offerings, as they are not just domain scoped but also project and user scoped.

Clearing the domain context
Cloud administrators can clear the scope of all domains, enabling visibility across all domains.

Procedure

1. Log in to the IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration -> Domain**.
3. In the domains page, click **Clear Domain Context** from **Actions** menu.

Results

All domains are visible.

Modifying Domain Administrators
You can create a new user and assign the role of a domain administrator or you can assign the domain administrator role to an existing user.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.
3. Select a domain and set the domain context using "Setting a domain context" on page 173.
4. Select a domain and select the **Modify Domain Administrators** from the **Actions** list.
5. In the **Modify Domain Administrators** page, add or remove users from the list of domain administrators.
6. Click **Ok**.

Modifying quota
The Cloud Administrator can change the quotas of a domain to set limits on the operational resources that a Domain administrator can distribute among all the projects in the domain. You must set the default quotas for the "Default" domain.

Procedure

1. Log in to IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Select `Default` domain and click **Modify Quota** from the **Actions** menu.
3. In the **Modify Quota** page, you can change or confirm the existing values.
4. Click **OK**.

Setting the default domain quotas

The Cloud Administrator sets the default domain quota for certain resources to specify the maximum amount of the resource that is available to the domain. The Domain Administrator can then distribute that quantity among all the projects in the domain.

Before you begin

You must be assigned the **admin** role to complete these steps.

About this task

The Cloud Administrator can assign a default *project quota* to certain resources, as described in “Configuring project quotas” on page 185.

To generate the default *domain quota* values, IBM Cloud Orchestrator multiplies the corresponding default project quota value by the **`projects_per_domain`** variable.

Note: The **`projects_per_domain`** variable is a multiplier that IBM Cloud Orchestrator applies to the default project quotas, to calculate the default domain quotas. The **`projects_per_domain`** variable does *not* specify the maximum number of projects that can be created in a domain.

The default value of the **`projects_per_domain`** variable is 5. The Cloud Administrator can change the value of the **`projects_per_domain`** variable, which consequently changes the default domain quotas, as follows:

Procedure

1. Log into IBM Cloud Orchestrator server as a root user.
2. Go to `/opt/ibm/ico/wlp/usr/servers/scui/etc/`.

   Note: The `/opt/ibm/ico/wlp/usr/servers/scui` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.
3. Add the following lines to the `local_settings.json` file:

   ```
   SCO_CONFIG = {

       'projects_per_domain': <number of projects>,

   }
   ```

   Note: Create `local_settings.json` file if it is not available.
4. Restart SCUI server and Business Process Manager server.

   ```
   service scui restart
   systemctl restart bpm
   ```

Modifying Availability Zones

After you set the quota for a domain, modify the Availability Zone of the Domain and then modify the Availability Zone of the project "admin".

Before you begin

When you install IBM Cloud Orchestrator, a default domain is created in IBM Cloud Manager with OpenStack. However, no default quota is set for the domain. For such domains that are created in IBM Cloud Manager with OpenStack or other OpenStack, set or modify quota first and then modify the Availability Zone of the domain.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.

3. Select a domain and set the domain context by using "Setting a domain context" on page 173.
4. From the **Actions** list, select the **Modify Availablity Zone**.
5. In the **Availability Zone** page, add or remove zones for the domain and click **Ok**.
6. Repeat steps 2 and 3.
7. Select **Projects** from the navigation and select the project from the available projects list.
8. In the **Availability Zone** page, add or remove zones for the project.
9. Click **Ok**.

Disabling or enabling a domain
You can disable an existing domain.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.
3. Select a domain and click **Disable Domain** from the **Actions** menu.
4. To enable a domain, select the disabled domain and from the **Actions** menu, click **Enable Domain**. The Enable Domain option is available in the **Actions** menu only for disabled domains.

Deleting a domain
You can delete a domain only after you disable it.

Procedure

1. Log into IBM Cloud Orchestrator Self Service UI as a Cloud Administrator.
2. Go to **Configuration** > **Domain**.
3. Select the disabled domain and from the **Actions** menu select **Delete Domain**. The **Delete Domain** option is available in the **Actions** menu only for disabled domains.
4. Confirm the deletion of the domain.

Managing a domain from OpenStack Dashboard

You can manage domains in IBM Cloud Orchestrator with the OpenStack Dashboard.

About this task

Domains represent a customer or an organization in a multi-tenant environment. Perform the following steps for on-boarding a customer in this environment. You must be assigned the **admin** role to perform this procedure.

Procedure

1. Create a domain resource. This step automatically creates a default project for the domain to facilitate user on-boarding.
2. Ensure that the domain has access to at least one deployment availability zone. This allows users in that domain to access virtual images and deploy virtual servers when logged in to the domain projects. The availability zones that are assigned to the domain are then visible to be assigned to projects within the domain.
3. To delegate the domain administration, ensure that at least one user is assigned to the domain with **domain_admin** role. With this role, the Cloud Administrator can delegate the administrative tasks of the domain to the Domain Administrator who can then start creating projects and assigning users.

   Note: The default OpenStack Cloud Administration domain **Default** must not be disabled. If it is disable, you are unable to log in to the Self-service user interface and to the OpenStack Dashboard as the default Cloud Administrator. If you disabled the domain, you can enable it again in one of the following ways:

   - Send an HTTP request as follows:

```
curl -i -X PATCH http://<HOST>:35357/v3/domains/default
-H "User-Agent: python-keystoneclient"
-H "Content-Type: application/json" -H "X-Auth-Token:<TOKEN>"
-d '{"domain": {"enabled": true, "id": "default", "name": "Default"}}'
```

- Update the domain to be enabled with the Python client by using V3 Keystone.

Creating a domain
The Cloud Administrator creates domains to organize projects, groups, and users. Domain administrators can update and delete resources in a domain.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the left navigation pane, click **IDENTITY > Domains**. The **Domains** page opens.
3. Select **Create Domain**. The **Create Domain** window is displayed.
4. Specify the domain name and, optionally, the domain description.
5. Optional: Clear the **Enabled** check box to disable the domain. If the domain is disabled, the Domain Administrator cannot create, update, or delete resources that are related to the domain. New domains are enabled by default.
6. Click **Create Domain.**

   Note: The Keystone collector synchronizes information every night and hence transparently handles the inclusion of new projects or domains.

Assigning a zone to a domain
You must be logged in with the **admin** role to complete these steps.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the domains page by clicking **IDENTITY** > **Domains** in the navigation pane.
3. In the domains page, find the entry for the domain and click the arrow icon in the **Actions** column. Then click the **Edit** option to open the **Edit Domain** window.
4. Click the **Availability Zones** tab. The **Available Zones** and the **Assigned Zones** are listed in the following format: *Zone_Name* - *Region_Name*
5. To assign a zone to a domain, from the list of **Available Zones**, click the plus button beside the zone name. The selected zone moves to the **Assigned Zones** list. To return an **Assigned Zone** to an **Available Zone**, select the minus button beside the zone name. Use the **Filter** field to search for specific zones.
6. When you have assigned all zones, click **Save**.

Results
A message indicates that the domain is modified successfully.

Setting the default domain quotas
The Cloud Administrator sets the default domain quota for certain resources to specify the maximum amount of the resource that is available to the domain. The Domain Administrator can then distribute that quantity among all the projects in the domain.

Before you begin
You must be assigned the **admin** role to complete these steps.

About this task

The Cloud Administrator can assign a default *project quota* to certain resources, as described in “Configuring project quotas” on page 185.

To generate the default *domain quota* values, IBM Cloud Orchestrator multiplies the corresponding default project quota value by the **`projects_per_domain`** variable.

Note: The **`projects_per_domain`** variable is a multiplier that IBM Cloud Orchestrator applies to the default project quotas, to calculate the default domain quotas. The **`projects_per_domain`** variable does *not* specify the maximum number of projects that can be created in a domain.

The default value of the **`projects_per_domain`** variable is 5. The Cloud Administrator can change the value of the **`projects_per_domain`** variable, which consequently changes the default domain quotas, as follows:

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Log in as a root user.
3. Add the following lines to the `/etc/openstack-dashboard/local_settings` file:

```
SCO_CONFIG = {

    'projects_per_domain': <number of projects>,

}
```

4. Restart the HTTPd service:

```
service httpd restart
```

Editing the domain quotas

The Cloud Administrator can change the quotas of a domain to set limits on the operational resources that a Domain Administrator can distribute among all the projects in the domain. Domain administrators can update and delete resources in a domain.

Before you begin

You must be assigned the **admin** role to complete these steps.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the navigation pane, click **IDENTITY > Domains**.
3. On the Domains page, find the entry for the domain that you want to modify. In the **Actions** column for that entry, click **More** > **Edit**.
4. In the **Edit Domain** window, click the **Quota** tab.
5. Edit the quota values as wanted.
6. Click **Save**.

Results

A message is displayed, indicating that the quotas were saved to the domain successfully.

Modifying the list of domain administrators

You can add or remove users from the list of Domain Administrators to control a domain.

About this task

To modify the list of domain administrators who are assigned to a domain, complete the following steps:

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the navigation pane, click **IDENTITY** > **Domains**.

3. In the Domains page, select the entry for the domain. In the **Actions** column, click **More** > **Edit**. The **Edit Domain** window opens.
4. Click the **Domain Adminstrators** tab.

   Note: The **Domain Adminstrators** tab shows the following lists of users:

   - **All Available**: Users that are members of the domain but are not **Domain Users**
   - **Domain Administrators**: Users who are Domain Administrators for the selected domain.
5. To add a Domain Administrator, click **+**. The user is promoted from Domain User to Domain Administrator for the default project only. You must manually add the Domain Administrator user to all other projects in the domain, as described in "Modifying user assignments for a project" on page 186.
6. To remove a Domain Administrator, click **-**. The user is demoted from Domain Administrator to Domain User for all projects in the domain, but is not removed from any projects.
7. Click **Save**.

Results
The changes that you made to the list of domain administrators has been saved.

Setting a domain context
The domain context is for Cloud administrators to refine the context that they are accessing. Cloud administrators can limit the scope to one domain, rather than having visibility across all domains. This allows the Cloud administrator to identify the projects, users, groups, and roles that are associated with a domain.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the left navigation pane, click **IDENTITY** > **Domains**.
3. In the domains page, find the entry for the domain and click **Set Domain Context**.

Results
The domain page title changes to `<domainName>:Domains`. Selecting the **Projects**, **Users**, **Groups**, or **Roles** web pages only displays details for the selected domain context.

Clearing the domain context
Cloud administrators can clear the scope of all domains, enabling visibility across all domains.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the left navigation pane, click **IDENTITY > Domains**.
3. In the domains page, select **Clear Domain Context** from the top right-hand corner.

Results
All domains are visible.

Managing security groups
As Cloud Administrator, you can create, modify, or delete security groups in a domain.

Before you begin
You must be assigned the **admin** role to complete these steps.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.

2. In the navigation pane, click **PROJECT** > **Access & Security**. In the **Access & Security** panel, you can create, modify or delete a security group.
3. To modify a security group, click **Manage Rules** for the group that you want to modify and add or delete rules for the security group.

   Note: Because IBM Cloud Orchestrator uses SSH protocol and RXA protocol to communicate with the deployed virtual machines, you must ensure that the communication is enabled through ports 22 and 3389.

Managing projects

You can manage the level of access for each project to IBM Cloud Orchestrator with the user interface.

Before you begin

You must be assigned the **admin** role to perform these steps.

Managing projects from IBM Cloud Orchestrator Self service user interface

From IBM Cloud Orchestrator Self service user interface, you can now manage projects.

Creating a project

You can create projects from the IBM Cloud Orchestrator Self service user interface.

Before you begin

Set the domain context by using Setting the domain context.

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** >**Projects**.
3. Click **Create Project**. The **Create Project** window is displayed.
4. Specify the name for the project.
5. Optional: Enter a description for the project
6. Optional: By clearing the **Enabled** check box, you disable and cannot authorize the project. Selecting the **Enabled** check box keeps the project enabled so that you can authorize the project.
7. Click **Create Project**.

   Note: The Keystone collector synchronizes information every night and hence transparently handles the inclusion of new projects or domains.

Results

A message indicates that the project is created successfully.

Enabling a project

Enabling a project allows you to set that project as your default project. The action only appears if the project is disabled.

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** >**Projects** and select the project.
3. Click **Actions** > **Enable Project**.
4. In the message box, click **Confirm**.

What to do next

A message is displayed indicating that the project is enabled.

Editing a project

You can modify the name and description of a project.

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** > **Projects** and select the project.
3. In the **Actions** menu, click **Edit Project**.
4. In the **Project Info** tab, edit the name and description of the project.
5. Click **Save**.

Results

A message is displayed indicating that the project information has been modified.

Disabling a project

Disabling a project in a domain means that the users who previously had that project set as their default cannot log in to it anymore. Other users also cannot switch to this project anymore.

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** >**Projects** and select the project.
3. In the **Actions** menu, click **Disable Project**.
4. In the message box, click **Confirm**.

Results

A message is displayed indicating that the project is disabled.

Deleting a project

You can delete a project from the IBM Cloud Orchestrator Self service user interface.

About this task

If a project is deleted, all of its assigned resources (virtual machines, stacks, networks, images, and so on) remain in the cloud. Only the Cloud Administrator can manage these orphan resources. The Domain Administrator cannot recover from this situation.

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** >**Projects** and select the project.
3. Click **Actions** > **Delete Project**.
4. In the message box, click **Confirm**.

Results

A message is displayed, indicating that the project has been deleted.

Modifying a project zone

You can modify zones of a project from the IBM Cloud Orchestrator Self service user interface.

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** > **Projects** and select a project.
3. In the **Actions** menu, click **Modify Availability Zones**.

4. Select a zone from the **Availability Zones of Domain** and click **>>**. The selected zone moves to the **Assigned Zones of Project** list. To return an assigned zone to **Availability Zones of Domain**, select a zone in **Assigned Zones of Project** list and click **<<**.
5. Click **Ok** to save.

Results

A message indicates that the project is modified successfully.

Modify the quota of a project

As a cloud administrator you can modify the quota of a project.

Procedure

1. Log in to the Self-service user interface as a cloud administrator.
2. In the navigation menu, click **CONFIGURATION** > **Projects**.
3. Select a project.
4. From the **Actions** menu, click **Modify Quota**.
5. In the **Modify Quota** page, select a region from the drop down menu.
6. Click **Next**.
7. In the quota dialog box, enter values for the number of cores, the number of instances, and amount of memory.

   Note: The sum of all project quota of a domain can not exceed the overall quota of the domain. The validate button checks if that condition is met. If the condition is not met, the quota can not be changed. To view the overall domain quota and the quota remaining in the domain, click **Show Domain Quota**.
8. Click **OK**.

Results

The changes you made to the quota of the project has been saved.

Reassigning VMware instances to a project

Reassigning VMware virtual machine instances to a project enables virtual machines that have been loaded to a default project to be assigned to the project of a user who owns them.

About this task

You can perform this task only from the OpenStack Dashboard. For more information and steps to reassign, see "Reassigning VMware instances to a project" on page 185.

Modifying user assignments for a project

You can assign users to extra projects or update and remove assignments. You can also specify the roles that the user have for the project.

About this task

To modify user assignments for a project, complete the following steps:

Procedure

1. Log in to IBM Cloud Orchestrator Self service user interface as a Cloud Administrator.
2. Go to **Configuration** > **Projects** and select a project.
3. In the **Actions** menu, click **Modify User**.
4. Select a user from the **Users in Domain** and click **>>**. The selected zone moves to the **User in Project** list. To return an assigned user to domain list, select a zone in **User in Project** list and click **<<**.
5. Click **Ok** to save.

6. To change roles that are assigned to a project member, in the **User in Project** list, edit the role of a user.
7. Click **Ok**.

Results

The changes that you made to user assignments for a project has been saved.

Managing projects from OpenStack Dashboard

From the OpenStack Dashboard, you can manage projects in IBM Cloud Orchestrator.

Creating a project

You can assign individual zones to each domain in IBM Cloud Orchestrator with the OpenStack Dashboard.

Before you begin

Set the domain context through Setting the domain context.

Procedure

1. Log in to the OpenStack Dashboard as a Cloud Administrator.
2. Open the projects page by clicking **IDENTITY** > **Projects** in the navigation pane.
3. Click **Create Project**. The **Create Project** window is displayed.
4. Specify the name for the project.
5. Optional: Enter a description for the project
6. Optional: By clearing the **Enabled** check box, you disable and cannot authorize the domain. Selecting the **Enabled** check box keeps the domain enabled so that you can authorize the domain.
7. Click **Create Project**.

   Note: The Keystone collector synchronizes information every night and hence transparently handles the inclusion of new projects or domains.

Results

A message indicates that the project is created successfully.

Enabling a project

Enabling a project allows you to set that project as your default project. The action only appears if the project is disabled.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the projects page by clicking **IDENTITY > Projects** in the navigation pane.
3. In the projects page, find the entry for the project and click **More** > **Edit Project** in the **Actions** column.
4. In the **Edit Project** window, click the **Enabled** check box so that the box contains a tick symbol.

What to do next

A message is displayed indicating that the project is enabled.

Editing a project

You can modify the name and description of a project.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the projects page by clicking **IDENTITY > Projects** in the navigation pane.

3. In the projects page, find the entry for the project and click **More** > **Edit Project** in the **Actions** column.
4. In the **Project Info** tab, edit the name and description of the project.

Results

A message is displayed indicating that the project information has been modified.

Disabling a project

Disabling a project in a domain means that the users who previously had that project set as their default cannot log in to it anymore. Other users also cannot switch to this project anymore.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the projects page by clicking **IDENTITY > Projects** in the navigation pane.
3. In the projects page, find the entry for the project and click **More** > **Edit Project** in the **Actions** column.
4. In the **Edit Project** window, clear the **Enabled** check box so that the box is empty.

Results

A message is displayed indicating that the project is disabled.

Deleting a project

Delete a project in the OpenStack Dashboard as the Cloud Administrator.

About this task

If a project is deleted, all of its assigned resources (virtual machines, stacks, networks, images, and so on) remain in the cloud. Only the Cloud Administrator can manage these orphan resources. The Domain Administrator cannot recover from this situation.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the projects page by clicking **IDENTITY > Projects** in the navigation pane.
3. Find the entry for the project that you want to delete. In the Actions column for that entry, click **More** > **Delete Project**.

   Note: Deleting the default project of a domain results in the domain quotas becoming empty because the domain quotas are a multiplier of the default project quotas. Refer to Setting the default domain quotas for more details on the domain quotas.

   Note: If an `admin` user deletes a project, he must ensure to assign all users of this project to another existing project.

Results

A message is displayed, indicating that the project has been deleted.

Assigning a zone to a project

Assigning a zone to a project enables users within a zone to access a specific project.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the domains page by clicking **IDENTITY** > **Domains** in the navigation pane.
3. In the domains page, find the entry for the domain and select **Set Domain Context** in the **Actions** column. The **Identity Panel** group is now in the context of the selected domain and the **Domains** page is also changed. You are now working within the context of the domain that you created.
4. Select **IDENTITY** > **Projects**.

5. In the **Actions** column in the table for the project, click the arrow icon then click the **Edit Project** option.
6. Click the **Availability Zones** tab. The available zones and the assigned zones are listed in the following format: *Zone_Name* - *Region_Name*.
7. To assign a zone to a domain, from the list of **Available Zones**, click the plus button beside the zone name. The selected zone moves to the **Assigned Zones** list. To return an **Assigned Zone** to an **Available Zone**, select the minus button beside the zone name. Use the **Filter** field to search for specific zones.
8. When you have assigned all zones, click **Save**.

Results

A message indicates that the project is modified successfully.

Configuring project quotas

The Cloud Administrator can configure the project quotas in OpenStack.

About this task

Use the command-line interface to set the default project quotas and to change the project quotas in OpenStack.

Quota controls are available to limit the following resources:

- Number of volumes that can be created
- Total size of all volumes within a project as measured in GB
- Number of instances that can be started
- Number of processor cores that can be allocated
- Publicly accessible IP addresses

For more information about the OpenStack commands to be used to manage quota values in the projects, see the OpenStack documentation.

Reassigning VMware instances to a project

Reassigning VMware virtual machine instances to a project enables virtual machines that have been loaded to a default project to be assigned to the project of a user who owns them.

Before you begin

You must have an admin role on the source project containing the instances to be reassigned.

Important: This feature is available only if you are using IBM Cloud Manager with OpenStack V4.3 fix pack 13.

Procedure

1. Log in to the OpenStack Dashboard as a Cloud Administrator.
2. In the navigation pane, click **PROJECT** > **Instances**.
3. Find the instances to be reassigned and select the check boxes beside their names.
4. Click **Reassign Instances**.
5. Selected instances contain a list of the instances that are selected from the Instances table. The following options are available:
   - To clear an instance, click the instance in the list box. At least one instance must be selected.
   - To select all instances press **Ctrl-Shift-End**.
6. In the **Reassign Instances** window, select the **Target Domain** where the instances are to be assigned.
7. Select **Target Project** where the instances are to be assigned.
8. Click **Reassign**.

Note:

- When a virtual machine instance is reassigned from one project to another, the resources that are associated with the virtual machine (such as networks, IPs, flavors) are owned by the source project. If there are access issues to these resources from the new project, you need to recreate the resources on the new project.
- After reassigning a VMware instance, IBM Cloud Orchestrator might loose information about the virtual machine IP address which is not displayed in the Self-service user interface. When this problem occurs, you cannot perform some actions (for example, managing volumes) on this virtual machine.

Results

A message is displayed indicating that the instances have been reassigned successfully from the source domain and project to the target domain and project.

Modifying user assignments for a project

You can assign users to extra projects or update and remove assignments. You can also specify the roles that the user have for the project.

About this task

To modify user assignments for a project, complete the following steps:

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the navigation pane, click **IDENTITY** > **Projects**.
3. Click **Modify Users** for the project that you want to modify.

   Note: The **Edit Project** window shows the following lists of users:

   - **All Users**: Users who are members of the domain but are not members of this project.
   - **Project Members**: Users who are members of this project and associated roles. This list also shows the roles that are assigned to each project member.
4. To assign a user to this project, click +. The user is moved from the **All Users** list to the **Project Members** list.
5. To remove a user from this project, click -. The user is moved from the **Project Members** list to the**All Users** list.
6. To change the roles that are assigned to a project member: in the Project Members list, expand the role list for the user, and select the roles.
7. Click **Save**.

Results

The changes that you made to user assignments for a project has been saved.

Managing groups

As an administrator you can manage groups from the **Configuration** section of the Self-service user interface. Alternatively an administrator can manage groups from the OpenStack Dashboard.

About this task

Project users who share a common set of roles can be added as members of a group. A group is a collection of users in a domain. The administrator can create groups and add users to them. A role can then be assigned to the group, rather than to the individual users. User roles are additive in project, individual user role on project plus the role of user through group.

You can create a group, add users to the group, add roles to the group, and associate a group to a Project.

Consider the following examples for the group-based roles:

- Suppose that *Group A* is granted the role *Admin* on *Project A*. If *User A* is a member of *Group A*, when *User A* gets a token scoped to *Project A*, the token also includes Role *Admin*.
- Suppose that *Group B* is granted the role *Admin* on *Project B*. If *User B* is a member of *Group B* and *User B* is also a *Domain_Admin* of *Project B*, when *User B* gets a token scoped to *Project B*, the token includes the role *Admin* and *Domain_Admin*.

Note: Before creating and managing groups, set the domain context via "Setting a domain context" on page 173.

Creating groups

From the Self-service user interface, you can create groups to organize users and roles.

Procedure

1. Log in to the Self-service user interface as a Cloud Administrator.
2. Click **CONFIGURATION** > **Domain** > **Groups**.
3. In the **Actions** column, click **Create Group**.
4. In the **Define Group** page, enter the name and the description of the group. The group is created in the selected domain.
5. Click **Ok**.

   After successful creation, the new group appears in the list of groups.

Modify groups

You can modify the details of a group, associate a group to a project, and assign roles to a group.

Modifying groups in a project

Modify one or more groups from a project.

Procedure

1. Log in to the Self-service user interface as a Cloud Administrator.
2. Open the groups page by clicking **CONFIGURATION** > **Domain** > **Groups**.
3. In the **Groups** page, select the group to modify.
4. In the **Actions** column for that entry, click **Edit Group**.
5. Modify the group details and click **Ok**.

Associating a group to a project

All groups that are available in the domain of a project can be associated to a project. You can also disassociate a group from a project and release it back to the domain pool.

Procedure

1. Log in to the Self-service user interface as a Cloud Administrator.
2. Click **CONFIGURATION** > **Domain** > **Projects** in the navigation pane.
3. From the **Actions** column of the selected project, click **Modify Groups**.
4. Select the group to be associated and click **>>**. Assign the roles and click **Ok**.
5. Click **>>** to remove a group from the project.

   Note: If role is not selected, the group is not associated with the project.

   The project details show the groups added to the project with the assigned roles. The group details show the projects that it belongs to and the roles of the group on the project. The user details show the groups with project and the roles of the group on the project.

Deleting a group

Delete one or more groups in a domain.

Procedure

1. Log in to the Self-service user interface as the Cloud Administrator.
2. Click **CONFIGURATION** > **Domain** > **Groups** in the navigation pane.
3. Select one or more groups to delete.
4. In case you want to delete a group, in the **Actions** column of that selected group, click **Delete Group**. In case you want to delete multiple groups, select the groups and click **Delete Group**.
5. Click Confirm in the confirmation message box.

   A message appears to confirm the successful deletion of the groups.

Adding users to a group

You can add members to a group. All users of the associated project can be members of a group. You can also remove an user from a group.

Procedure

1. Log in to the Self-service user interface as the Cloud Administrator.
2. Click **CONFIGURATION** > **Domain** > **Groups** in the navigation pane.
3. In the **Actions** column of the selected group, click **Modify Users**.

   A **Users in Domain** page appears with a list of users of the selected group.
4. Select one or more users and click **>>**.

   You can also use the filter to search and select users.

   The selected users are added as members of the group.
5. Click **<<** to remove users from the group, in the **User in Group** page.

Managing users

You can manage the level of access for each individual user to IBM Cloud Orchestrator with the user interface.

About this task

Note:

- When the current user password is modified, after relogin, the user is redirected to the **Edit User** dialog, where the user must click **Cancel**.
- You cannot remove the email address by editing a user in the OpenStack Dashboard. To remove the email address, use the following command:

  ```
  keystone user-update --email "" <USER_NAME or USER_ID>
  ```

Creating a user

You can manage the level of access for each user in IBM Cloud Orchestrator. Users can be assigned to different roles on different projects.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the navigation pane, click **IDENTITY > Users**.
3. Click **Create User**. The **Create User** window is displayed.
4. Specify the required parameters, and then click **Create User**.

Results

A message indicates that the user is created successfully.

Deleting a user

You can delete one or multiple users in a domain.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. Open the users page by clicking **IDENTITY > Users** in the navigation pane.
3. Find the entry for the user that you want to delete. In the **Actions** column for that entry, click **More** > **Delete User**.

Results

A message is displayed, indicating that the user has been deleted.

Managing networks

As a cloud administrator you can manage networks in IBM Cloud Orchestrator with the user interface.

Creating a network

As a cloud administrator, you can create a new network in your environment.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. You can create a network in either of the following ways:
   - To create both a network and a subnet, click **PROJECT** > **Network** > **Networks**. Then, click **Create Network**. The **Create Network** window is displayed.

     You cannot specify the provider network type by using this method.

     After you entered the required information, click **Create**. A message appears indicating that the network is created successfully.

     For more information, see Create and manage networks and refer to the *"Create a network"* section.
   - To create a network and specify the provider network type, click **ADMIN** > **System Panel** > **Networks**. Then, click **Create Network**. The **Create Network** window is displayed.

     After you entered the required information, click **Create Network**. A message appears indicating that the network is created successfully.

     Use this method also to create networks that are shared among different projects.

     You cannot create a subnet by using this method. You can create the subnet after the network is created by following the procedure that is described in "Adding a subnet to an existing network" on page 190.

     For more information about managing networks in OpenStack, see the OpenStack Cloud Administrator Guide.

   Note: For a VMware region, the name of the new network must match the name of the network as defined in the vCenter Server that is being managed.

Deleting a network

As a cloud administrator, you can delete one or multiple networks in your environment.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the left panel, click **ADMIN** > **System Panel** > **Networks**. The **Networks** window is displayed.
3. Select the networks that you want to delete.

4. Click **Delete Networks**. A confirmation window is displayed.
5. Click **Delete Networks**. A message appears in the top right of the screen confirming that the networks have been deleted.

Modifying a network

As a cloud administrator, you can edit a network to modify the name and some options as, for example, if the network is shared among different projects.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the left panel, click **ADMIN** > **System Panel** > **Networks**. The **Networks** window is displayed.
3. Click **Edit Network** on the right of the network that you want to modify. The **Edit Network** window is displayed.
4. Make the required changes and click **Save Changes**. A message appears in the top right of the screen confirming that the network has been updated.

Adding a subnet to an existing network

As a cloud administrator, you can add a subnet to an existing network.

Procedure

1. Log in to the OpenStack Dashboard as the Cloud Administrator.
2. In the left panel, click **PROJECT** > **Network** > **Networks** or click **ADMIN** > **System Panel** > **Networks**. The **Networks** window is displayed.
3. Click the name of the network to which you want to add a subnet. The **Network Detail** window is displayed.
4. Click **Create Subnet**. The **Create Subnet** window is displayed.
5. Enter the required information in the **Subnet** and in the **Subnet Details** tabs.
6. Click **Create**. A message appears in the top right of the screen confirming that the subnet has been created.

Administering as domain administrator

Administer as a domain administrator.

A typical scenario for a Domain Administrator in a multitenancy environment is the on-boarding of users into the domain. The administrator assigns the users to projects and ensures that they can deploy virtual machines and use cloud resources.

Managing domains

Log in to the Self-service user interface as a Domain Administrator.

Click **CONFIGURATION** > **Domain** and select **Domains** from the menu that is displayed to see a list of domains that you can administer. To see more details about a particular domain, click the domain name. You can search for an instance by specifying the instance name or description in the search field. The instance table can be sorted by using any column that has the sort icon. Substrings are supported too but also wildcard.

Domain Administrators can manage projects, groups, users, actions, offerings, and categories. Domain Administrators can distribute the domain quotas among the projects in their domain.

Note: Domain Administrators cannot create domains, and they cannot change the domain quotas.

The Domain Administrator must perform the following steps to set up projects in the domain and to grant access to cloud resources to the domain users:

1. Create a project resource.
2. Ensure that the project has access to at least a deployment availability zone. This allows users in that project to access virtual images and deploy virtual servers when working within this project in the

logged in domain. The availability zones that can be assigned to the project are the ones that have been previously assigned to the domain to which the project belongs to.

3. Set the quota in the project.
4. Create and add users to the project.

These steps are detailed in "Managing projects" on page 191 and "Managing users" on page 194.

Managing projects

As a domain administrator you can manage the level of access for each project to IBM Cloud Orchestrator with the user interface.

You can search for a particular instance by specifying the instance name or description in the search field. The instance table can be sorted using any column that has the sort icon. Substrings are supported too but also wildcard.

Creating a project

As a domain administrator you can create a new project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Domains** in the menu below the navigation menu
4. Select the check box next to the domain you want displayed in the list.
5. Click **Create Project** in the **Actions** menu. The **Create Project** window is displayed.
6. Specify the name for the project.
7. Enter a description for the project
8. By clearing the enabled check box, you disable and cannot authorize the domain. Selecting the enabled check box keeps the domain enabled so that you can authorize the project.
9. Click **OK**.

   Note: The Keystone collector synchronizes information every night and hence transparently handles the inclusion of new projects or domains.

Results

A new project is created.

Enabling a project

As a domain administrator you can enable a project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box next to the project you want displayed in the list.
5. Click **Enable Project** in the **Actions** menu. This option only appears if the project is disabled.
6. Click **Confirm**

Results

A window appears at the top right of the screen confirming that the project has been enabled.

Edit a project

As a domain administrator you can edit a project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box next to the project you want displayed in the list.
5. Click **Edit Project** in the **Actions** menu.
6. Specify the name of the project.
7. Enter a description and domain for the project. By clearing the enabled check box, you disable and cannot authorize the project. Selecting the enabled check box keeps the project enabled so that you can authorize the project.
8. Click **OK**.

Results

A window appears in the top right of the screen confirming that the project has been edited.

Disabling a project

As a domain administrator you can disable a project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box next to the project you want displayed in the list.
5. Click **Disable Project** in the **Actions** menu. A window is displayed asking if you want to confirm launching the action: **Disable Project**.
6. Click **Confirm**.

Results

A window appears in the top right confirming that the project has been disabled.

Deleting a project

As a Domain Administrator you can delete a project in a domain.

About this task

If a project is deleted, all of its assigned resources (virtual machines, stacks, networks, images, and so on) remain in the cloud. Only the Cloud Administrator can manage these orphan resources. The Domain Administrator cannot recover from this situation.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box next to the project you want deleted.
5. Click **Delete Project** in the **Actions** menu.
6. A window appears asking if you want to delete the project. Click **Confirm**.

Note: Deleting the default project of a domain results in the domain quotas becoming empty because the domain quotas are a multiplier of the default project quotas. Refer to Setting the default domain quotas for more details on the domain quotas.

Results

A window appears at the top right of the screen confirming that the project has been deleted.

Modifying the availability zones of a project

As a domain administrator you can grant and revoke access of availability zones to a single project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box next to the project you want displayed in the list.
5. Click **Modify Availability Zones** in the **Actions** menu. The **Availability Zones of Domain** and the **Availability Zones of Project** are listed in the following format: **`Availability_Zone - Region.`**
6. Complete one or more of the following options to modify the user availability zones of a project:
   - To assign a zone to a domain from the list of **Availability Zones of Domain**, select an availability zone by selecting the check box beside it, then click the **>>** button. The selected zone moves to the **Availability Zones of Project** list.
   - To return an **Availability Zone of Project** to an **Availability Zone of Domain**, select an availability zone by selecting the check box beside it, then click the **>>** button beside the zone name.
7. Click **OK**.

Results

The changes that you made to the availability zones of the project were saved.

Modify the quota of a project

As a domain administrator you can modify the quota of a single project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domains**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box next to the project you want displayed in the list.
5. Click **Modify Quota** in the **Actions** menu.
6. Select the region from the drop down menu.
7. Click **Next**.
8. In the quota dialog box, enter values for the number of cores, the number of instances, and amount of memory.

   Note: The sum of all project quota of a domain can not exceed the overall quota of the domain. The validate button checks if that condition is met. If the condition is not met, the quota can not be changed. To view the overall domain quota and the quota remaining in the domain, click **Show Domain Quota**.
9. Click **OK**.

Results

The changes you made to the quota of the project has been saved.

Modifying users in a project

As a domain administrator you can add and remove users of a single project in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Projects** in the menu below the navigation menu.
4. Select the check box for the project that you want to edit.
5. In the **Actions** menu, click **Modify Users**.

   Note: The **Modify Users** page shows the following lists of users:

   - **Users in Domain**: Users in the current domain who are not assigned to the selected project.
   - **Users in Project**: Users assigned to the current project, with roles assigned. The user roles are also shown.
6. To assign a user to a project, select the check box beside the user, then click **>>**. The selected user moves to the **User in Project** list.
7. To remove a user from a project, select the check box beside the user, then click **<<**. The selected user moves to the **Users in Domain** list.
8. To edit the role assignment of a user in the **User in Project** list, click the **Role** column for the user. Select one or multiple roles from the roles list.
9. Click **OK**.

Results

The changes you made to user roles and user assignments for a project have been saved.

Managing users

As a domain administrator you can manage the level of access for each individual user to IBM Cloud Orchestrator with the user interface.

You can search for a particular instance by specifying the instance name or description in the search field. The instance table can be sorted using any column that has the sort icon. Substrings are supported too but also wildcard.

Creating a user

As a domain administrator you can create a new user in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Domains** in the menu below the navigation menu.
4. Select the check box next to the domain you want displayed in the list.
5. Click **Create User** in the **Actions** menu. The **Create User** window is displayed.
6. Specify the name for the user, the default project to assign the user to and the users role in that project.
7. Enter an email, password and domain for the user. By clearing the enabled check box, you disable and cannot authorize the user. Selecting the enabled check box keeps the user enabled so that you can authorize the project.
8. Click **OK**.

Results

A new user is created and appears in the **User** view. This action applies only to a single domain.

Deleting a User

As a domain administrator you can delete one or multiple users in a domain.

Procedure

1. Log in to the Self-service user interface as a Domain Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Domain**.
3. Click **Users** in the menu below the navigation menu.
4. Select the check box next to the user you want displayed in the list.
5. Click **Delete User** in the Actions menu.
6. Click **Confirm** in the window that opens

Results

A window appears in the top right of the screen confirming that the user has been deleted.

Managing networks

As Domain Administrator, you can create, modify, and delete the networks related to a specific project in your domain by using the OpenStack Dashboard.

For information about managing networks, see "Managing networks" on page 189.

Auditing login

When a user tries to log in to the Self-service user interface, the login action is logged for auditing purpose.

If the login fails, also the failure reason is logged.

The login actions are logged in the `/opt/ibm/ico/wlp/usr/servers/scui/logs/scoui.log` file on the IBM Cloud Orchestrator Server. Other log messages related to the Self-service user interface are available in `/opt/ibm/ico/wlp/usr/servers/scui/logs/scoui.log` file. You can find the messages related to the login actions by searching for the `login` string.

Note: The `/opt/ibm/ico/wlp/usr/servers/scui` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

Example of messages related to login actions:

```
[2015-04-13 15:36:57,892] [qtp-380990413-64] [INFO]
n3.app.handler.action.LoginHandler
Successful login: for user admin
...
[2015-04-13 15:37:38,764] [qtp-380990413-66] [INFO]
n3.app.handler.action.LoginHandler
Failed login: Invalid credentials for user admin
...
[2015-04-13 15:38:00,192] [qtp-380990413-55] [INFO]
n3.app.handler.action.LoginHandler
Failed login: No password provided for user admin
```

Changing the password

You can change your password by using the OpenStack Dashboard.

About this task

This procedure describes how to change the password of an `admin` user. For special users, additional steps might be necessary. For information about changing the password of other users, see "Changing the various passwords" on page 131.

Procedure

1. Log in to the OpenStack Dashboard as a Cloud Administrator.
2. In the upper-right corner of the window, you can see the name of the current project, the current region, and the current user. Click the name of the current user, and click **Settings** to display the **User Settings** page.
3. In the navigation pane on the left, click **SETTINGS** > **Change Password**.
4. In the **New password** and **Confirm new password** fields, enter the new password.
5. Click **Change**.
6. Update the `admin` user password in the `openrc` and `v3rc` files on all the OpenStack Controllers. Change the password by setting the `export OS_PASSWORD` variable to the new password.

Results

The password is changed successfully.

Synchronizing the directory for the scripts

In a high-availability installation, the directory where the scripts are stored must be synchronized between the two IBM Cloud Orchestrator servers. After a change to the directory, manually synchronize the directories on both servers.

For more information, see *File upload restrictions section of Content Development Guide*.

Chapter 7. Managing orchestration workflows

Create custom orchestration workflows in the Business Process Manager user interface and run them in your IBM Cloud Orchestrator environment.

Orchestration workflows

An orchestration workflow, which is based on Business Process Manager Business Process Definition, defines a logical flow of activities or tasks from a Start event to an End event to accomplish a specific service.

You can use the following types of orchestration workflows:

Offerings
Are used to define the offerings that cloud users can select in the Self-Service Catalog. They include user interface and the service request flow. For more information, see "Managing offerings" on page 228.

Actions
Are used to define IBM Cloud Orchestrator actions. They include user interface and the action flow. For more information, see "Managing actions" on page 230.

The service can be started either by events that are triggered by IBM Cloud Orchestrator management actions or by user actions in the IBM Cloud Orchestrator user interface. The activities that comprise the service can be either scripts (JavaScript), Java implementations, web services or REST calls, human tasks, and so forth. They can be either run in sequence or in parallel with interleaving decision points.

Each activity within an orchestration workflow has access to the cloud environment data in the form of the `OperationContext` object, which is passed as input parameter to each orchestration workflow. The operation context is an umbrella object that contains all data that is related to the execution of an operation. The operation context object must be defined as an input parameter variable for all business processes that are started as an extension for an IBM Cloud Orchestrator operation. Human services must define the operation context ID as an input parameter and as a first activity, must retrieve the operation context object with its ID. The operation context object contains metadata information, for example:

- User
- Project
- Event topic
- Status

For more information about the operation context object, see *OperationContext section of Content Development Guide*.

Workflows can give error events or post-status messages, which are then shown in the Self-service user interface.

An orchestration workflow can also have extra user interface panels to collect data that is needed as input. These panels are also implemented based on workflow technology, and they are called human services in Business Process Manager. For information about Business Process Manager, see "Working with Business Process Manager" on page 198.

Self-service offerings

Self-service offerings are typical administrative actions that are used to automate the configuration process.

Offerings, like actions, are custom extensions to IBM Cloud Orchestrator. You can develop these extensions by using Business Process Manager Process Designer, and then add them as offerings in the

CONFIGURATION tab in the IBM Cloud Orchestrator Self-service user interface. An offering can consist of:

- A Business Process Manager business process defining the activities to be performed by the extension.
- User interface panels that collect extra data, which is implemented by a Business Process Manager human service (optional).

Users access offerings in the **SELF-SERVICE CATALOG** tab, where they are grouped into categories.

For more information about offerings, see "Managing offerings" on page 228.

Samples and standard extensions for orchestration workflows

IBM Cloud Orchestrator provides an easy way to build the most common types of workflows. No programming is required. With IBM Process Designer, a graphical tool to build workflows, and the prebuilt samples and templates, you can create your own workflows with simple drag and drop editing, and attach common types of automation routines that are widely available.

Pre-built samples

IBM Cloud Orchestrator includes a set of toolkits that contain templates that you can reuse and adapt to your needs.

The `SCOrchestrator_Toolkit` provides the essential building blocks, which are needed to build Business Process Manager business processes and human tasks, which are then used as extensions for IBM Cloud Orchestrator.

For more information about the `SCOrchestrator_Toolkit`, see *Base orchestrator toolkit section of Content Development Guide*. For more information about all the provided toolkits, see *Developing IBM Cloud Orchestrator content section of Content Development Guide*.

You can also search for extra samples in the IBM Cloud Orchestrator Catalog. IBM Cloud Orchestrator Catalog is a platform and one-stop-shop for IBM customers, partners, and employees, where developers, partners, and IBM Service teams continuously share content among one another.

Advanced programming

To create more sophisticated automation, involving richer programming languages, see *Developing IBM Cloud Orchestrator content section of Content Development Guide*.

Working with Business Process Manager

You can use Business Process Manager processes to extend the capabilities of IBM Cloud Orchestrator.

About this task

Business Process Manager provides two user interfaces with which you can work to extend the capabilities of IBM Cloud Orchestrator - Process Center and Process Designer. You can switch between them to use different functions of the product.

Process Center is a web-based application that you start from the IBM Cloud Orchestrator user interface. In this application, you can review and manage process applications and toolkits that are known to the process server.

Process Designer is a stand-alone application that you install manually on a local computer. It includes a **Process Center** view that provides access to the repository but is enhanced with an **Open in Designer** option, with which you can design and configure your own workflows. You use the Process Designer graphical tool to create custom extensions to IBM Cloud Orchestrator. Prebuilt sample artifacts are provided in Business Process Manager to make process creation quick and easy.

For detailed information about IBM Business Process Manager, see the IBM Business Process Manager Knowledge Center.

For more information about developing process applications and toolkits, see *Developing IBM Cloud Orchestrator content section*.

Setting up the IBM Process Designer

Install, configure, and log in to IBM Process Designer.

About this task

IBM Process Designer is a stand-alone, local application that you install on a Windows operating system computer.

Procedure

1. Log in to the Business Process Manager user interface as an administrator user:

   ```
   https://ico_server_fqdn:443/ProcessCenter/login.jsp
   ```

   where *ico_server_fqdn* is the fully qualified domain name of the IBM Cloud Orchestrator Server.
2. Install Process Designer on a Windows machine on which you design the workflows:
   a) On the right-side panel of Process Center, click **Download Process Designer**.
   b) Install the package as described in Installing IBM Process Designer in the Business Process Manager information center.
3. Click **Start** > **IBM Process Designer Edition** > **Process Designer** and log in as user `bpm_admin` with password `passw0rd`.

Results

The Process Designer application opens and a list of process applications is displayed in the **Process Apps** tab. When you click the process application name, you can view its details, such as snapshots, history, you can edit some details such as name, or who can access it, but you are not able to configure the process application in this view. To configure a process application, click **Open in Designer** next to the item name.

You can switch between **Designer**, **Inspector**, and **Optimizer** tabs.

- To plan and design processes, use the **Designer** view.
- To test and debug processes, use the **Inspector** view.
- To analyze and optimize processes, use the **Optimizer** view.

To return to the **Process Center** view, click **Process Center** in the upper right corner of the panel. In the **Process Center** view, click **Open in Designer** to get back to the **Designer** view.

Adding users to IBM Process Designer

If you want a new IBM Cloud Orchestrator user to be able to use Process Designer, you must grant it access to the Process Center repository.

Procedure

1. Open Business Process Manager Process Designer or Process Center.
2. Go to **Admin** > **Manage Users**.
3. Click **Add Users/Groups** on the right side of the panel.

   A dialog box opens.
4. In the **Search for Name** box, enter the user name.
5. Select your user in the **Results** box and click **Add selected**.

Creating a process application in Process Designer

Create a process application and search through artifacts.

About this task

Process applications are containers in the Process Center repository for process artifacts such as process models and supporting implementations that are created in Process Designer.

Toolkits are collections of process assets that can be shared and reused across multiple projects in Process Designer.

Toolkits enable Process Designer users to share library items across process applications. Process applications can share library items from one or more toolkits, and toolkits can share library items from other toolkits.

IBM Cloud Orchestrator provides a toolkit that is called `SCOrchestrator_Toolkit`. It provides basic capabilities with which you can design orchestration processes. Any process application that contains processes for use with IBM Cloud Orchestrator must depend on this toolkit. For more information about `SCOrchestrator_Toolkit`, see *Base orchestrator toolkit section of Content Development Guide*.

When you create a dependency on a toolkit, you can use the library items from that toolkit for the implementation of the process steps you are building in your current project. For example, after you create a dependency on a toolkit that includes several services, the Designer view automatically makes those services available when a developer is choosing the implementation for an activity.

Procedure

1. Open the Process Designer and log on with administrative credentials.

   The **Process Center** panel is displayed. In this panel, you can review and manage process applications and toolkits that are known to the process server.
2. Create a process application:
   a) Click the **Process Apps** tab and in the panel on the right side, click **Create New Process App**.

   b) Provide a name and a unique acronym for your new process application. Optionally, provide a description.

      Remember: After the process application is created, do not change its acronym because it is used to reference the processes in self-service offerings.

   c) Click **Create**.

   Tip: The steps from a to c can be performed in both Process Designer and Process Center, with the same result, but only in Process Designer view you can configure the process application when it is created.
3. Click **Open in Designer** for your newly created process application.

   The **Designer** view is opened.
4. In the **Designer** view, click one of the categories from the pane on the left.

   A list of artifacts that are relevant for this category is displayed. In this pane, you can also review the existing artifacts and add new artifacts to toolkits or process applications.

   Note: You can click **All** in the newly created process application to see that it initially contains only one artifact.
5. Create a dependency on `SCOrchestrator_Toolkit`:
   a) Make sure the process application for which you are creating a toolkit dependency is open in the **Designer** view.

   b) Click the plus sign next to **Toolkits** in the library.

   c) From the **Add Dependency** window, click to select `SCOrchestrator_Toolkit`.

6. Click **TOOLKITS** > **System Data** from the list to open the toolkit. It is one of the default toolkits. Click **All** and then choose **Tag**, **Type**, or **Name** from the list on the right to sort the artifacts by tag, type, or name.
7. Click the arrow in the upper right corner to toggle between expanding all groups or expanding only one group.

   Tip: You can type any set of characters on the keyboard to search for artifacts that contain these characters.

Reusing processes and human services in a process application

After you create a process application, you can create processes and human services within this process application. You can also use the items that exist in other process applications or toolkits.

About this task

In this task, an example of the process application that is created in "Creating a process application in Process Designer" on page 200 is used.

Note: The **GetOperationContext** activity that is delivered with the `SCOrchestrator_Toolkit` is required as the first activity of the human service to integrate with IBM Cloud Orchestrator correctly. For information about the `SCOrchestrator_Toolkit`, see *Base orchestrator toolkit section of Content Development Guide*.

Procedure

1. In the Process Designer view search for `SCOrchestrator_Toolkit`.
2. On the right side of the toolkit name, click **Open in Designer**.

   Details for the toolkit are displayed.
3. From the list of available items on the navigation pane on the left, in the **User Interface** section, right-click the **Template_HumanService**. That is the user interface that you want to copy into your process application. The contextual menu for the selected item opens.
4. In the contextual menu, click **Copy Item to** > **Other Process App**. Select your process application from the list.

   The process is copied in the background. No confirmation is provided.

   Repeat steps 3. and 4. for all the items that you want to copy.
5. To return to the list of process applications and toolkits, click **Process Center** in the upper right corner of the screen.

Results

When you open your process application, the **Template_HumanService** is now available on the list.

Editing process applications and toolkits

You must be the author of a process application or a toolkit, if you want to edit them.

Table 16. Editing process applications and toolkits

Action	Authorization required
Editing imported toolkits	`admin`
Editing current Business Process Manager system toolkits	`tw_admins`, `tw_authors`
Creating a new process application and referencing `SCOrchestrator_Toolkit` as a dependency	`admin`
Opening sample applications as `admin`	`tw_admins` group added to the Process Center (open **Manage Access to Process Library**)

.

Creating a process

Create a process by using IBM Process Designer and incorporate a new activity into it.

About this task

In this example, create a sample process called `Hello World`. You can modify this procedure to suit your needs.

Procedure

1. In the **Designer** view, click the newly created process application and then click the plus sign next to **Processes** to open the **Create new** menu.
2. From the menu, select **Business Process Definition**.
3. Provide a name for the new business process definition, for example `Hello World`. Click **Finish**.

   The new process definition opens in the main canvas. Initially, the diagram view contains two lanes:

   - The **System** lane that is the default lane for system activities.
   - The **Participant** lane that is the default lane for user services.

   The start event and the end event are added automatically.
4. To add a user activity to the process, select **Activity** from the palette to the right. Add the activity to the **Participant** lane.

   An activity represents one step in the process. Properties of an activity are shown in the bottom panel.
5. In the **Properties** panel at the bottom of the screen, you can set the name of the activity, for example `Say Hello`.
6. To make the activity part of a flow, connect it with the start and end event. Select **Sequence Flow** from the palette.
7. With the **Sequence Flow** tool, click the connection points of the elements. First, connect the start event with the activity and then connect the activity with the end event.
8. To create an implementation for this process, click the plus sign next to **User Interface** in the **Designer** view.
9. From the menu that opens, select **Human Service** and name it `Say Hello`.

   The main canvas opens. You can now use the **Coach** element to create a simple user interface dialog that brings up the `Hello World` string.

   Note: The checkered background indicates that an implementation is being designed. It also indicates that there are now some elements in the palette that were not available in the process definition step.
10. Drag the **Coach** element onto the canvas. Specify its name, for example `Say Hello`, and double-click the element.

    The **Coaches** tab opens.
11. Drag the **Output text** element onto the canvas. In the **Properties** panel at the bottom of the screen, change the label to `Hello World`.
12. Drag the **Button** element and change its label to **OK**.
13. Click **Diagram** to open the **Diagram** tab. Use the **Sequence Flow** tool to connect the start event with the `Say Hello` step. Then, connect the `Say Hello` step with the end event.

    Note: Notice the **OK** label of the connection between the `Say Hello` step and the end event. It indicates that the **OK** button is used to trigger the transition to the end event.
14. Click **Save** to save the new human service.
15. From the list at the top, select **Hello World** to switch back to the previously created process definition.
16. Click the **Say Hello** step. In the **Properties** tab at the bottom, click **Implementation** and click **Select**.
17. From the list, choose the **Say Hello** human service that you created. Save the process definition.

Note: Notice that the implementation has changed to **Say Hello**.

18. Click **Run Process** in the upper right corner. The view automatically switches to the **Inspector** view.

    Notice that there is one active **Hello World** process instance. The **Diagram** view at the bottom shows that the instance is at step 2 and that there is a task that is waiting to be performed.
19. Click **Run the selected task** in the upper right corner to run the task. The **Pick User From Role** window opens. Select administrative user to run the task.

    Note: The selection of users in this window is based on the **Participant Group** setting of the **Participant** lane.

    A browser window opens that shows the simple user interface that you have defined. Such user interface is called a coach.
20. In the user interface, click **OK** and close the browser.
21. In the **Inspector** view, click **Refresh**. Notice that the **Say Hello** task is closed and the **Hello World** process instance is completed.

User input required at service request time

You can configure processes that require manual input from the user.

When manual input is required for a request to complete, the user gets a task assignment in the **INBOX** tab in IBM Cloud Orchestrator Self-service user interface. You must claim the task by clicking **Claim** before you can work with the task. If the task was already claimed by another user, the **Taken** status is displayed. There are two types of tasks:

Approval requests

To approve or reject a selected request, click **Accept** or **Reject** accordingly. Optionally, you can provide a comment in the **Reason** field.

General tasks

These are all tasks that are not of the approval type. Such tasks include a Business Process Manager human service. When you select the task from the list, the Business Process Manager coach opens. Provide any required parameters and click **Submit**.

Making a new process available as a self-service offering

You can make a process that you created available as a self-service offering in IBM Cloud Orchestrator.

Procedure

1. In Business Process Manager, expose the process to make it available in the IBM Cloud Orchestrator Self-service user interface:

   a) Open IBM Process Designer.

   b) Select your process and switch to the **Overview** tab.

   c) In the **Exposing** section, click **Select** in the **Expose to start** row.

   d) Select the **All Users** participant group or any other group that you want to expose the process to, and save the setting.

   Tip: A similar procedure must be performed to make a user interface (human service) visible in IBM Cloud Orchestrator:

   a. In Process Designer, select the human service and open the **Overview** tab.

   b. In the **Exposing** section, click **Select** in the **Expose to** row.

   c. Select the **All Users** participant group or any other group that you want to expose the process to, and save the setting.

   d. In the **Expose as** row, click **Select**.

   e. Select **URL** and save the setting.

2. In IBM Cloud Orchestrator, create an offering that is based on the process, and a category for it. For information about creating self-service categories and offerings, see “Creating a category” on page 229 and “Creating an offering” on page 228.

Results

The user can now access the offering in the Self-Service Catalog, and can request the offering.

Upgrading a process on a development system or production system

IBM Cloud Orchestrator enables you to distinguish between development mode and production mode.

About this task

You can define different upgrade methods for a process, depending on whether IBM Cloud Orchestrator is configured as a development system or as a production system.

Development mode

When IBM Cloud Orchestrator is configured in development mode, the engine always calls the most recent version of process code.

In IBM Process Designer, the most recent version of process code is called **Current** or **TIP**. Development mode is especially convenient during the development of processes because it is not necessary to create snapshots for each code change to be tested. Instead, the already running process instances (also known as *inflight* process instances) use the new modified code for the remaining part of the flow.

Development mode is configured by default after installation. For details of how to configure development mode, see “Configuring development mode” on page 204.

Production mode

When IBM Cloud Orchestrator is configured as a production system, the engine uses a snapshot version of the toolkit or process application code instead of the latest version of code.

A Business Process Manager process must be called by a dedicated snapshot ID, so that any running process continues to use that code version, even if a new snapshot is imported into the production system while the process is running.

The snapshot version that is used in production mode is determined as follows:

Toolkits

A production system uses the most recent snapshot (of the toolkit) that is available at the start time of a process instance.

Process applications

In general, a production system uses the most recent snapshot (of the process application) that is available at the start time of a process instance. However, an administrator can specify a default version of a process application. If a default snapshot is configured, the production system uses the default snapshot instead of the most recent snapshot.

For details of how to configure production mode, see “Configuring production mode” on page 205.

Configuring development mode

Development mode is configured by default after installation. You do not need to explicitly configure a system in development mode unless you are changing a production system to a development system.

About this task

To change IBM Cloud Orchestrator from a production system to a development system, complete the following steps:

Procedure

1. Log in to the WebSphere Application Server Console as an administrator user.

```
https://ico_server_fqdn:9043/ibm/console/logon.jsp
```

where *ico_server_fqdn* is the fully qualified domain name of the IBM Cloud Orchestrator Server.

2. In the console navigation tree, click **Servers** > **Server Types** > **WebSphere application servers** > ***server_name*** > **Process definition** > **Java Virtual Machine** > **Custom properties**.
3. Select the **ORCHESTRATOR_DEVELOPMENT_MODE** variable.
4. To specify development mode, set the value of the **ORCHESTRATOR_DEVELOPMENT_MODE** variable to `true`.
5. Set the description of the **ORCHESTRATOR_DEVELOPMENT_MODE** variable to `Activate Development Mode`.
6. Restart the Business Process Manager server.

Results

IBM Cloud Orchestrator is now configured as a development system.

Configuring production mode

To configure production mode, IBM Cloud Orchestrator must be manually configured to use a snapshot instead of the latest version of code.

About this task

To configure IBM Cloud Orchestrator as a production system, create a property in Business Process Manager to define the operational mode, as follows:

Procedure

1. Log in to the WebSphere Application Server Console as an administrator user.

   ```
   https://ico_server_fqdn:9043/ibm/console/logon.jsp
   ```

   where *ico_server_fqdn* is the fully qualified domain name of the IBM Cloud Orchestrator Server.
2. In the console navigation tree, click **Servers** > **Server Types** > **WebSphere application servers** > ***server_name***.
3. In the **Java and Process Management** category in the **Server Infrastructure** section, click **Process definition** > **Java Virtual Machine** > **Custom properties**.
4. Click **New** to create a new variable called **ORCHESTRATOR_DEVELOPMENT_MODE**.
5. To specify production mode, set the value of the **ORCHESTRATOR_DEVELOPMENT_MODE** variable to `false`.
6. Set the description of the **ORCHESTRATOR_DEVELOPMENT_MODE** variable to `Activate Production Mode`.
7. Restart the Business Process Manager server.

Optional: To specify a default snapshot for a process application, complete the following steps:

8. Activate the process application snapshot, as follows:
   a) Open the IBM Process Designer.
   b) Click the **Process Apps** tab.
   c) Select the process application.
   d) On the **Snapshots** page, expand the target snapshot, and then select **Activate**.
9. Specify the default process application snapshot, as follows:
   a) In a web browser, open the Process Admin Console.
   b) Click the **Installed Apps** tab.
   c) Select the process application.
   d) In the right pane, click **Make Default Version**.

Results

IBM Cloud Orchestrator is now configured as a production system.

Guidelines for working with Business Process Manager

When you create toolkits or process applications, there are some best practices to be followed in naming conventions, structuring, modeling, and error handling.

Guidelines for naming and documenting your toolkit or process application

When you create toolkits, use the following naming conventions:

- Name the toolkit after the utility or services it provides.
- Add words like "Toolkit" or "Framework" so that you can differentiate it from other process applications.
- Avoid long names. You must use fewer than 64 characters.
- White spaces between words can be added if it improves readability.
- Avoid the version number in the name, unless you want to bring attention to the major changes in the solution.
- Add more information about the toolkit in **Description** field.
- Choose an acronym for your toolkit. Do not use the prefix "IC“ as it is used for content that is delivered by IBM.
- Name your snapshots according to this scheme: AABB_YYYYMMDD. Exported TWX archives of your toolkit get this snapshot name appended, so you can easily identify the exported snapshots later.

  AA

  The IBM Cloud Orchestrator release that is prerequisite for the toolkit or process application, for example, 25 for IBM Cloud Orchestrator V2.5.

  BB

  Counting up the version of the toolkit, for example. 00 for the first release, and 01 for the second release

  YYYYMMDD

  The date the snapshot was created
- When updating an existing process application or toolkit, do not change the chosen acronym because it is used to reference the processes in self-service offerings.

Guidelines for creating artifacts in a toolkit

The general best practices are as follows:

- In the documentation field for a Business Process Manager artifact, enter a description of the input and output parameters of that artifact.
- Use the `note` object of Business Process Manager to improve the readability of complex processes and human services.
- As mentioned in the naming conventions, provide an understandable and meaningful name for your artifacts.
- Keep the interface definition between a Business Process Manager Human Service and its associated Business Process Manager Business Process Definition as short as possible. The interface is defined by a Business Process Manager Business Object. This object is used to correlate a business process with its associated human services in the IBM Cloud Orchestrator Self-service user interface. Use the Business Process Manager human service only to collect the parameters that are needed by its associated business process. Implement the business logic in the business process. It also helps if you enable the business process to be called by using the REST API from an external application, such as a portal application.
- Avoid **Pre** and **Post** execution assignments. Instead, add explicit activities, if needed. The execution assignments are hidden in the Business Process Manager Process Designer, and the logic of the corresponding activity or service becomes difficult to understand. If needed, use the **Pre** and **Post**

executions to make simple assignments like initializing the associated Business Process Manager artifact. For example, consider having two consecutive coaches in a human service. In such cases, do not initialize the objects that are used by the second coach as being **Post** execution assignment of the first coach. If needed, do the initialization as a preexecution assignment of the second coach.

- Do not use passwords in environment variables or other artifacts that are visible to everyone.
- When you deliver a solution for IBM Cloud Orchestrator, make sure that there are no validation errors. These errors can be seen in the Process Designer.
- Avoid changing the interface of a building block that is delivered as a part of a toolkit. If you change the interface of building blocks in a toolkit, it becomes cumbersome for all its dependent toolkits or Process Applications. Even changing the name might lead to redoing the mapping for all activities or services that use the building block.

Guidelines to structure your solution

- In general, an extension content solution for IBM Cloud Orchestrator consists of a Business Process Manager Process Application and a Business Process Manager Toolkit.

  The basic rule is that a process application contains artifacts that are ready to be used by the user and not meant to be changed or adapted to be useful. All other artifacts are better placed in a toolkit.
- When structuring your solution, always consider the visibility of your artifacts. Artifacts of one process application are not visible by default to another process application.

  For example, a Business Process Manager, process A, can be called by another Business Process Manager, process B. The ‘Linked Process’ activity is used if both are in the same process application or if process A is in a dependant toolkit.

  Avoid cyclic dependencies, that is, when toolkit A depends on toolkit B, avoid having a dependency on toolkit A. If such a cyclic dependency occurs, restructure your toolkits to resolve it.
- Use Business Process Manager tags and smart folders to structure your solution to make it more understandable. If you have UI parts that can be used in UI panels, define them as Coach views. These views can be reused in different Coaches. If you must change something later, for example, wording, you change only the reusable Coach view.

Guidelines for handling errors

An IBM developerWorks article explains extensively about exception handling and logging from a business process management perspective. See Related Links. It identifies the types of exceptions that are encountered in a Business Process Manager scenario. Also, it shows you how to handle them using IBM Business Process Manager.

The following are best practices in error handling:

- Define error message as localization resources.
- Raise errors in your integration services or processes by using the Error End Event node.
- Catch raised errors that are raised from integration services by using Intermediate Error Events or Event Subprocesses.
- For Java classes that are used in Business Process Manager processes or human services, define logging framework. For example, `java.util.logging` to log messages to the WebSphere log.
- Use the logging capabilities of Business Process Manager to log messages to the WebSphere log. A good practice is to log in the entry and exit of an activity to support debugging better.

Related information

Chapter 8. Working with self-service

IBM Cloud Orchestrator provides an intuitive Self-service user interface, where you can use the Self-Service Catalog to request resources. For example, you can deploy virtual machines, add volumes, or manage key pairs. From this interface, you can also monitor your requests and manage your resources.

About this task

IBM Cloud Orchestrator provides a rich set of predefined offerings in the Self-Service Catalog. In addition, Service Designer can create offerings with IBM Process Designer, and populate the Self-Service Catalog with extra offerings.

An offering is a Business Process Manager process in the Self-Service Catalog. For information about toolkits that you can use to build offerings, see *Developing IBM Cloud Orchestrator content of Content Development Guide*.

Using self-service

Use the IBM Cloud Orchestrator Self-service user interface to request resources, monitor the status of your requests, and do additional tasks related to resources.

Viewing the dashboard

Use the **DASHBOARD** tab to monitor tasks, requests, quota usage, and virtual machine status for the current project.

Inbox

The **Inbox** area provides an overview of the inbox statistics.

The section header displays the number of each of the following types of task:

- **New today**
- **To-do** (tasks that have not yet been claimed)
- **Overdue**

The table displays the following information about the most recent tasks:

- **Latest Items**
- **Requested by**
- **Priority**
- If a task is overdue, the overdue icon is displayed.

Click a task type in the section header, or click an item in the table, to open the **INBOX** tab.

Requests

The **Requests** area provides an overview of requests statistics.

The section header displays the number of each of the following types of requests:

- **New today**
- **In progress**
- **Failed**

The table displays the following information about the most recent requests:

- **Latest Requests**

- **Submitted On**
- **Status**

Click a request type in the section header, or click an item in the table, to open the **REQUESTS** tab. If you click a request type, only requests of that type is displayed.

Quota Usage of Current Project

The **Quota Usage of Current Project** area provides the current aggregate of the following items for the current project:

- **vCPU Usage**
- **RAM (MB) Usage**
- **Volume (GB) Usage** (for attached volumes only)

Aggregates are based on all virtual machines in your project across all regions.

Usage is displayed as a percentage. The dial displays 50%, 75%, and 100% threshold indicators, which are color-coded as follows:

Color	% Usage
Green	0-50
Yellow	51-75
Red	76-100

Note: For VMware regions, if you are logged as `admin` user in the `admin` project, the displayed usage is always `0%` even if deployed virtual machines exist.

VM Status

The **VM Status** area provides information about the deployed virtual machines for the current project. The total number of deployed virtual machines in your project across all regions is displayed, with a breakdown based on status. The virtual machine status is color-coded as follows:

Color	Status
Green	Active
Blue	Paused
Red	Error
Yellow	Shutoff

Click a status type to open the **RESOURCES** tab. The tab contents are filtered to display only virtual machines with the selected status.

Submitting a self-service request

Use the **SELF-SERVICE CATALOG** tab to view the list of offerings and submit a self-service request.

Procedure

1. Log on to the IBM Cloud Orchestrator Self-service user interface and click the **SELF-SERVICE CATALOG** tab.
2. Open the category of your choice to view the offerings. You can also use the **Search** field to look up a specific offering by name. The search works fine also for substrings and wildcard are not supported.
3. Select an offering from the list.

   A window with request details opens.

4. Specify any required parameters for the request.
5. Click **OK** to submit the request.

Results

The request is submitted. A message is displayed at the top of the page, reporting the result. You can also check the request status in the **REQUESTS** tab.

Viewing the status of your requests and actions

Use the **REQUESTS** tab to review the status of your requests and actions.

About this task

You can view the progress of all the requests that you submitted from the Self-Service Catalog. You can also view the progress of all actions that you have submitted and that are related to resources and configuration. Administrators can also view all the requests and actions submitted by the users that they administer.

Procedure

1. Click the **REQUESTS** tab.

   All the requests and actions that you have access to are displayed on the left side of the page.

   You can search for a specific request or action and sort them in the view.
2. Click any request or action to view its details.

Deleting request history

You can remove requests from the history of IBM Cloud Orchestrator. Modify parameters in the delete request response file `./delete_request_history.rsp`.

To delete request history, you must log in either as an administrator or a domain admin.

The `delete_request_history.sh` script allows you to remove requests from the history of IBM Cloud Orchestrator, and it accepts the following parameters:

- **`FAILED_REQUESTS`** - If you set the parameter to "True", then the script deletes all failed requests from IBM Cloud Orchestrator.
- **`REQUEST_BEFORE_DATE`** - Enter a date value to delete all jobs launched before the specified date. For example, `2019-08-02`.
- **`REQUEST_DATE`** - Enter a date value to delete all jobs on the specified date. For example, `2019-08-02`.

Note: In case of external DB setup, the `./delete_request_history.sh` must be manually copied to the installed database node. Ensure that you do not copy `./delete_request_history.sh` and `./delete_request_history.rsp` files to the home directory of the root user. Example:

```
./delete_request_history.sh delete_request_history.rsp
```

Managing resources

Use the **RESOURCES** tab to manage your assigned resources.

The columns for each instance type table can vary. From the table view you can launch actions on a single instance or on multiple instances. To view detailed information about an instance click anywhere on the instance row. The details screen contains actions that pertain only to the selected instances.

Resource types

IBM Cloud Orchestrator supports several types of resources, including domains, virtual machines, and volumes. Resources types are also known as instance types (for example, in the Core Services REST API).

IBM Cloud Orchestrator provides the following resource types:

Action
An action is an instance that can be applied to other instances. An action always includes a Business Process Manager process that can be run on the associated instance.

Category
A category is a container for self-service offerings that are displayed in the Self-Service Catalog. You can organize your offerings inside the catalog.

Domain
A domain is the highest entity in the identity model. It is a container and namespace for projects and users of a customer.

Heat template
A Heat template represents an OpenStack Heat Orchestration Template (HOT).

Heat stack (also called Stack)
A Heat stack represents a combined set of resources, for example, virtual machines, created through the deployment of a Heat template in OpenStack.

Offering
An offering is an IBM Cloud Orchestrator process made available in the Self-Service Catalog. IBM Cloud Orchestrator provides a set of offerings out of the box. However, you can create your own offerings with IBM Process Designer.

Openstackvms
An instance of type `openstackvms` represents a single OpenStack virtual server.

Project
A project is a container that owns resources such as virtual machines, stacks, and images.

User
A user represents the account of a person. You can log in to IBM Cloud Orchestrator with a user account. A user must always be member of at least one project.

Volumes
The volumes instance type is the disk space resource that can be attached to a virtual server.

VMware Deployments
An instance of type VMware Deployments represents a VMware virtual machine. For more information about this resource type, see *Direct Driver VMware toolkit section of Content Development Guide*.

PowerVC Deployments
An instance of type PowerVC Deployments represents a PowerVC virtual machine. For more information about this resource type, see *Direct Driver PowerVC toolkit section of Content Development Guide*.

Working with resources

You can work with your assigned resources from the **RESOURCES** tab in the navigation menu.

You can search for a particular resource by selecting the resource type and specifying the resource name, description, or status in the search field. The resource table can be sorted using any column that has the sort icon.

The search on resources works also for substring of what you are searching for but not through wildcard. For example, if you are searching for a virtual machine named Linux06, this is correctly returned specifying Linux06 (entire string) or Linu (substring) or even inu (substring) but wildcard are not accepted so Linu* or Linu.* are not valid here.

Applying an action to a resource

Use the **RESOURCES** tab to select an action for an instance, for example, to start or stop a virtual machine.

Procedure

1. Log in to the Self-service user interface and click **RESOURCES**.

2. Select an instance by selecting the single row check box. Select multiple instances by selecting the check box in the header row.
3. Select the action that you want to run from the **Actions** box on the left. The following options are displayed based on the action type selected:

   Human service
   If the action you select requires a human service, a window with request details opens. Specify any required parameters for the request. Click **OK** to run the action or **Cancel** to return to the previous view.

   No human service
   If the action you select does not require a human service, a confirmation dialog box is displayed. Select **Continue** to run the action or **Cancel** to close the dialog and return to the main view.

   Note: When an action is run, a message is displayed at the top of the page, reporting the result. You can also check the request status in the **REQUESTS** tab.

Related concepts

"Managing actions" on page 230
A Service Designer can manage actions and their access control list in the Actions Registry.

Removing from the list of managed servers in PowerVC

PowerVC treats base instances that are used for the capture of images no differently from other servers on the system. This means that there is the possibility for the base servers to be deleted accidentally by administrators. To avoid this, it is recommended that after a server is captured as an image, it must be removed from PowerVC. Removing a server does not affect the server except that it is no longer managed by PowerVC and it cannot be deleted inadvertently.

About this task

Removing a server does not affect the server or its associated disk. If needed, the server can be managed through the servers panel.

Procedure

1. Click the **Hosts** icon in the side panel on the left of the screen.
2. Click a specific Host. A panel appears displaying a list of managed servers on that host.
3. Select the server that you want to remove and click **Remove** in the menu bar.
4. A window appears asking to confirm removing the virtual machine that you have selected from PowerVC management. Click **OK**.

Results

The server has been removed and can no longer be managed by PowerVC.

Managing virtual machines

The IBM Cloud Orchestrator Self-Service Catalog provides self-service offerings to deploy virtual machines using the OpenStack Nova component. It can also register and unregister a public and private key pair that can be used to access the virtual machine.

Deploying a virtual machine

The IBM Cloud Orchestrator Self-Service Catalog provides a default offering that you can use to deploy a single virtual server by using OpenStack Nova.

Before you begin

- Resources such as images, flavors, keys, and networks, must be defined in the OpenStack environment.
- Images must be stored within OpenStack Glance.
- Flavors and networks must be defined in OpenStack.
- Keys must be registered with a project.

- For tasks such as processing user data or metadata, the image must be prepared to include the `cloud-init` package.
- Assign the region and availability zone to the project.

About this task

To deploy a single virtual server, use the following procedure.

Alternatively, you can deploy a predefined Linux virtual machine or Windows virtual machine by following the procedure described in *Deploying a Linux virtual machine section of Content Development Guide* or *Deploying a Windows virtual machine section of Content Development Guide*.

Note: AIX® images do not have the ability to change the password of a user or inject an SSH key at deployment time. Though these options do not work for AIX, they are available on the page.

Note: In VMware, if you have virtual machine instances in disconnected or inaccessible status, and with an empty value for CPU or memory, the deployment of a new virtual machine fails with the message:

```
Services in selected region are not available
```

To deploy a new virtual machine, fix the status of the virtual machine instance or remove it from the inventory in VMware.

Note: You must match the number of NICs with the number of networks and IP addresses used for deployment. For example, an image with two defined NICs must be deployed by using two or more networks. If you try to deploy it using only one network, the deployment fails.

Procedure

1. Log in to the Self-service user interface as an End User.
2. In the menu bar, click **SELF-SERVICE CATALOG**.
3. Click **Deploy cloud services** > **Deploy single virtual server**. The **Deploy single virtual server** page opens.
4. Select the region where the virtual server must be deployed to.
5. Click **Next**.
6. Enter the virtual server details:
   a) Specify a **Server Name**.

      Note: Depending on the selected region and the associated hypervisor, restrictions might apply on the set of characters which can be used for the server name. For example, VMware does not allow to use blanks or underscores in the server name.

   b) Select an **image**, **availability zone**, and **flavor** from the drop-down menus. Select the **network** by selecting the check box beside the network name.

      For more information about availability zones, see:

      - Assigning a zone to a domain
      - Modify the availability zones of a project

   c) Optional: If a key was generated, you can select **Use Key to Access Virtual Machine** and, in the displayed menu, select one of the keys to access a virtual machine.
   d) Optional: Select **Set UserId and Password**.

      Enter a user ID and specify whether a password must be set on the virtual machine. To set the password on the virtual machine, the `cloud-init` package must be installed on the deployed image. Another check box is displayed to specify whether the password must be changed at the first login.

      The default user to be used can be configured in the `cloud-init` configuration file. Each Linux distribution has its own default user. For more information, see http://cloudinit.readthedocs.org/en/latest/topics/examples.html.

Note: For Microsoft Windows virtual machines: Do not edit the **UserId** field. You can change the password only for the user who is specified in the `cloudbase-init` configuration file.

Note: The default implementation of the **Deploy single virtual server** offering does not check any password rules. This can be cumbersome, for example if the provisioning of a Windows virtual machine fails after a while with an invalid password exception as Windows does not accept the password provided. To avoid this you can copy the Business Process Manager human service **Deploy Single Virtual Machine** and add your own password checking to the **Validate** building block called after the user has entered the password.

Note: When deploying a virtual machine to a public cloud as Amazon AWS EC2 or IBM SoftLayer®, you cannot set user ID and password to access the deployed virtual machine.

e) Optional: Click **Attach Volume** and in the displayed window select the volumes to be attached to the deployed virtual machine.

Note: The selected volumes are only attached to the deployed machine but they are not formatted or mounted. Use the disk management tools of the virtual machine operating system to format and mount the attached volumes.

What to do next

Proceed to Managing virtual machine instances.

Managing virtual machine instances

Virtual machine instances represent the servers (virtual machines) that are running in the OpenStack backend of IBM Cloud Orchestrator.

IBM Cloud Orchestrator provides a built-in instance type that is called OpenStack virtual machines that provide the functions to manage deployed virtual machines.

To work with virtual machines, complete the following steps:

1. Log in to the Self-service user interface as an End User.
2. Click **RESOURCES**.
3. Click **Virtual Machines** to display a table of virtual machines.
4. From the region list, select a region. The table shows only the virtual machines in the specified region.

From this view, you can perform the following actions:

Starting one or more virtual machines

1. In the table, select one or more virtual machines that have the `SHUTOFF` status.
2. In the **Actions** menu to the left of the table, click **Start**.

   Note: This action is available only if all of the selected virtual machines have the `SHUTOFF` status.

Stopping one or more virtual machines

1. In the table, select one or more virtual machines that have the `ACTIVE` status.
2. In the **Actions** menu to the left of the table, click **Stop**.

   Note: This action is available only if all of the selected virtual machines have the `ACTIVE` status.

Deleting one or more virtual machines

1. In the table, select one or more virtual machines that have the `ACTIVE` status.
2. In the **Actions** menu to the left of the table, click **Delete**.

   Note: This action is available only if all of the selected virtual machines have the `ACTIVE` status.

Resizing a virtual machine

1. In the table, select a single virtual machine.
2. In the **Actions** menu to the left of the table, click **Resize** to change the flavor of an existing virtual machine. You can see the current flavor of the virtual machine, and select the wanted flavor.

Tip: After a resize action to increase the disk size of the virtual machine successfully completes, the disk is increased from a hypervisor point of view. If you log on to the virtual machine and the file system does not reflect the new disk size, you must rescan or restart to reflect the changed disk size. This action depends on the operating system and disk type, as follows:

- Microsoft Windows: For information about how to resize the file system without a restart, see the Microsoft TechNet article Update disk information.
- Linux: For information about how to adapt a file system after you increase the virtual disk, see the VMware Knowledge Base article Increasing the size of a disk partition (1004071) .

Executing a command or uploading and executing a script

1. In the table, select a virtual machine that has the `ACTIVE` status.
2. In the **Actions** menu to the left of the table, click **Execute Script**.
3. In the displayed panel, specify the following parameters:

   Select Operating System Type
   Select the operating system of the deployed virtual machine.

   Select Network
   Select the network of the deployed virtual machine.

   User Name
   Specify the user for logging in to the deployed virtual machine.

   Password
   For Windows virtual machines or if the Linux virtual machine was not deployed with an SSH key, specify the password for logging in to the deployed virtual machine. This field is not displayed if the Linux virtual machine was deployed with an SSH key.

   Using SSH Key
   If the Linux virtual machine was deployed with an SSH key, specify the key used to connect to the deployed virtual machine. This field is not displayed for Windows virtual machines and if the Linux virtual machine was not deployed with an SSH key.

   Note: Microsoft Windows must be configured for RXA connections as described in Requirements for using Remote Execution and Access (RXA).

4. Select one of the following options:

   Execute Command
   Select this option to run a command or a script already existing on the deployed virtual machine. Specify the following parameters:

   Command Line
   Specify the command that is run on the deployed machine, for example: `sh HelloWorld.sh`. The command or script used in the command line must be available on the target virtual machine.

   Working Directory
   Specify the directory on the deployed virtual machine where the specified command is run.

   Execute Script
   Select this option to upload and run a script on the deployed virtual machine. Specify the following parameters:

   Select File For Upload
   Select the script to be uploaded to the deployed virtual machine.

   Command Line For Script
   Specify the command to run the uploaded script on the deployed virtual machine.

   Destination Folder
   Specify the directory on the deployed virtual machine where the script is uploaded.

   Working Directory
   Specify the directory on the deployed virtual machine where the uploaded script is run.

5. Click **OK**.

Note: The **Execute Script** action is implemented by using the `SCOrchestrator_Scripting_Utilities_Toolkit`. There are additional building blocks available to extend and customize the capabilities. For more information, see the *Scripting utilities toolkit* section in the *IBM Cloud Orchestrator Content Development Guide*.

Managing volume

1. In the table, select one virtual machine that has the `ACTIVE` status.
2. In the **Actions** menu to the left of the table, click **Manage volume**.
3. Select a volume and the related action for the volume (attach, detach, delete).

For more information, see "Working with volumes" on page 225.

Note: If Microsoft Windows is the guest operating system, there might be situation where the default RXA connection timeout is too small. The result is that the volume is not correctly formatted or mounted within the guest operating system. If this problem occurs, you might see connection errors to the provisioned virtual machine in the `SystemOut.log` file. The following property must be added to the custom properties of the Business Process Manager Java WebSphere Configuration:

```
com.ibm.tivoli.remoteaccess.connection_timeout_default_millis
```

If the provisioned virtual machine resides on a remote cloud managed by Public Cloud Gateway, also see "Public Cloud Gateway overview" on page 249 and "Network planning" on page 258.

Run the following configuration steps:

1. Log on to `http://<ico_server_fqdn>:9060/admin` using `bpm_admin`.
2. Navigate to **Servers > All servers > SingleClusterMember1 > Java and Process Management > Process definition > Java Virtual Machine > Custom properties**.
3. Click **New** to add the following property:

   Property name
   `com.ibm.tivoli.remoteaccess.connection_timeout_default_millis`

   Value
   At least 180000 which is 3 minutes. Time is in milliseconds.
4. Restart Business Process Manager to activate the change.

Related tasks

"Applying an action to a resource" on page 212
Use the **RESOURCES** tab to select an action for an instance, for example, to start or stop a virtual machine.

Detach drive

You can add and manage a vCenter connection from within the user interface of IBM Cloud Orchestrator. In addition, you can automatically detach the ISOs that got attached during the deployment of a single virtual server. It also includes virtual servers that were deployed by using the heat template. Detach operation supports all attached drives. For example, configdrive that is attached during deployments, such as single virtual server and heat template.

Prerequisites
As a prerequisite, add VMware server certificate to WebSphere Application Server of IBM Cloud Orchestrator.

Procedure

1. Log in to WebSphere Application Server Administrative Console at `https://<ICO_Server_IP>:9043/ibm/console/login.do?action=secure`.
2. In the left navigation, select **Security** > **SSL certificate and key management**. On the right hand side, the SSL certificate and key management page is displayed.

3. In the **Related Items** section, select **Key stores and certificates**.
4. Select **CellDefaultTrustStore**.
5. In the **CellDefaultTrustStore** page, click **Additional Properties** section > **Signer Certificates**.
6. In the **Signer certificates** page, click **Retrieve from port**.
7. In the **General Properties** section of the **Retrieve from port** page, enter the **Host**, **SSL configuration for outbound connection**, and **Alias**.
8. Click **Retrieve signer information** to retrieve certificate information from the provided host.
9. Click **Apply** to apply the certificate on WebSphere Application Server.
10. Click **Save** to save the changes to master configuration.
11. Verify whether the certificate is added in **CellDefaultTrustStore**.
12. In the left navigation, select **Security** > **SSL certificate and key management**. On the right hand side, the SSL certificate and key management page is displayed.
13. In the **Related Items** section, select **Key stores and certificates**.
14. In the **Preferences** section of **Key stores and certificates** page, click **NodeDefaultTrustStore**.
15. In the **Additional Properties** section of **NodeDefaultTrustStore**, click **Signer Certificates**.
16. Click **Retrieve from port** to retrieve certificate information from the provided host.
17. Enter **Host**, **Port**, **SSL configuration for outbound connection**, and **Alias**.
18. Click **Retrieve signer information** to retrieve certificate information from the provided host.
19. Click **Apply** to apply the certificate on WebSphere Application Server.
20. Click **Save** to save the changes to master configuration.
21. Verify whether the certificate is added in **NodeDefaultTrustStore**.

    Note: For HA topology, add the NodeDefaultTrustStore certificate on both primary and secondary IBM Cloud Orchestrator nodes.

Working with actions and offerings
When you install IBM Cloud Orchestrator, the following actions are imported automatically in your IBM Cloud Orchestrator environment.

- "Add vCenter Details" on page 218
- "Update vCenter Details" on page 219
- "Delete vCenter" on page 219
- "View vCenter Details" on page 219

Add vCenter Details
You can add a vCenter connection from IBM Cloud Orchestrator Self-Service user interface.

About this task

- At any given point in time, you can add only one connection of VMware virtual center per region.
- Add only the details of the vCenter that is configured with IBM Cloud Orchestrator and acts as a VMware controller.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service user interface.
2. Go to **CONFIGURATION** > **Actions Registry**.
3. Click **vCenter Details**.
4. From the **Actions** menu, click **Add vCenter Details**.
5. In the **Add VMware Virtual Center Details** page, select your region and click **OK**.
6. Enter the following details of VMware Virtual Center:
    - **VMware Virtual Center Connection Name -** The name of the VMware vCenter connection.

- **Description** - The description of the connection.
- **VMware Virtual Center IPAddress** - The IP address of the VMware vCenter.
- **VMware Virtual Center Username** - The user name to log into VMware vCenter.
- **VMware Virtual Center Password** - The password of the VMware vCenter user.
- **Re-Enter Password** - Re-enter the password of the VMware vCenter user.
- **DataCenter Path** - The location of the DataCenter.

7. Click **OK**.

   After the success of the operation, the newly added vCenter connection is listed in the **vCenter Details** section.

Update vCenter Details
You can update vCenter connection details from IBM Cloud Orchestrator Self-Service user interface.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service user interface.
2. Go to **CONFIGURATION** > **Actions Registry**.
3. Click **vCenter Details**.
4. From the **Actions** menu, click **Update vCenter Details**.
5. In the **Edit VMware Virtual Center Details** page, you can update the vCenter values.
6. Click **OK**.

View vCenter Details
You can view vCenter connection details from IBM Cloud Orchestrator Self-Service user interface.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service user interface.
2. Go to **CONFIGURATION** > **vCenter Details**.
3. Search for your vCenter and click to view its details.

   From the **Actions** menu of the **View VMware Virtual Center Details** page, you can update or delete a vCenter connection.

Delete vCenter
Whenever you do not need a vCenter connection, you can delete it from IBM Cloud Orchestrator Self-Service user interface.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service user interface.
2. Go to **CONFIGURATION** > **Actions Registry**.
3. Click **vCenter Details**.
4. Select your vCenter.
5. From the **Actions** menu, click **Delete vCenter** .

   A message is displayed confirming the success of the delete operation.
6. Click **OK**
7. Go to **REQUESTS** to confirm the status of the operation in the page.

Deploy single virtual server
From IBM Cloud Orchestrator Self-Service user interface, you can deploy virtual machine. If there are any default ISO drives attached to the virtual machine, then they are detached automatically during the deployment.

Before you begin

- A vCenter connection must exist. For details to add a vCenter connection, see "Add vCenter Details" on page 218.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service user interface.
2. Go to **SELF-SERVICE CATALOG** and select **Deploy cloud services**.
3. Click to open **Deploy single virtual server** offering.
4. In the **Deploy single virtual server** page, select your region and click **OK**.
5. Enter the following details:
   a. Enter the **Server Name** of the server.
   b. **Image**, **Availability Zone**, and **Flavor** - Select the image, Availability zone, and flavor from the drop-down list.
   c. Select a network and authentication mechanism.
6. Click **Attach Volume** to attach a volume to the virtual server.
7. Click **Deploy**.
8. Go to **REQUESTS** to confirm the status of the operation.

   After the deployment is successful, a script is triggered to detach CD and DVD drives. To verify in VMware vCenter server, do the following steps:

   a. Log in to VMware vCenter server.
   b. From the **Inventory** panel, open your virtual server.
   c. In the **Summary** tab, check whether the ISO drives are in disconnected state.

Deploy cloud service using heat template
From IBM Cloud Orchestrator Self-Service user interface, you can deploy virtual machines using heat template. If there are any default ISO drives attached to the virtual machine, then they are detached automatically during the deployment.

Before you begin

A vCenter connection must exist. For details to add a vCenter connection, see "Add vCenter Details" on page 218.

Procedure

1. Log in to IBM Cloud Orchestrator Self-Service user interface.
2. Go to **SELF-SERVICE CATALOG** and select **Deploy cloud services**.
3. Click to open **Deploy virtual machine using heat template** offering.
4. In the **Deploy virtual machine using heat template** page, select your region and click **Next**.
5. Enter the template text or select a template.
6. Click **Next**.
7. In the **Launch Heat Template** page, enter the following details:
   a. Enter the **Stack Name**.
   b. Enter the **Timeout** value in minutes.
   c. If required, select **Rollback on failure**.

8. Click **Stack Details** to view details of the stack.
9. Click **Deploy**.
10. Go to **REQUESTS** to confirm the status of the operation.

    After the deployment is successful, a script is triggered to detach CD and DVD drives. To verify in VMware vCenter server, do the following steps:

    a. Log in to VMware vCenter server.

    b. From the **Inventory** panel, open your virtual server.

    c. In the **Summary** tab, check whether the ISO drives are in disconnected state.

Developer reference
There are a number of scenarios that are immediately available. These services can be used as starting points and as samples for developing new content. You can clone and adapt them to better fit your needs.

This document provides the functionality of the following building blocks:

- "Human services" on page 221
- "Business processes" on page 221
- "Integration services" on page 222

For more information about working with these building blocks, see Developing IBM Cloud Orchestrator content section.

Business processes
These business processes collects the information from the human service and passes it on to the appropriate integration service.

Add vCenter Details

This process collects details of the vCenter to create a connection.

Delete vCenter Details

This process collects the name of the selected vCenter connection to delete.

Update vCenter Details

This process collects the modified details of the vCenter connection to update.

Show vCenter Details

This process collects the name of the selected vCenter connection to display the details.

Human services
These human Service artifacts are user interfaces to collect information from the end-user.

Add vCenter Details

This human service is an entry point user interface for collecting details about the vCenter connection.

Show vCenter Details

This human service collects the name of the vCenter connection and displays the details.

Update vCenter Details

This human service is an entry point user interface for collecting modifications made to a vCenter connection.

Integration services
The `detachISOImage` integration service detaches any default ISO drives that got attached to the virtual machine during deployment. It is also used to detach ISO drives from virtual machines that got deployed by using the heat template. The ISO drive is deleted from the datastore after it is detached.

Working with Heat templates and stacks

As an End User, you can deploy Heat templates and manage the related Heat stack instances.

IBM Cloud Orchestrator supports:

- OpenStack Heat Orchestration Templates (HOT), called Heat templates.
- OpenStack Heat stacks, called Heat stacks or Stacks, that are instances of deployed Heat templates.

As a Service Designer, you can create and manage Heat templates. For more information, see "Managing Heat templates" on page 232.

Deploying a Heat template

The IBM Cloud Orchestrator Self-Service Catalog provides a built-in offering that you can use to deploy an OpenStack Heat template.

Before you begin

The Heat template must be a valid Heat Orchestration Template (HOT), as defined in the OpenStack HOT specification. All resources that are referenced in the Heat template must be defined in the OpenStack environment:

- Images must be stored in OpenStack Glance.
- Flavors and networks must be defined in OpenStack.
- Keys must be registered in a project.

For tasks like processing user data or metadata, the image must be prepared to include the `cloud-init` package and the `heat-cfntools` package. For more information about creating images, see "Creating base images" on page 239.

Note: If you want to deploy a complex Heat template (for example, templates with `WaitCondition` and `waithandle` or templates with `SoftwareDeployment` resource), perform the following steps:

1. Set the option `deferred_auth_method=trusts` in the configuration file of the Heat engine (`/etc/heat/heat.conf`) and restart the Heat engine by running the `service openstack-heat-engine restart` command.
2. Create a `heat_stack_owner` role in Keystone.
3. Assign the `heat_stack_owner` role to the user.

Note: For AIX images, you cannot change the password of a user or inject an SSH key at deployment time.

Procedure

1. Log in to the Self-service user interface as an End User.
2. Click **SELF-SERVICE CATALOG**.
3. Click **Deploy cloud services** > **Deploy cloud service using Heat template**. The **Deploy a cloud service by using stacks** page opens.
4. If there are more than one region in your environment, select a region and click **Next**.
5. Select **Direct Input** if you want to enter the template text directly in the **Enter Heat Template** field, or select **Stored Template** to select a Heat template from the list of the templates available in your environment. For information about Heat template format and specification, see "Heat template examples" on page 234.

   Note: When you write the Heat stack template, use the forward slash character (/) instead of the backslash character (\) in the `user_data` section (for example, in path names).

A Heat stack template is written in YAML format, and is translated into a JSON object when the self-service offering is processed by Business Process Manager. If you use the backslash character and the target system is a Microsoft Windows system, the YAML-to-JSON parser inserts a second backslash character as an escape character. If the escaped backslash character (\\) is combined with a newline character (\n), the YAML code cannot be processed by the parser. To work around this problem, use the forward slash character instead. Microsoft Windows can process path names that contain forward slashes, if the path names do not include any spaces.

6. Click **Next**. The **Launch Heat Template** page opens.
7. In the **Stack Name** field, specify the name of the Heat stack instance to be deployed.
8. Specify the timeout value for the deployment. If the stack is not deployed within the specified time, an error message is displayed.
9. Optional: If you want to roll back the Heat instance if the stack fails to deploy, select the **Rollback on failure** check box.
10. If the template contains parameter definitions, each parameter name, description, and value is listed in the Parameters table.

    For each parameter, specify the parameter value:

    - Default parameter values might be provided in the parameter definition in the template.
    - If a parameter description is prefixed with the name of a supported lookup annotation, you can select the parameter value from a list in the **Select Value** column.
    - Otherwise, you must type the parameter value in a field in the **Enter Value** column.
11. Optional: To modify the Heat stack resources, click **Stack Details**:
    a. Select the resource that you want to modify.
    b. To view details of the volumes that are attached to the selected resource, click **Volumes**. To attach a volume to the selected resource, click **Add Volume**, specify the volume size and mount point, and click **OK**.
    c. To view details of the networks that are attached to the selected resource, click **Network Interfaces**. To attach a network to the selected resource, click **Add Network Interface**, specify the network name and fixed IP address, and click **OK**.
    d. To return to the **Launch Heat Template** page, click **OK**.
12. Click **OK**. A REST call is posted to the OpenStack Heat engine, and the Heat template is deployed.
13. Monitor the status of your deployment request, as described in "Viewing the status of your requests and actions" on page 211.

    Tip: If a problem occurs while you are deploying a Heat template, check the following log files for detailed information:

    - In the Business Process Manager server:

      ```
      /opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/logs/SingleClusterMember1/SystemOut.log
      ```

    - In the virtual machine where Heat is installed:

      ```
      /var/log/heat/api.log
      /var/log/heat/engine.log
      ```

What to do next

You can manage the deployed Heat stack. For information, see "Managing Heat stacks" on page 224.

Related tasks

"Managing Heat templates" on page 232

You can manage Heat templates that can be selected and deployed from the Self-Service Catalog.

Managing Heat stacks

You can use the Self-service user interface to manage deployed Heat stacks.

Procedure

1. Log in to the Self-service user interface as an End User.
2. Click **RESOURCES** > **Stacks**.

   The page shows a list of deployed Heat stacks, with an **Actions** menu to the left of the list.

   If you select one or more Heat stacks in the list, the **Actions** menu is updated to show only the actions that you can apply to the selected Heat stacks.
3. To show more details about a Heat stack, click the Heat stack name in the instance list.

   The **Heat Stack Details** page is displayed. The details page also displays a list of the virtual machine instances that are associated with the Heat stack. The **Actions** menu is updated to show only the actions that you can apply to the selected Heat stack.

Related tasks

“Applying an action to a resource” on page 212
Use the **RESOURCES** tab to select an action for an instance, for example, to start or stop a virtual machine.

Managing key pairs

You can use offerings in the Self-Service Catalog to manage key pairs.

Registering a key pair

The Self-Service Catalog provides a self-service offering to register a public or private key pair in the context of a project. The key can be used to access virtual machines securely without using user ID and password. All users of the project can see and use this key pair.

Procedure

1. Log in to the Self-service user interface as a Cloud Administrator.
2. In the menu bar, click **SELF-SERVICE CATALOG**.
3. Click **Deploy cloud services** > **Create or register key** to enable access to virtual servers.
4. Enter a **Project name**, **Key Name**, **Public Key**, and **Private key** in the fields provided.

   Note: If you click **Generate Public And Private Key**, a public or private key pair is generated and displayed in the corresponding fields of the panel.

   Note: [For PowerVC only:] You must specify the keys that you generated by following the procedure described in “Setting up PowerVC RSA keys” on page 172.
5. Click **OK**.

Results

A message appears indicating that a key was registered successfully.

Unregistering a key pair

The Self-Service Catalog provides a self-service offering to unregister a public or private key pair that is defined in the context of a project.

About this task

You cannot unregister a key pair if a virtual machine exists that has the key pair defined.

Procedure

1. Log in to the Self-service user interface as a Cloud Administrator.

2. In the menu bar, click **SELF-SERVICE CATALOG**.
3. Click **Deploy cloud services** > **Unregister key**.
4. Enter a name for the project.
5. A table shows all the key pairs that are defined in the context of the project. Select one or multiple key pairs to be unregistered by selecting the check box beside the key pair.
6. Click **OK**.

Results

A message appears indicating that the key pairs you selected are now unregistered and no longer available to be selected during provisioning of a virtual machine.

Working with volumes

The IBM Cloud Orchestrator provides a self-service offering and actions to manage volumes using the OpenStack Cinder component.

It also provides options for mounting and formatting the attached volumes on the operating system level.

Creating storage volumes

1. Navigate to **SELF-SERVICE CATALOG** > **Deploy cloud services** > **Create storage volume** for creating new volumes.
2. Select a region and availability zone, then specify the volume name and the size of the volumes.
3. You can create either one or multiple volumes at the same time by specifying the number of the volumes to be created in the **Instances** field. Optionally, add a description for the volumes.

Attaching a volume

1. Navigate to **RESOURCES** > **Volumes**.
2. In the table, select a volume that is in `available` status.
3. In the **Actions** menu to the left of the table, click **Attach volume**.
4. Select a server.
5. Specify whether the volume must be formatted on the operation system level. If yes, select the filesystem. Specify whether the formatted volume must be mounted on the server. If yes, mention a mount point or drive letter. If neither format nor mount option is selected, the volume gets attached only to the server.

   Note: Already formatted volumes can be reformatted and mounted on the selected server. A raw volume cannot be mounted without formatting.

   The format operation requires the following additional parameters: the operation system type, the file system type, the mount point, a user, and the related credentials for accessing the server to run the format operation. The given user must have the rights to perform the operations for partitioning, formatting, and mounting the new volume on the operating system.

   Note the following restrictions for the format operation:

   - Do not run format operations on the same server in parallel.
   - If the attached volume cannot be detected on the operating system level after 5 minutes, the operation stops with an error.
   - If a non-root user is provided to partition, format, and mount a volume, then sudo is used to run the commands elevated. Ensure that the non-root user can perform password-less sudo actions.
   - [For Windows instances:] Initially, only one disk (Disk 0) must be specified. The format operation works on Disk1, Disk2, and so forth, for each newly attached volume. This means it would erase data on these disks if you would initially start with more than one disk.
   - [For Windows instances:] The operating system must be enabled for RXA access as described in Requirements for using Remote Execution and Access (RXA).

- [For Windows instances:] Even when the volume is only opted to be formatted and not mounted, the operating system may assign a drive letter. By default, the automount is enabled. You can disable it using this command - **`diskpart automount disable`**.
- [For Windows instances:] Volume can be mounted on a non-existing drive but not on a directory in the non-existing drive. For example, `P:\` is valid but not `P:\cinder_vol`.
- [For AIX instances:] The format operation is not supported.
- Custom scripts can be used to partition, format, and mount a volume. For more information, see *Registering scripts for storage volumes section of Content Development Guide*.
- The format and mount options are not supported whenever you attach a volume by using PowerVC.

If you have any issue during the format operation, see the Business Process Manager log file, `/opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/logs/SystemOut.log`, for troubleshooting information.

Note: If the attachment of the new volume does not terminate within 30 minutes, the action stops with an error.

Note: If a volume is attached to a multidisk image, the device data associated to the volume might not display correctly in the Self-service user interface.

Deleting volumes

1. Navigate to **RESOURCES** > **Volumes**.
2. In the table, select one or more volumes that are in `available` status.
3. In the **Actions** menu to the left of the table, click **Delete volumes**.

Note: Do not run this action for formatted volumes if there are pending attach or detach actions in place for these volumes.

Detaching volume

1. Navigate **RESOURCES** > **Volumes**.
2. In the table, select one volume that is in `in-use` status.
3. If the volume was formatted and mounted on the operating system, the following parameters for unmounting the volume are required: the operating system type, a user name, and the related credentials for accessing the system. The specified user must have the rights to unmount the volume on operating system level.

Note: In Amazon EC2, you cannot detach a mounted volume. Make sure to unmount the volume before detaching it.

Note: Because of an operating system limitation, after you detach a volume from a SLES instance on VMware, you must reboot the SLES instance to complete the detach operation.

Managing the Inbox

Use the **INBOX** tab to view and process pending tasks and approvals.

Viewing the Inbox

The Inbox lists the tasks and approvals that are currently waiting to be claimed.

To view the Inbox, complete the following steps:

1. Log in to the Self-service user interface as an End User.
2. Click the **INBOX** tab.

If new tasks or approvals are created, the notification number increases and is displayed in the **INBOX** tab. The notifications are updated every minute. The **INBOX** notifications have three common states:

- The notifications number increases when:

  - A user clicks **Reassign Back** on one or many of the tasks or approvals.
  - New tasks or approvals are generated. After approximately 1 minute, the system sees these unassigned tasks and approvals and increases the notifications number.

  Note: If there are more than 100 approval requests waiting to be claimed, the notifications number is displayed as **100+**.

- The notifications number decreases when:
  - A user claims one or many of the tasks or approvals. After approximately 1 minute, the system sees that the tasks or approvals have been claimed and decreases the notifications number.
- No notifications number is displayed when:
  - There are no tasks waiting to be claimed by the user.

Processing an Inbox assignment

When a submitted request requires any user interaction, such as an approval, or providing extra parameters, the responsible users get an assignment in their Inbox.

Procedure

1. Log in to the Self-service user interface.
2. Click the **INBOX** tab to display a list of assignments that require user interaction.

   The following types of assignments are displayed:

   - ✓ Approval request
   - ⚙ General task

   To view the details of an assignment, you must first claim the assignment. Then click the assignment icon to display the assignment details.
3. The status of the assignment is indicated by the button that is displayed:
   - If the assignment is not currently claimed by any user, the **Claim** button is displayed. To take the ownership of the assignment, click **Claim**.
   - If you claimed the assignment, the **Reassign Back** button is displayed. To release the assignment and allow another user to claim it, click **Reassign Back**.
4. To complete an assignment that you claimed, perform the following steps:

   a) Click the assignment icon to view the assignment details.

   b) To complete a general task, enter any information that is required and click **Submit**.

   c) To complete an approval request, click **Accept** or **Reject**. You can optionally enter a reason.

   A completion message is displayed and the assignment is deleted from the **INBOX** tab.

Designing self-service

A Service Designer can manage the artifacts in the Self-Service Catalog, and use them to customize the IBM Cloud Orchestrator environment. A Service Designer is a user with the **catalogeditor** role.

Self-Service Catalog default contents

IBM Cloud Orchestrator provides a set of default offerings and categories in the Self-Service Catalog to help you to manage cloud services and deploying virtual systems. You can modify the catalog to add, modify, and remove offerings according to your needs.

In the Self-service user interface, click **SELF-SERVICE CATALOG** to view the categories available in the Self-Service Catalog and the related description. Click a category to view the associated offerings.

Self-Service Catalog population tool

The Self-Service Catalog population tool is used to create categories and offerings in the catalog using an XML file as input.

For every content pack released by IBM, a specific XML file is provided. You can use the file to automate the IBM Cloud Orchestrator catalog update to add the categories and offerings delivered with the content pack.

To use the Self-Service Catalog population tool run the following script that is stored in the `/opt/ibm/ico/ccs/catalog` directory:

```
catalogTool.sh <xml_file_name_for_the_content_pack> <cloud_admin_user> <cloud_admin_pwd>
```

Managing offerings

A Service Designer can managing offerings and their access control list in the Self-Service Catalog.

In the Self-service user interface, click **CONFIGURATION** > **Self-Service Catalog** in the navigation menu, and then click **Offerings**. You can search for an offering by specifying the offering name or description in the search field. The offering table can be sorted using any column that has the sort icon.

If you select one or more offerings in the table, the **Actions** menu is updated to show only the actions that you can apply to the selected offerings.

Depending on your permissions, you can perform the following actions:

Create an offering
See "Creating an offering" on page 228.

Edit an offering
Select an offering in the table and click **Edit Offering**.

Delete offerings
Select one or more offerings in the table and click **Delete Offering**.

Modify the access control list of an offering
See "Modifying the access control list of an offering" on page 229.

Creating an offering

You can create a new offering in a domain.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. In the navigation menu, click **CONFIGURATION** > **Self-Service Catalog**.
3. Click **Offerings** in the menu below the navigation menu.
4. Click **Create Offering** in the **Actions** menu. The **Create Offering** window is displayed.
5. Enter a name for the offering.

6. Select an icon and a category for the offering.
7. Enter a description for the offering.
8. Select a process, application and human service for the offering.

   To find the process, select the application to filter the processes by that application. After the process is found, select the user interface from the list of available human services for the selected process. Configure the access control. By default any user in the same domain can use the offering. The Domain Administrator and the Service Designer in the domain are allowed to modify the offering.
9. Click **Create**.

Results

A message appears indicating that the offering is created successfully.

Modifying the access control list of an offering

You can modify the access control list of an offering by adding or removing access.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. In the navigation menu, click **CONFIGURATION** > **Self-Service Catalog**.
3. Select an offering and click **Modify Access Control List** in the **Actions** menu. The **Modify Access Control List** window appears displaying the list of the roles in the specified domain and project that have access rights to the offering.
4. You can perform the following actions:
   - To create a new entry in the list, specify the a domain, a project, a role, and select the appropriate access rights. Click **Add to Access Control List**.
   - To remove an access control entry from the list, click the related **Delete** icon.
5. Click **Save**.

Managing categories

A Cloud Administrator can manage categories in the Self-Service Catalog.

In the Self-service user interface, click **CONFIGURATION** > **Self-Service Catalog** in the navigation menu, and then click **Categories**. You can search for a category by specifying the category name or description in the search field. The category table can be sorted using any column that has the sort icon.

If you select one or more categories in the table, the **Actions** menu is updated to show only the actions that you can apply to the selected categories.

You can perform the following actions:

Create a category

See "Creating a category" on page 229.

Edit a category

Select a category in the table and click **Edit Category**.

Delete categories

Select one or more categories in the table and click **Delete Category**.

Creating a category

You can create a new category in a domain.

Procedure

1. Log in to the Self-service user interface as a Cloud Administrator.
2. In the navigation menu, click **CONFIGURATION** > **Self-Service Catalog**.
3. Click **Categories** in the menu below the navigation menu.

4. Click **Create Category** in the **Actions** menu. The **Create Category** window is displayed.
5. Enter a name for the category.
6. Select an icon for the category.
7. Enter a description for the category.
8. Click **Create**.

Results

A message appears indicating that the category is created successfully.

Managing actions

A Service Designer can manage actions and their access control list in the Actions Registry.

In the Self-service user interface, click **CONFIGURATION** > **Actions Registry** in the navigation menu to manage actions. You can search for an action by specifying the action name or description in the search field. The action table can be sorted using any column that has the sort icon.

If you select one or more actions in the table, the **Actions** menu is updated to show only the actions that you can apply to the selected actions.

Depending on your permissions, you can perform the following actions:

Create an action
See "Creating an action" on page 230.

Edit an action
Select an action in the table and click **Edit Action**.

Delete actions
Select one or more actions in the table and click **Delete Action**.

Modify the access control list of an action
See "Modifying the access control list of an action" on page 231.

Creating an action

You can create a new action in a domain.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. In the navigation menu, click **CONFIGURATION** > **Actions Registry**.
3. Click **Create Action** in the **Actions** menu. The **Create Action** window is displayed.
4. Enter a name for the action.
5. Select an icon and a process for the action.
6. Enter a description for the action.
7. Select the type of resource the action applies to and include the tags you want the action to apply to.

   You must specify which instance the action applies to. Based on the selection of the type, choose from a list of tags that the instance might have. The action only appears on instances having the type and tag. The field **Specify the item selection criteria** allows you to specify whether the action is able to:

   - Create an instance. Select **createInstanceAction**.
   - Modify only a single instance. Select **singleInstanceAction**.
   - Modify multiple instances. Choose **multiInstanceAction**.

   Tags are working as an extra filter mechanism for actions to be performed on selected instances. As an example take the `Start` action for virtual servers. It has the tag `shutoff`. This means that the `Start` action is only be available for virtual servers (**openstackvms**) that are stopped. Those tags can be set during action creation or modification in the Actions Registry depending on the selected instance type (for example, **openstackvms**). The following tags are provided by IBM Cloud Orchestrator through their instance providers:

For virtual machines (openstackvms):

shutoff: stopped virtual machine.

active: running virtual machine.

nova: virtual machine that is created directly in OpenStack.

heat: virtual machine that is created through a Heat template.

keynamedefined: virtual machines having an SSH key defined for access.

For Stacks (heat):

createcomplete: Heat stack instances that are created and are ready to use.

createfailed: Heat stack instances that failed to be created.

For Volumes (volumes):

available: volumes ready for use.

cinder: all volumes that are created as OpenStack Cinder volumes.

in-use: volumes that are already occupied or used by virtual machines.

formatted: volumes that are formatted.

For Domains (domain):

disabled: domains not ready for use (disabled).

enabled: domains that are enabled for use.

For Projects (project):

disabled: projects not ready for use (disabled).

enabled: projects that are enabled for use.

For Users (user):

disabled: users not ready for use (disabled).

enabled: users that are enabled for use.

For Offerings (offering):

offering: a resource that is of type offering.

For Action Registry (action):

multiInstanceAction: actions performed on multiple instances.

singleInstanceAction: actions performed on a single instance.

For Categories (categories):

Not applicable.

8. Select the application to filter the processes by that application. Once the process has been found, select the user interface from the list of available human services for the selected process. Then, configure the access control. The Domain Administrator and the Service Designer are allowed to modify the offering.
9. Click **Create**.

Results

A message appears indicating that the action is created successfully.

Modifying the access control list of an action

You can modify the access control list of an action by adding or removing access.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. In the navigation menu, click **CONFIGURATION** > **Actions Registry**.
3. Click **Modify Access Control List** in the **Actions** menu. The **Modify Access Control List** window appears displaying the list of the roles in the specified domain and project that have access rights to the action.
4. You can perform the following actions:

- To create a new entry in the list, specify the a domain, a project, a role, and select the appropriate access rights. Click **Add to Access Control List**.
- To remove an access control entry from the list, click the related **Delete** icon.

5. Click **Save**.

Managing Heat templates

You can manage Heat templates that can be selected and deployed from the Self-Service Catalog.

Before you begin

The Heat template must be a valid Heat Orchestration Template (HOT), as defined in the OpenStack HOT specification. All resources that are referenced in the Heat template must be defined in the OpenStack environment:

- Images must be stored in OpenStack Glance.
- Flavors and networks must be defined in OpenStack.
- Keys must be registered in a project.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. Click **CONFIGURATION** > **Templates**, and then click **Heat Templates**. The list of the Heat templates is displayed. You can click a template to view the template details.

   If you select one or more Heat templates in the list, the **Actions** menu is updated to show only the actions that you can apply to the selected Heat templates.

   Depending on your permissions, you can perform the following actions:

   - **Creating a Heat template**

     See "Creating a Heat template" on page 233.

   - **Importing a Heat template**

     Click **Import Heat Template** to import a template from a local file.

     If the Heat template content is not valid, an error message is displayed and the imported file is shown in the Heat template edit view where you can correct the Heat template or cancel the import. For information about Heat template format and specification, see "Heat template examples" on page 234.

     By default, the imported template can be deployed to any region and the users with Domain Administrator or Service Designer role in your domain have full control permission on the template. You can change these settings by editing the Heat template after you imported it.

   - **Editing a Heat template**

     Select a template and click **Edit Heat Template**.

     Note: When you edit a Heat template, ensure no other user is editing the same Heat template at the same time, otherwise your changes might be overwritten.

   - **Deleting Heat templates**

     Select one or more templates and click **Delete Heat Template**.

   - **Modifying the access control list of a Heat template**

     See "Modifying the access control list of a Heat template" on page 233.

Related tasks

"Deploying a Heat template" on page 222

The IBM Cloud Orchestrator Self-Service Catalog provides a built-in offering that you can use to deploy an OpenStack Heat template.

Creating a Heat template

You can create a new Heat template to be deployed in your environment.

Before you begin

The Heat template must be a valid Heat Orchestration Template (HOT), as defined in the OpenStack HOT specification. All resources that are referenced in the Heat template must be defined in the OpenStack environment:

- Images must be stored in OpenStack Glance.
- Flavors and networks must be defined in OpenStack.
- Keys must be registered in a project.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. Click **CONFIGURATION** > **Templates**, and then click **Heat Templates**. The list of the Heat templates is displayed.
3. Click **Create Heat Template** in the **Actions** menu.
4. Specify a name for the new Heat template.
5. In the **Heat Template Source** tab, enter the source template in text format. For information about Heat template format and specification, see "Heat template examples" on page 234.
6. In the **Deployment Details** tab, specify whether the template can be deployed to any region (default) or only to the regions that you select from the list of the available regions.
7. In the **Access Control List** tab, you can add entries to the access control list by defining the scope (domain, project, and role) and the access rights (view, deploy, full control) and by clicking **Add to Access Control List**.

   Only users with the specified role in the project and domain can access the Heat template, where:

   view
   Specifies if the users can see the Heat template.

   deploy
   Specifies if the users can view and deploy the Heat template.

   full control
   Specifies if the users can view, deploy, and modify the Heat template.

   You can also modify the access rights of the existing default entries or remove them from the list. Even if an access control list is empty, the cloud administrator is still allowed to manage the Heat template.
8. Click **Save** to create the new Heattemplate.

Related tasks

"Deploying a Heat template" on page 222
The IBM Cloud Orchestrator Self-Service Catalog provides a built-in offering that you can use to deploy an OpenStack Heat template.

Modifying the access control list of a Heat template

You can modify the access control list of a Heat template by adding or removing access.

About this task

The access control list defines the scope (domain, project, and role) and the access rights (view, deploy, or full control). Only users with the specified role in the project and domain can access the Heat template, where:

view
Specifies if the users can see the Heat template.

deploy
Specifies if the users can view and deploy the Heat template.

full control
Specifies if the users can view, deploy, and modify the Heat template.

Even if an access control list is empty, the cloud administrator is still allowed to manage the Heat template.

Procedure

1. Log in to the Self-service user interface as a Service Designer.
2. In the navigation menu, click **CONFIGURATION** > **Templates**, and then click **Heat Templates**.
3. Select a template and click **Modify Access Control List** in the **Actions** menu. The **Modify Access Control List** window appears displaying the list of the roles in the specified domain and project that have access rights to the Heat template.
4. You can perform the following actions:
    - To create a new entry in the list, specify a domain, a project, a role, and select the appropriate access rights. Click **Add to Access Control List**.
    - To remove an access control entry from the list, click the related **Delete** icon.
5. Click **Save**.

Heat template examples

A Heat template is a valid Heat Orchestration Template (HOT), as defined in the OpenStack HOT specification.

For detailed information about the Heat Orchestration Templates, see the OpenStack *Template Guide* at http://docs.openstack.org/developer/heat/template_guide/. In the guide, you can find the following information:

- The introduction and some basic examples at http://docs.openstack.org/developer/heat/template_guide/hot_guide.html
- The Heat Orchestration Template specification at http://docs.openstack.org/developer/heat/template_guide/hot_spec.html
- The OpenStack Resource Types and the related parameters at http://docs.openstack.org/developer/heat/template_guide/openstack.html

The Heat Orchestration Templates are under development, so the OpenStack *Template Guide* is periodically updated by the community.

When developing a template, it is recommended to use parameters and to avoid hardcoded values.

In the following examples, “Example 1” on page 235 and “Example 2” on page 235 are taken from the OpenStack *Template Guide* and show the differences in using hardcoded values or parameters.

“Example 3” on page 235 shows how to use lookup annotation to generate a list of possible values for a parameter, which helps the user to select a valid parameter value. The following lookup annotations are supported:

SCOIMAGE
Lookup of images from the image repository for the region.

SCOFLAVOR
Lookup of flavor size from the selection available in the region.

SCONETWORK
Lookup of available networks in the region.

SCOKEY
Lookup of the registered keys for the project.

"Example 4" on page 236 shows how to set the `admin` password for a virtual machine by using the `user_data` section.

"Example 5" on page 236 shows how to deploy an AIX or Linux on Power® server on PowerVC where you specify the Storage Connectivity Group to use and the Storage Template on which the boot volume is placed.

Note: Heat Orchestration Templates are sensitive to formatting issues. To avoid template validation errors, use the correct indentation.

Example 1

The following example is a simple Heat template to deploy a single virtual system and it is limited to a single combination of image, key, and flavor values that are hardcoded in the template:

```
heat_template_version: 2013-05-23

description: Simple template to deploy a single compute instance with hardcoded values

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: my_key_pair_1
      image: cirros-0.3.1-x86_64
      flavor: m1.tiny
```

Example 2

The following example is a Heat template to deploy a single virtual system with parameters and it is therefore reusable for other configurations:

```
heat_template_version: 2013-05-23
description: Simple template to deploy a single compute instance with parameters

parameters:
  key_name:
    type: string
    label: Key Name
    description: Name of key-pair to be used for compute instance
  image_id:
    type: string
    label: Image ID
    description: Image to be used for compute instance
  instance_type:
    type: string
    label: Instance Type
    description: Type of instance (flavor) to be used
resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: { get_param: key_name }
      image: { get_param: image_id }
      flavor: { get_param: instance_type }
```

Example 3

The following example is a simple Heat template to deploy a stack with two virtual machine instances by using lookup annotations for parameters:

```
heat_template_version: 2013-05-23

description: Simple template to deploy a stack with two virtual machine instances

parameters:
  image_name_1:
    type: string
    label: Image Name
    description: SCOIMAGE Specify an image name for instance1
    default: cirros-0.3.1-x86_64
  image_name_2:
    type: string
    label: Image Name
    description: SCOIMAGE Specify an image name for instance2
    default: cirros-0.3.1-x86_64
  network_id:
    type: string
    label: Network ID
    description: SCONETWORK Network to be used for the compute instance

resources:
  my_instance1:
    type: OS::Nova::Server
    properties:
      image: { get_param: image_name_1 }
      flavor: m1.small
      networks:
        - network : { get_param : network_id }
  my_instance2:
    type: OS::Nova::Server
    properties:
      image: { get_param: image_name_2 }
      flavor: m1.tiny
      networks:
        - network : { get_param : network_id }
```

Example 4

The following example is a simple Heat template to set the `admin` password for a virtual machine by using the `user_data` section:

```
heat_template_version: 2013-05-23

description: Simple template to set the admin password for a virtual machine

parameters:
  key_name:
    type: string
    label: Key Name
    description: SCOKEY Name of the key pair to be used for the compute instance
  image_name:
    type: string
    label: Image Name
    description: SCOIMAGE Name of the image to be used for the compute instance
  password:
    type: string
    label: password
    description: admin password
    hidden: true


resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: { get_param: key_name }
      admin_user: sampleuser
      image: { get_param: image_name }
      flavor: m1.small
      user_data:
        str_replace:
          template: |
            #!/bin/bash
            echo "Setting  password to  " $password
            echo $password |passwd --stdin sampleuser

          params:
            $password: { get_param: password }
```

Example 5

The following example is a simple Heat template to deploy an AIX or Linux on Power server on PowerVC where you specify the Storage Connectivity Group to use and the Storage Template on which the boot volume is deployed:

```
heat_template_version: 2013-05-23
description: Template to Deploy on NPIV v7k storage only
```

```
parameters:
  network_id1:
    type: string
    description: SCONETWORK ID of the (nova) network a server should be deployed to.
  flavor_id:
    type: string
    description: SCOFLAVOR The flavor to be applied to the server DatabaseTierVM.
  image:
    type: string
    label: Image
    description: SCOIMAGE The Image to be deployed
resources:
  heat:
    type: OS::Nova::Server
    properties:
      image: { get_param: image }
      flavor: { get_param: flavor_id }
      availability_zone: D0EB
      metadata: { selected-scg: d91acbbe-3d81-4279-b389-54b3ad4a1c8c,
 selected-storage-template: 0431b2f3-fea6-4aa5-b3fb-d0e82ccf5ebb }
      networks:
         - network : { get_param : network_id1 }
```

Note: You can create the availability zones for a PowerVM® server by using the **Host Aggregates** panel in the OpenStack Dashboard. Ensure that, if you created new availability zones, they are then added to the relevant domains and projects before attempting to use them. `selected-scg` (Storage Connectivity Group) and `selected-storage-template` can be found from the `Image_topology` of the image you are planning to use. On the OpenStack Controller, run the **`glance image-show <image id>`** command. Image Topology specifies which Storage Connectivity Group and which Storage Templates are supported by that specific image.

Related information

OpenStack Heat Orchestration Template (HOT) Specification

OpenStack Heat Orchestration Template (HOT) Guide

OpenStack Building JEOS images for use with Heat

Chapter 9. Managing virtual images

You can manage virtual images that can be deployed by using IBM Cloud Orchestrator.

A base image that can be deployed through IBM Cloud Orchestrator is an image that can be deployed through OpenStack. For this reason, such images are also described as *OpenStack-ready images*. For more information about base images, see "Creating base images" on page 239.

This kind of image is suitable for single instance deployments and deployment of OpenStack Heat stacks.

Creating base images

You can create images suitable for single instance deployments or deployment of OpenStack Heat stacks.

To create these images, follow the instructions in the Create images manually chapter in the OpenStack Virtual Machine Image Guide.

After the operating system is installed, you must install additional software for customizing the instances deployed by using this image, by performing one of the following procedures:

- "Creating Windows base images" on page 239
- "Creating Linux base images" on page 241
- "Creating base images for Linux on System z" on page 243

For information about creating images for Amazon EC2 and SoftLayer through the Public Cloud Gateway, see "Creating a supported image" on page 261.

Creating Windows base images

You can create Windows base images by running the following procedures.

1. "Adding cloudbase-init to Windows images" on page 239.
2. "Installing virtio driver (KVM hypervisor only)" on page 240.
3. "Running sysprep.exe" on page 240.
4. If you want to execute scripts on the deployed virtual machine, you must enable RXA by following the procedure described in Requirements for using Remote Execution and Access (RXA). For more information about executing scripts on virtual machine instances, see "Managing virtual machine instances" on page 215.
5. In Windows 2016, the SMBv3 is enabled and SMBv1 is turned off by default, hence the ability to make connections without secure negotiation is disabled in IBM Cloud Orchestrator. For steps to re-enable SMBv1 on Windows 2016 image, see https://support.microsoft.com/en-us/help/2696547/how-to-detect-enable-and-disable-smbv1-smbv2-and-smbv3-in-windows-and.

Note: Port 445 on the default security group must be enabled to use Remote Execution and Access (RXA) in a KVM region.

Run the following on the KVM region controller:

```
nova secgroup-add-rule default tcp 445 445 0.0.0.0/0
```

Adding cloudbase-init to Windows images

You can add `cloudbase-init` to Windows operating system images.

To add `cloudbase-init` to your image, download it from https://cloudbase.it/cloudbase-init/#download and install it by following the procedure at http://www.cloudbase.it/cloud-init-for-windows-instances/.

Note:

- If you want to deploy Windows images, use the latest version of the `cloudbase-init` tool while creating the Windows image to support the OpenStack updates related to the network interface management of Windows images.
- After the `cloudbase-init` installation, do not select the option to run `sysprep.exe` in the **Finish** page.
- When you create the network, set the `dns-nameservers` and `gateway` parameters.
- To speed up the IP address injection, when you create the image template, specify the `metadata_services` parameter in the `cloudbase-init.conf` file:

```
metadata_services= cloudbaseinit.metadata.services.configdrive.ConfigDriveService,
                   cloudbaseinit.metadata.services.httpservice.HttpService,
                   cloudbaseinit.metadata.services.ec2service.EC2Service,
                   cloudbaseinit.metadata.services.maasservice.MaaSHttpService
```

- If you get the `OS can not be restarted automatically` message after changing the host name, use the latest `cloudbase-init` version.
- You can configure `cloudbase-init` to set the password for a user. The user name is configured at image preparation time and cannot be modified at virtual machine creation time. You can specify a user name during `cloudbase-init` installation or in the `cloudbase-init.conf` file. If the user does not exist, a new user account is created at virtual machine initialization time. If there are multiple Windows users at image preparation time, at virtual machine initialization time the password is changed only for the user specified in the `cloudbase-init` configuration. The passwords for other users are not changed.
- If `cloudbase-init` cannot run scripts during an instance boot, set the PowerShell execution policy to be unrestricted:

```
C:\powershell
PS C:\Set-ExecutionPolicy Unrestricted
```

- `cloudbase-init` also installs the Cloud Initialization Service on your image. This service is not used by IBM Cloud Orchestrator and it can be removed after the image is correctly deployed.

Installing virtio driver (KVM hypervisor only)

To use Windows operating system images on a KVM hypervisor, install the virtio driver into the system because OpenStack presents the disk using a virtio interface while launching the instance.

You can download a `virtio-win*.iso` file containing the virtio drivers from https://fedoraproject.org/wiki/Windows_Virtio_Drivers.

Use virt-manager to connect `virtio-win*.iso` to the image and update the network adapter in the virtual machine by completing the following steps:

1. From the Control Panel, click **Device Manager**.
2. Right-click the network adapter and click **Update Driver Software** > **Browse my computer for driver software**.
3. Select the virtual CD/DVD drive and then select the inf file.
4. Restart the virtual machine.

Running `sysprep.exe`

Run `sysprep.exe` to remove all the unique system information, like computer name and hardware-specific information, from your Windows image.

To run `sysprep.exe` on Windows 2008 R2, complete the following steps. Refer to the Microsoft documentation for the other Windows platforms.

1. Download and install the Windows Automated Installation Kit (AIK). You can download Windows AIK from the Microsoft Download Center: http://www.microsoft.com/en-us/download/details.aspx?id=9085. Windows System Image Manager is installed as part of the Windows Automated Installation Kit (AIK).

2. Copy the `install.wim` file from the `\sources` directory of the Windows 2008 R2 installation DVD to the hard disk of the virtual machine.
3. Start the Windows System Image Manager.
4. In the Windows Image pane, right-click **Select a Windows image or catalog file** to load the `install.wim` file you copied.
5. When a warning that the catalog file cannot be opened is displayed, click **Yes** to create a new catalog file. Remember to select the Windows 2008 R2 Edition.
6. In the Answer File pane, right-click to create a new answer file:

   Language and Country or Region:

   a. Generate the answer file from the Windows System Image Manager by expanding Components in your Windows Image pane, right-click, and add the Microsoft-Windows-International-Core setting to Pass 7 oobeSystem.
   b. In your Answer File pane, configure the InputLocale, SystemLocale, UILanguage, and UserLocale with the appropriate settings for your language and country or region.

   Administrator Password:

   - In the Windows Image panel, expand the Microsoft-Windows-Shell-Setup component, and expand User Accounts, right-click on **AdministratorPassword**, and add the setting to the Pass 7 oobeSystem configuration pass of your answer file.
   - In the Answer File panel, specify a password next to **Value**.

     Note: You can read the AIK documentation and set more options depending on your deployment. The steps described here are the minimum needed for the Windows unattended setup.

   Software License Terms:

   In the Windows Image panel, expand Components and find the Microsoft-Windows-Shell-Setup component. Highlight the OOBE setting, and add the setting to the Pass 7 oobeSystem. In the Answer File panel, set HideEULAPage true in OOBE settings.

   Product Key and Computer Name:

   - In the Windows Image panel, right-click on the **Microsoft-Windows-Shell-Setupcomponent** and add the settings to the Pass 4 specialize configuration pass of your answer file.
   - In the Answer File panel, enter your Product Key in the space provided next to **ProductKey**. Furthermore, to automate the Computer Name Selection page, specify a computer name next to **ComputerName**.
7. Save the answer file as `unattend.xml`. Ignore the warning messages that appear in the validation window.
8. Copy the `unattend.xml` file into the `c:\windows\system32\sysprep` directory of the Windows 2008 R2 virtual machine.
9. Clean the environment of the virtual machine.
10. Uninstall Windows AIK that might not be part of the virtual machine you create.
11. Remove the `install.wim` file that was copied to the virtual machine.
12. Run the `sysprep` tool as follows:

```
cd c:\Windows\System32\sysprep
sysprep.exe /oobe /generalize /shutdown
```

The Windows 2008 R2 virtual machine shuts down automatically after `sysprep` is complete.

Creating Linux base images

Create a Linux image to be used in IBM Cloud Orchestrator.

You must install the following software to your Linux base image:

- `cloud-init`
- `heat-cfntools`

When you create a Linux base image, be sure that the image meets the following prerequisites:

- Be sure that you create a single `ext3` or `ext4` partition (not managed by LVM), otherwise you might have issues when installing `cloud-init`.
- Be sure that the image has IPv6 networking that is disabled. For more information, see the documentation that is related to your Linux distribution.
- If you use a template for a hypervisor that is not VMware, be sure that the image has a single disk. You can add extra disks at deployment time.
- If the `/etc/sudoers` file contains the following line:

  ```
  Defaults  requiretty
  ```

  you must comment it out or change it to:

  ```
  Defaults  !requiretty
  ```

- When creating a Linux image for Hyper-V, be sure that the kernel parameter `no_timer_check` is specified to the kernel parameters in the boot-loader configuration. Without this option, your image might fail to boot due to a problem validating the system timer. Some Linux distribution versions might enable this option automatically when they detect they are running on Hyper-V.

To add `cloud-init` to your image, set up a YUM repository that holds the `cloud-init` RPMs and install them by referring to documentation of your Linux distribution documentation. For example, for Red Hat, run the following commands:

```
yum install cloud-init
yum install cloud-utils
yum install dracut-modules-growroot
```

Note: If RHEL 7.1 is being used, you must run:

```
yum install cloud-utils-growpart.x86_64
```

instead of

```
yum install dracut-modules-growroot
```

After installing `cloud-init`, enable `root` login and password authentication by modifying the following parameters in the `/etc/cloud/cloud.cfg` configuration file:

```
disable_root: 0
ssh_pwauth: 1
```

For more information, see OpenStack Linux image requirements.

Note: After installing the `dracut-modules-growroot` package on Red Hat, to automatically resize the root file system when the template is provisioned, run the **`mkinitrd`** command (with option `--force`, if required) to rebuild the `initramfs` used at boot time.

For information about installing `heat-cfntools`, see https://wiki.openstack.org/wiki/Heat/ApplicationDeployment.

For information on how to build AIX images or images for Linux on Power, see http://www.ibm.com/support/knowledgecenter/SSXK2N_1.2.1/com.ibm.powervc.standard.help.doc/powervc_images_hmc.html.

For information about installing `cloud-init` for Linux on Power, see http://www-01.ibm.com/support/knowledgecenter/SSXK2N_1.2.1/com.ibm.powervc.standard.help.doc/powervc_install_cloudinit_hmc.html.

`cloud-init` is not supported on AIX. You can perform basic deployments only.

heat-cfntools for Linux on Power can be downloaded from http://dl.fedoraproject.org/pub/epel/6Server/ppc64/ for PowerVC images.

For information about creating base images for Linux on System z, see "Creating base images for Linux on System z" on page 243.

For information about how to configure cloud-init for any Linux operating system, including Red Hat Enterprise Linux or SUSE Linux, see the documentation for your Linux distribution.

Creating base images for Linux on System z

Add cloud-init to images for Linux on System z.

For information about how to build z/VM images, see the chapter 6 of the *Enabling z/VM for OpenStack (Support for OpenStack Kilo Release)* guide at http://www.vm.ibm.com/sysman/openstk.html.

Note: Reversal of the order of the services startup for z/VM: sshd must be run before cloud-init.

The service startup order is determined by /etc/rc.d/rc[0-6].d/ directory which contains symlinks to /etc/init.d/ files. To change startup order, in particular runlevel, you must change the name of files in the appropriate directory. Scripts are run by name order so that script with lower number behind letter S starts earlier. In z/VM, sshd needs to start before cloud-init so it must have a lower number. Changes must be done for at least the default runlevel, which in RHEL is 3.

To update the RHEL image to make sure that cloud-init-local starts after sshd service is started, perform the following steps:

1. In /etc/init.d/cloud-init-local file, add sshd in the Required-Start statement:

```
# Required-Start:    $local_fs $remote_fs xcatconf4z sshd
```

2. From the command line, run following commands:

```
chkconfig cloud-init-local off
chkconfig cloud-init-local on
```

heat-cfntools is not supported on Linux on System z.

Adding images to your OpenStack environment

You can add images to your OpenStack environment to be used by IBM Cloud Orchestrator.

About this task

To use an image in IBM Cloud Orchestrator, you must add the image to your OpenStack environment.

If you are using Linux on System z, follow the instructions in *Enabling z/VM for OpenStack (Support for OpenStack Kilo, Mitaka or Ocata releases)* guide at http://www.vm.ibm.com/sysman/openstk.html.

If you are using VMware, you can populate the Glance repository automatically by using the discovery process (see "Configuring vmware-discovery" on page 152) or you can add images to the Glance repository manually. If you rely on VMware discovery, you can skip the remaining part of this topic.

If you are using PowerVC, images are displayed in Glance automatically, without any additional action.

To add an image to OpenStack, complete the following steps on the OpenStack Controller:

Procedure

1. Set the environment by running the following command:

```
source /root/openrc
```

2. Run the following command on one line:

```
glance image-create
--name image_name
--disk-format disk_format
--container-format container_format
--is-public [True|False]
< image_path
```

where

image_name

Specifies a name for the new image that you are adding.

disk_format

Specifies one of the following disk formats:

raw

An unstructured disk image format.

qcow2

A disk format that is supported by the QEMU emulator that can expand dynamically and supports copy-on-write.

vmdk

For a VMware hypervisor, another common disk format that is supported by many common virtual machine monitors.

container-format

Specifies the container format for the image. The acceptable formats are: `aki`, `ami`, `ari`, `bare`, and `ovf`.

--is-public

Specifies whether the image is accessible by other users. The value can be `true` or `false`. If you specify the `false` value, the image is accessible only in the scope of the project specified in the `/root/openrc` file.

image_path

Specifies the full path of the image to be added.

For more information about the **glance image-create** command, see the OpenStack documentation.

Attention: If you are deploying on VMware, specify the additional properties **vmware_adaptertype**, **vmware_ostype**, and **vmware_disktype**.

For example:

```
glance image-create
--name my_vmware_windows_image
--disk-format vmdk
--container-format bare
--is-public False
--property vmware_disktype="preallocated"
--property vmware_adaptertype="lsiLogic"
--property vmware_ostype="Windows764_Guest"
< /tmp/images_to_create
```

where **vmware_disktype** can be `sparse`, `preallocated`, or `streamOptimized`, and the **vmware_adaptertype** can be `ide`, `busLogic`, or `lsiLogic`. VMDK disks that are converted by the **qemu-img** utility are always monolithic sparse VMDK disks with an IDE adapter type. If the image does not come from the **qemu-img** utility, **vmware_disktype** and **vmware_adaptertype** might be different.

To determine the image disk type and adapter type from an image file, use the `head -20` *`vmdk_filename`* command, and find the `createType` and `ddb.adapterType` in the command output. Currently, the operating system boots VMDK disks with an IDE adapter type that cannot be attached to a virtual SCSI controller. Disks with one of the SCSI adapter types (such as `busLogic` or `lsiLogic`) cannot be attached to the IDE controller. Therefore, as the previous examples show, it is important to set the **vmware_adaptertype** property correctly.

The default adapter type is lsiLogic, which is SCSI. You can omit the **vmware_adaptertype** property only if the image adapter type is lsiLogic.

When you create an image by using the **glance image-create** command or the OpenStack Dashboard, and the image format is not of raw type, you must specify the minimum disk space that is required for the image to run (that is, the current disk size). The disk size is specified in GB, and the value must be greater than or equal to the virtual size of the image. You can use the following command to find the virtual size of an image:

```
# qemu-img info
```

Tip: If using the **glance image-create** command, specify the minimum disk size by using the --min-disk *value* option. If using the OpenStack Dashboard, specify the required value in the **Minimum Disk (GB)** field.

Note: Windows has a different mechanism of interpreting the hardware clock than Linux does. Therefore, the following settings are recommended for a Windows guest image:

- Set the time zone of the image the same as the compute node.
- Disable time synchronization with Internet inside the image, so that the guest virtual machine gets its time solely from the hypervisor.
- Set the os_type=windows metadata with the --property option when registering the image.

Note: When you use the OpenStack Dashboard to create an image and your image size is large (>1 GB for example), it is recommended to use Image Location as Image Source. If you select Image File as Image Source for a large image size, your image might not be loaded correctly because the Horizon server might timeout and the image may remain in Saving status.

Note: To deploy a Hyper-V generation 2 virtual machine, when creating the image in the OpenStack environment, you must specify the parameter hw_machine_type=hyperv_gen2.

For more information about the **glance** command, see the OpenStack documentation.

Chapter 10. Migrating to IBM Cloud Automation Manager

Learn how to make your IBM Cloud Orchestrator to coexist with IBM Cloud Automation Manager.

For complete information about this coexistence, see https://www.ibm.com/support/knowledgecenter/SS2L37_3.1.2.0/cam_integrate_ICO.html.

Chapter 11. Managing a hybrid cloud

A hybrid cloud is a cloud computing environment in which an organization provides and manages some resources in-house and has others that are provided externally. For example, an organization might use a public cloud service, such as Amazon, for archived data but continue to maintain in-house storage for operational customer data.

IBM Cloud Orchestrator provides a component that is called Public Cloud Gateway to integrate with public clouds.

Examples of public clouds that are managed with the Public Cloud Gateway are:

- Amazon AWS EC2
- IBM SoftLayer

Additionally, IBM Cloud Orchestrator contains support to on-board, create, and manage Cloud Services and Cloud Services Deployments on Microsoft Azure.

Using the Public Cloud Gateway

Use the Public Cloud Gateway to integrate with public clouds as Amazon EC2 or IBM SoftLayer.

Public Cloud Gateway overview

The Public Cloud Gateway is a web application that provides an OpenStack API compatibility layer so that the Amazon Elastic Compute Cloud (Amazon EC2) and IBM SoftLayer work like the standard OpenStack, Cinder, and Glance instances.

The Public Cloud Gateway is automatically installed as part of the IBM Cloud Orchestrator installation process. You can deploy and manage virtual machines and storage across both private and public clouds, for example, Amazon EC2. Public Cloud Gateway calls the management APIs of the Remote Cloud to perform the OpenStack API.

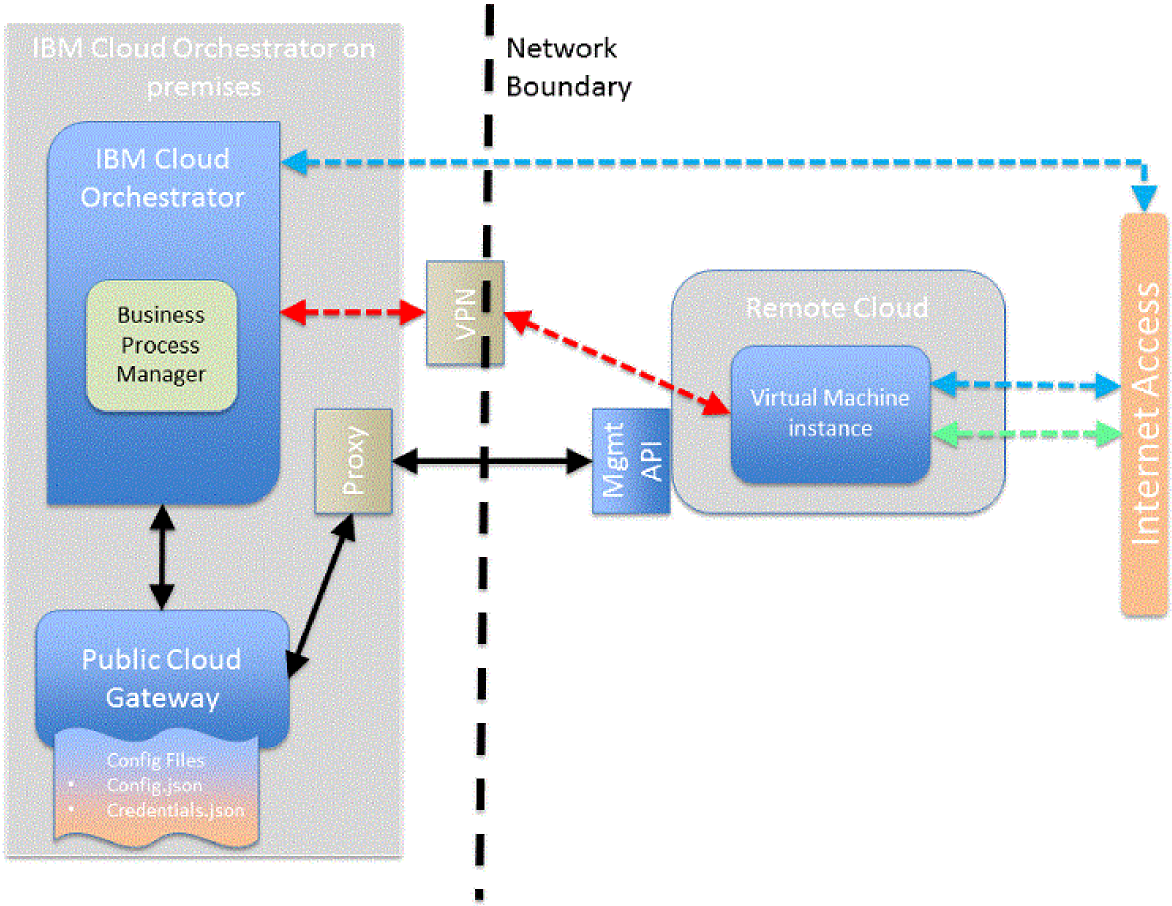


Figure 4. Overview

The Public Cloud Gateway provides a subset of the available OpenStack API. For more information, see "OpenStack API support" on page 252.

The invocation of the management APIs of the Remote Clouds might go through an HTTP/HTTPS proxy. For information about the configuration of a proxy, see "Remote Cloud API proxy configuration" on page 271.

Access to the provisioned virtual machine instance is either through a VPN connection the on-premises IBM Cloud Orchestrator instance to the management network of the provisioned virtual machine instances or through the internet.

The Remote Cloud capabilities together with the Public Cloud Gateway configuration define which connectivity to the provisioned virtual machine instance is used.

The configuration of the VPN between the IBM Cloud Orchestrator on-premises instance and the remote clouds must be done out of band depending on the available or chosen hardware devices available on the on-premises site. The Business Process Manager node requires access to the remote cloud through VPN.

Configure the Public Cloud Gateway by following the procedure described in "Configuring the Public Cloud Gateway" on page 253.

Review the list of capabilities and limitations for the Public Cloud Gateway.

Capabilities and limitations

This topic describes capabilities and limitations for IBM Cloud Orchestrator in managing hybrid clouds together with the Public Cloud Gateway using OpenStack.

All IBM Cloud Orchestrator capabilities are supported on managing hybrids clouds with the following limitations:

- Heat stacks management is not supported.
- Administering as cloud administrator (Project and Admin section) is not supported.
- Unable to import images from sources outside of public cloud repositories.
- Supports only single predefined Dynamic Host Configuration Protocol (DHCP) network.
- OpenStack command line clients are not supported for Public Cloud Gateway.
- OpenStack API support. The Public Cloud Gateway supports a limited set of OpenStack APIs.
- OpenStack Cinder Storage Volumes toolkit: when attaching a volume, the **Format Volume** and **Mount Volume** options are not supported on Windows operating systems. These options are only supported on RHEL operating system by running the **Attach volume** action as `root`.

Amazon AWS EC2

The following topic describes capabilities and limitations for Amazon AWS EC2 with the Public Cloud Gateway.

Public Cloud Gateway capabilities with Amazon AWS EC2

- **Amazon VPC support**

  The Public Cloud Gateway supports the placement of machines in different subnets and security groups when a non-default VPC is configured for a given account and region.

  This capability is available when the only supported platform for your account is **VPC**. It is not enabled when other supported platforms are listed, for example, EC2. You can check the supported platforms of your account on the EC2 dashboard in the **Account Attributes** section. You must perform more configuration tasks to use Amazon VPC support. See "Configuring the Public Cloud Gateway for Amazon EC2" on page 272 and "Configuring subnets and security groups in a non-default VPC region" on page 276.
- **Amazon API support**

  The Public Cloud Gateway supports regions with Version 4 Security like, for example, Europe-Central Frankfurt.

Public Cloud Gateway limitations with Amazon AWS EC2

All IBM Cloud Orchestrator capabilities are supported by the following limitations for Amazon AWS EC2:

- Heat stacks management is not supported.
- The OpenStack Dashboard is not supported.
- Windows images are automatically activated and license keys are provided by the remote cloud.
- Support Single NIC only.
- Resize of a virtual machine instance supports only CPU and Memory and is only possible in the shutdown state.
- Depending on the network configuration, the start/stop of the virtual machine instance assigns a new public IP address.
- You cannot detach a mounted volume. Make sure to unmount the volume before detaching it.
- Linux Amazon Machine Images that use HVM virtualization type are not supported.
- To attach a pre-partitioned and formatted EC2 volume, you need to mount it manually.

IBM SoftLayer

The following topic describes capabilities and limitations for IBM SoftLayer with the Public Cloud Gateway.

Public Cloud Gateway capabilities with IBM SoftLayer

- Supporting single or dual NIC depending on region level configuration.

Public Cloud Gateway limitations with IBM SoftLayer

All IBM Cloud Orchestrator functions are supported by the following limitations for IBM SoftLayer:

- Heat stacks management is not supported.
- The OpenStack Dashboard is not supported.
- Windows images in IBM SoftLayer are automatically activated and license keys are provided by the remote cloud.
- Resize of a virtual machine instance does only support CPU and RAM.
- Volume Provider Limitations:
  - Name is supported and description is not supported. If no name is supplied, a name is created in the following form:

    ```
    HybridStorage-<volume_size>GB-<date>
    ```

  - If no mount point information is returned by IBM SoftLayer, `/mnt` is shown as mount point in IBM Cloud Orchestrator.
  - The size of the volume is defined through the size options available for SoftLayer Portable Storage. Consider that the SoftLayer Portable Storage is attached as a local storage, which means that only a subset of the size options is available for use. Storage is only attached in the granularity available for Portable Storage, for example, if you request 1 GB, you get 10 GB.
- To attach a pre-partitioned and formatted SL volume, do the following steps:
  1. Attach the volume by using the **Manage volume** action without the **Partition volume** option.
  2. Manually mount the volume.

OpenStack API support

The Public Cloud Gateway supports a limited set of OpenStack APIs.

Note: Because the Kilo release OpenStack does not support the XML version of the OpenStack API, you must not use the XML version of the OpenStack API implemented by Public Cloud Gateway anymore.

Supported OpenStack APIs include:

- OpenStack Nova
- OpenStack Glance
- OpenStack Cinder

Supported OpenStack Nova API

- Create (boot) a virtual machine with:
  - SSH key
  - Availability zone
  - Single DHCP network
- List virtual machine instance:
  - Filter by status
  - Filter by SSH key name
- Delete virtual machine
- Start / stop virtual machine
- Show virtual machine detail
- List images
- Show image detail
- List Availability Zones
- List Networks providing a single DHCP network

- List Extensions
- List Flavors
- Show Flavor details
- Get version info
- Get Limits
- Show Network providing a single DHCP network
- Query quota for tenant
- Query quota defaults for tenant
- Set quota for tenant
- Delete quota of tenant
- Attach Volume
- Detach Volume
- Create key pair providing the SSH key import within the request
- List key pairs
- Delete key pair

Supported OpenStack Glance API

- List images
- Show image detail

Supported OpenStack Cinder API

- Create volume
- List volumes
  - Filter by status
- Filter by status
- Show volume detail
- Delete volume
- List volume types
  - Single hardcoded entry

Note: All other OpenStack documented APIs are not supported by the Public Cloud Gateway.

Configuring the Public Cloud Gateway

The Public Cloud Gateway is deployed as part of the IBM Cloud Orchestrator installation. However, the Public Cloud Gateway is not enabled by default and certain updates to the configuration files are required before you can use the Public Cloud Gateway.

Before you begin

Ensure that the prerequisites are satisfied.

About this task

To configure the Public Cloud Gateway, set up the following configuration files:

- `admin.json`
- `config.json`
- `credentials.json`
- `flavors.json`

By default, these files are located in the `/opt/ibm/ico/wlp/usr/servers/pcg/etc` directory.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

When the Public Cloud Gateway is installed, default values for all parameters are provided except *cloud access* keys. Only administrators must change these settings in the configuration files that affect their particular setup.

The configuration of the Public Cloud Gateway is slightly different depending on the actual remote cloud:

- "Configuring the Public Cloud Gateway for Amazon EC2" on page 272
- "Configuring the Public Cloud Gateway for SoftLayer" on page 277

There is a set of "Common configuration tasks" on page 260 common across all supported remote clouds.

Related reference

Unable to connect to a public cloud due to missing credentials
In the Public Cloud Gateway, you might receive the error `Unable to connect to public cloud due to missing credentials`.

SSH key management

The Public Cloud Gateway provides the capabilities for SSH key management, that is, OpenStack key pairs.

For Amazon EC2 and SoftLayer, you can support register and unregister SSH key offerings in the catalog.

For information about SSH keys, see the description of the register and unregister key offerings:

- "Registering a key pair" on page 224.
- "Unregistering a key pair" on page 224.

Assumptions:

- Key pairs are subject to multitenancy. Key pairs are scoped on a per project basis.
- If a key pair is generated during execution of the `Register a key pair` offering, it is not immediately deployed into a Public Cloud Gateway-managed region. The SSH key is deployed when the first virtual machine deployment with the registered SSH key is run.

Limitations:

- IBM SoftLayer does not allow storing the same SSH key (with the same fingerprint) multiple times under different key pair names. If an SSH key is registered multiple times with the `Register a keypair` offering, the deployment fails when the SSH key is deployed the second time during the `deploy single server` offering with a different name but with the same fingerprint.
  - Remove the second registered SSH key with the same fingerprint by using the `Unregister a key pair` offering. Use every time that you register an SSH key a newly created one.
- IBM SoftLayer and Amazon EC2 SSH keys have an account scope. The Public Cloud Gateway postfixes the name of the SSH key with a `<tenantuuid>` when the SSH key is deployed on the remote cloud. On the Keypair list/show or instance list/show the `<tenantuuid>` is removed.

  Note:

  - IBM SoftLayer SSH keys use the fingerprint of the SSH key as a unique key. Therefore, a specific SSH key can only be registered once into an IBM SoftLayer account independent of its given name.
  - If multiple regions are mapped through the Public Cloud Gateway to a single IBM SoftLayer account, you must set the `keypairTimeout` and the `keypairQuotaTimeout` in `config.json` to 0. If you do not do this, there are deployment errors. This is because SSH keys in IBM SoftLayer have account scope and the caches are on region level. The 0 value for the two cache properties disables the caches.

Multitenancy support

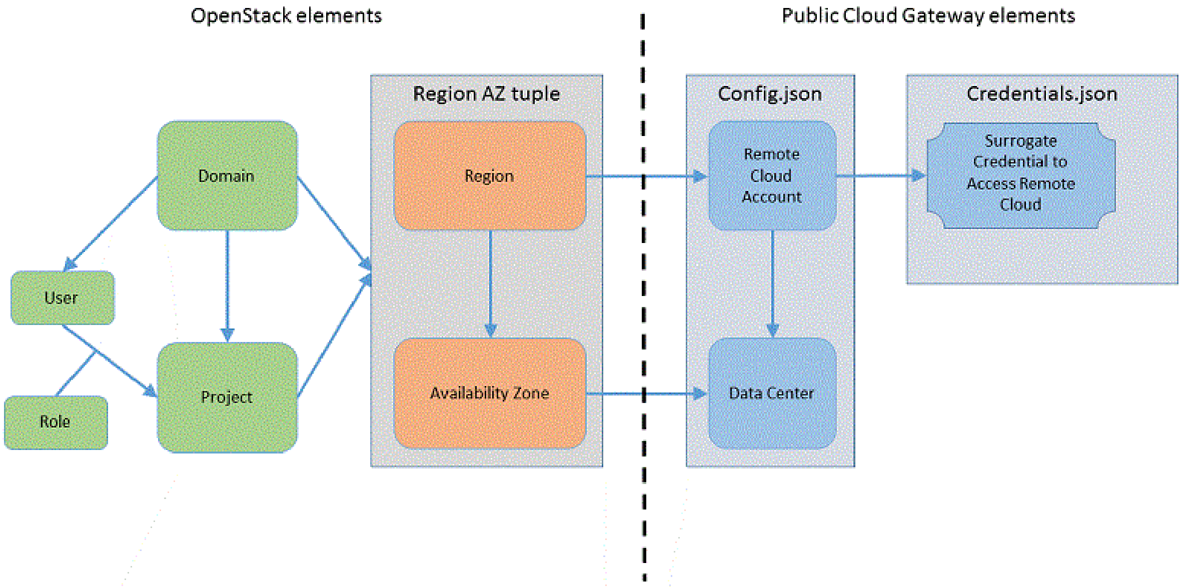
The Public Cloud Gateway provides capabilities for multitenancy.

These capabilities are in addition to the general multitenancy capabilities in the core product.

The Public Cloud Gateway contains the following capabilities that are related to multitenancy:

- Supports non-default domains and projects.
- Limits the view of resources on project scope.
- Creates resources in scope of a project.
- Supports quotas on a per project and per region base.

Multitenancy concepts that are provided by the Public Cloud Gateway:



Public Cloud Gateway has two major configuration files:

- `config.json`
- `credential.json`

`config.json` defines the region and availability zone that might be attached to an OpenStack project. The `credentials.json` defines the surrogate user ID that is used during provisioning for a given region and project combination. The regions that are defined in the `config.json` must be registered in Keystone. After this registration is done, you must map the new region and availability zone tuples to domain and project.

The sequence is:

1. Register the new region and availability zone on the domain by using the OpenStack Dashboard that uses the identity section.
2. Register the new region and availability zone on the project with your domain by using the OpenStack Dashboard that uses the identity section.

Note: If you miss the first step for the domain of your project, you do not see the new region and availability zone available for your project.

Amazon EC2 and IBM SoftLayer support that a remote cloud account might be shared by multiple project. Public Cloud Gateway takes care of segregating the resources in the remote cloud for the different project sharing the same remote cloud account.

All provisioning actions that are done in the remote cloud are run under the surrogate user credentials that are registered for the specified region or project. Public Cloud Gateway adds the OpenStack project ID and user ID to the resources that are created in the remote cloud.

Note: For Amazon EC2 the `projectUUID` and `userUUID` of the requester are added to the remote cloud resources. For IBM SoftLayer only the `projectUUID` of the requester is added to the remote cloud resources.

Mapping to OpenStack concepts:

The Public Cloud Gateway supports the following OpenStack constructs and resources in a multitenancy model:

- Virtual machine instances
- Storage volumes
- SSH keys

The following resources are public within the remote cloud account level:

- Images
- Networks

Scoping assumptions within the Public Cloud Gateway:

- A region must map to a single remote cloud account.
- The project to which the cloud administrator belongs to, must be mapped for each region to a credential that can see all resources within a remote cloud account. The default project name within IBM Cloud Orchestrator is `admin`, where the cloud administrator belongs to.

Mapping capabilities within the Public Cloud Gateway:

- The `config.json` defines the regions that are exposed by the Public Cloud Gateway. Each region maps to a single remote cloud account and a data center within the account.
- Accounts in a remote cloud and data center map to Region and Availability Zone in OpenStack.
- The OpenStack Dashboard identity section maps Region and Availability Zone in OpenStack to a domain and then to a project within the domain.
- The `credential.json` maps projects to credentials within the remote cloud account. There are two options:
  - Map credentials to projects globally.
  - Map credentials to projects within the scope of a region.

  Note: If you have multiple regions per remote cloud type (Amazon EC2 or SoftLayer) it is required to do the mapping in scope of a region, because it is not possible to share a user ID in the remote cloud among accounts.

Share an Amazon AWS EC2 or SoftLayer account across projects by using a single set of credentials.

When a resource is created in the remote cloud, it is created within a project scope. This means that you can only see the resource that belongs to the project you are logged in to. This feature is not apparent in the Public Cloud Gateway.

Note: Resources are either tagged by project ID or separated by namespace. If you log on to the remote cloud management console, you can see the tags and namespaces for the various resources. In the `credentials.json` file, there is a new construct to provide a mapping of all projects within a region to a specific credential. An example statement for Amazon AWS EC2 is:

```
{
            "tenantName":"*",
            "region": "yyy",
            "access_key_ID":"xxx",
            "secret_access_key":"xxx"
          }
```

An example statement for SoftLayer is:

```
{
            "tenantName":"*",
            "region": "yyy",
            "user_id":"xxx",
            "api_access_key":"xxx"
          }
```

Share a cloud account across projects by using dedicated credentials per project:

The Public Cloud Gateway contains a configuration file that maps the credentials that are used for a region to a project. This feature allows you to supply different logon credentials on a per project and per region basis.

Limitations:

- The multitenancy support does not provide a physical segregation of resources because they still belong to the same account. It provides different views of the account on a per project basis.
- The network is shared across an account.
- Storage comes from a shared pool.
- Images are public.

Quota support overview

The Public Cloud Gateway provides capabilities for quota management.

The following are the types of quota definitions in Public Cloud Gateway:

- A single global default quota that is used if no project level definitions are set.
- Per region and project quota set managed by the administration view in the project detail section.

The following quotas are supported in Public Cloud Gateway. This is only a subset of the OpenStack defined set:

instances

Total number of instances that can be provisioned.

cores

Total number of cores that can be consumed.

ram

Total RAM size in MB that can be consumed. Must be larger than the **gigabytes** quota value.

gigabytes

Largest size of a single volume in gigabytes.

volumes

Number of volumes that can be created.

key_pairs

Number of key pairs that can be created.

The following actions can be done on quotas:

- Set global quota defaults. Maintained within the Public Cloud Gateway configuration.
- Query and Set per project quota defaults. Maintained within the administration view in the project details.
- Delete a project quota. Supported only through OpenStack API calls. Not supported through the IBM Cloud Orchestrator UI.

Assumptions:

- The Administrator of the Public Cloud Gateway region must ensure that the sum of the per project quota does not exceed the region capability.
- Quotas are enforced on a per project and region base.
- Quota calculation is based on the flavor values for virtual machine instances, the number of additional disks which are connected and the sshkey usage. For more information, see "Configuring flavors" on page 265.

- Quota calculation is done at certain intervals configurable in `config.json`. This means that there might be time frames were the actual situation in the remote cloud is different than reported through the quota management. For more information, see "Configuring caching" on page 267.
- Each remote cloud has some specific assumptions in flavor support and management that might impact the accuracy of the quota calculation. For more information, see "Configuring flavors" on page 265.
- The IBM Cloud Orchestrator Dashboard view uses the data from the quota management system within the Public Cloud Gateway.

Network planning
Public Cloud Gateway scenarios require a set of network configuration to successfully provision resources within the remote cloud. This topic provides an overview about which network configuration is assumed and required.

- Access to the remote cloud REST API entry points.
- Communication from the IBM Cloud Orchestrator management stack to and from the provisioned virtual machine instances.

Access to the remote cloud REST API entry points

During the management lifecycle, the Public Cloud Gateway requires access to the remote cloud REST API entry points for:

- Amazon AWS EC2 REST API.
- SoftLayer REST API.

Public Cloud Gateway provides two scenarios for accessing the remote cloud API REST entry points:

- Direct connection from the IBM Cloud Orchestrator Server where the Public Cloud Gateway connects to the remote entry point.
- Indirect connection through a customer provided proxy server. For more information, see "Remote Cloud API proxy configuration" on page 271.

Connectivity from IBM Cloud Orchestrator management stack to and from provisioned virtual machines

In addition to the remote cloud REST API entry points, communication several management actions require access to the provisioned virtual machine instances. Examples are:

- Running of scripts
- Actions from the Instance View

In the Public Cloud Gateway scenarios, it is assumed that the connectivity from and to the IBM Cloud Orchestrator management from the virtual machine instance in the remote clouds is provided by the customer. This is an example list of what might be used to establish the communication requirements:

- Open VPN
- Amazon EC2 VPN gateway
- Vyatta / Fortigate Security Appliance (FSA) in SoftLayer

The following network protocols are used:

- Protocols that are used by any of the agents running on the provisioned virtual machine to their infrastructure servers. For example, Tivoli Monitoring.
- For Windows provisioning, RDP port (3389) must be enabled in the security groups on Amazon EC2, when you create the image, and it must be enabled in the `Default` security group for deployment.

Note: The network requirements must be in place and working before the first provisioning in the remote cloud by using the Public Cloud Gateway.

Amazon AWS EC2 related network planning

This topic covers Amazon AWS EC2 specific network planning topics like:

- Supported capabilities
- Assumptions and limitations
- VPN

Introduction:

Amazon AWS EC2 provides three different network models depending on when the account was created. An account supports either `EC2Classic` and `non-Default VPC` or `Default` and `non-Default VPC`.

To use all of the capabilities the Public Cloud Gateway supports, it is required to have an actual account that supports `Default` and `non-Default VPC`. All accounts that are created after December 2013 are fine. You can check the account capabilities by using the Amazon AWS EC2 console under **Account Attributes** on the main EC2 Dashboard. If you see `Supported Platforms VPC` and `Default VPC` then your account supports all Public Cloud Gateway capabilities.

Supported capabilities:

Amazon AWS EC2 distinguishes between three types of networking models:

- EC2Classic
- Default-VPC
- non-Default VPC

Each of the three networking models has specific capabilities and limitations. Refer to the Amazon Documentation in the EC2 and VPC user guide to understand the main capabilities and limitations. Public Cloud Gateway supports all the three network models. The following table provides a quick overview about the supported capabilities.

Table 17. Supported capabilities of the networking models

Capability	**EC2Classic**	**Default-VPC**	**Non-Default VPC**
Private IP address	Yes	Yes	Yes
Public IP address	Yes	Configurable per region	Configurable per region/ project
Hide Public IP address	Yes, it is required for VPN setup	No, done by setting `privateOnly` to `true`	No, done by setting `privateOnly` to `true`
Set Private Subnet	No, EC2Classic defines subnet	No, Default VPC defines subnet	Yes, Granularity on project or region
Set Security Group	No, only Default security for EC2Classic is used	No, Default security of default VPC is used	Yes, Single Security group on region per project granularity
Elastic IP address	No	No	No
Config	On Amazon account level	On Amazon account level	"vpc" property in `config.json` on region definition and VPC definition in the Amazon Management portal

Assumptions and limitations:

- Each virtual machine that is provisioned by Public Cloud Gateway gets a private IP address. The available networking model of the Amazon EC2 Account defines from which subnet the private IP address is derived.

- Public IP addresses are derived from a global subnet within Amazon AWS EC2. These IP addresses are only assigned to a virtual machine as long as the virtual machine is running. A Stop/Start sequence assigns a new public IP address to the virtual machine. Public IP addresses are realized through NAT and are not reachable through a VPN. Public IP addresses can only be reached through an internet connectivity.
- Elastic IP addresses are not supported.
- Only a single security group can be assigned to a virtual machine controlled by a Public Cloud Gateway.
- For non-Default VPC, only a single subnet or security group must be tagged with a `projectUUID` or "`*`" per availability zone. If either multiple or none of the subnets are tagged, provisioning fails. If no security group is tagged, the default security group of the VPC is used.

VPN setup:

Virtual machines can be accessed through a VPN (virtual private network) on their private IP address. Setting up a VPN is out of scope of this documentation. There are various options to set up a VPN within Amazon AWS EC2. For example, VPN support in VPC, using an `OpenVPN` or `IPSEC` gateway. It is required to set up the VPN to the private IP address as the public IP addresses are only reachable through an Internet connectivity.

Common configuration tasks

There are some configuration tasks that you must perform.

Prerequisites

Before configuring the Public Cloud Gateway, ensure that the following requirements are satisfied.

General requirements

Depending on which public cloud you are using, there are certain requirements necessary on the cloud that is integrated.

You must have an Amazon Web Service (AWS) account with Amazon EC2 credentials for each tenant or project by using the Public Cloud Gateway. For more information, see the AWS Management Console at https://console.aws.amazon.com/console/home.

Set up an account in SoftLayer and create one or more user IDs. Each ID has its own unique password and API access key. The API access key is required to configure SoftLayer integration in the Public Cloud Gateway.

Network requirements

- **Port requirements** – The Public Cloud Gateway requires access to a number of ports in the installation environment and in the default Amazon EC2 security group. If these ports are blocked by a firewall or used by another process, some Public Cloud Gateway functions do not work.

Table 18. Ports used by Public Cloud Gateway

Port	TCP or UDP	Direction	Description
22	TCP	Outbound	SSH communication with the virtual machine instances.
ICMP			ICMP communication with the virtual machine instances.
443	TCP	Outbound	HTTPS communication with: – Amazon EC2 management endpoints.

Note: Make sure that the Amazon EC2 security groups are configured according to the table. For more information about Amazon EC2 security groups, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html

- **DNS requirements –** Ensure that the DNS is configured correctly. It must be able to resolve the Amazon EC2 management endpoints as defined in the `/opt/ibm/ico/wlp/usr/servers/pcg` file.

  Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

Access to public cloud resources
To provision virtual machines or use any services in the public cloud, users are required to have credentials to access public cloud resources. These credentials are then used in configuration of the Public Cloud Gateway.

Images in public cloud
To deploy images in the public cloud, users are required to provide image templates in the public cloud image repositories. See "Creating a supported image" on page 261.

Creating a supported image

You can create an image to be deployed in hybrid clouds using the Public Cloud Gateway.

Creating Linux operating system images

You can create Linux operating system images to be deployed in hybrid clouds.

The image creation depends on the target hybrid cloud.

Amazon AWS EC2

- Many of the current Amazon AWS EC2 images are `cloud-init` enabled.
- If you are using an image that is not `cloud-init` enabled, follow the steps to add `cloud-init` support here: "Creating Linux base images" on page 241
- You must make a private copy out of the existing Amazon EC2 images.
  - Only private images can be used together with Public Cloud Gateway.
  - To create a Linux Amazon Machine Image, follow the description in the Amazon EC2 documentation here: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html.

    Note: Linux Amazon Machine Images that use HVM virtualization type are not supported. Specify paravirtual (PV) as virtualization type.

    Note: The root disk size of the provisioned virtual machine depends on the block devices setting of the image. For example, Block Devices:

    ```
    /dev/sda1=snap-1d9beb9c:10:true:gp2
    ```

    The value of 10 in the example defines the size of the root disk during provisioning. The value is in GB. The root disk size adds to the cost of a virtual machine. The size is defined when the image is created from a provisioned virtual machine.
- To enable password base authentication, see "Password authentication on Amazon EC2 images" on page 283

IBM SoftLayer
To create a `cloud-init`-ready image template in SoftLayer, complete the following steps:

1. From the SoftLayer portal, create an instance by using SoftLayer-provided base OS images or create an instance from an existing image template that needs a `cloud-init` script installed.
2. To add `cloud-init` support to the image, follow the procedure in "Creating Linux base images" on page 241.

Note: If you do not correctly modify the `cloud-init` configuration file as specified in the procedure, you cannot access the provisioned virtual machine anymore.

3. Download the file `http://<ico_server>:9797/downloads/scripts/softlayer/linux/cloud-init/DataSourceSL.py`, where `<ico_server>` is the IP address of the machine where Public Cloud Gateway is installed, and store the file in the `/usr/lib/python2.6/site-packages/cloudinit/sources` directory.
4. Update the `settings.py` script in the `/usr/lib/python2.6/site-packages/cloudinit` directory. Add SL in the data source list and comment out all the others.
5. When the instance is running, from the SoftLayer portal, access the computing instance: **Device** > **Device List** > **Device Name**.
6. From the Device List, select the Computing Instance from actions menu. Select **Create Image Template**.
7. Follow the prompts to create the image template.

Creating Windows operating systems images

You can create Windows operating system images to be deployed in hybrid clouds.

The image creation depends on the target hybrid cloud.

IBM SoftLayer

To create a `cloudbase-init` image template in SoftLayer, complete the following steps:

1. Deploy single virtual server from a public image:
   a. Select a public image, for example Windows Server 2012 Standard Edition (64 bit).
   b. Flavor: Small.
   c. No key.
   d. No user/password.
   e. No volume attaches.
2. Log on to the virtual machine:
   a. The virtual machine password is generated at first. So once the virtual machine is reported as `ACTIVE`, open the SoftLayer portal and navigate to your devices list.
   b. Expand the virtual machine that you provisioned and click the **show password** box to reveal the administrator password.
   c. Use this password to log on to the virtual machine by using RDP with the IP address provided.
   d. Depending on the load on the SoftLayer data center, you are using, it might take up to 20 minutes before the login information becomes available.
3. Install `cloudbase-init` on the virtual machine:
   a. Download the installer from https://www.cloudbase.it/downloads/CloudbaseInitSetup_Beta_x64.msi.
   b. Run the installer.
   c. Enter the correct administrator user name for your version of Windows. For example, `Administrator` for the English version.
   d. Make sure that you check the **use metadata password** option.
   e. Click **next** and wait until the installation completes. Do NOT select to run `sysprep` or to shut down the virtual machine.
   f. Click **finish** to close the installer.
4. Copy the SoftLayer metadata service to the virtual machine:
   a. The built-in version of `cloudbase-init` does not support loading metadata from SoftLayer. Therefore, the `cloudbase-init` installation on your virtual machine must be extended with a small file that implements a SoftLayer metadata service.

Download from http://<ico_server>:9797/downloads/scripts/softlayer/windows/cloudbase-init/.

b. Copy slservice.py to the services folder of your cloudbase-init installation. The default is C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\Python27\Lib\site-packages\cloudbaseinit\metadata\services.

c. Adjust the configuration settings of cloudbase-init:

   1) Open the file cloudbase-init.conf in an editor. The default is C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf.

   2) Make sure that it contains these lines:

```
metadata_services=cloudbaseinit.metadata.services.slservice.SLService
plugins=cloudbaseinit.plugins.windows.setuserpassword.SetUserPasswordPlugin
```

d. Create a private image from the virtual machine:

   1) Leave the virtual machine (do not shut it down, run sysprep or anything).

   2) Go back to the SoftLayer portal and click on the virtual machine to open its details.

   3) In the action menu, select the **create image template** action and provide an image name.

   4) You can now use this private image for provisioning with IBM Cloud Orchestrator.

e. (Optional) Delete the original virtual machine:

   1) Once the new image has been created, you can safely delete the original virtual machine by using the IBM Cloud Orchestrator UI.

   2) Be aware that creating the private image template might take up to 20 minutes depending on the size of the virtual machine.

   3) Until the image creation transaction has been completed, the original virtual machine cannot be deleted.

Note: Keep in mind that the password that you enter during provisioning is visible to anyone who can log on to the provisioned virtual machine. So it is best to change the password as soon as possible after the first login.

Note: Setting the password during provisioning works only if the chosen password complies with the password policy of the Windows operating system on the image. If the password you chose at provisioning time does not comply with the password policy, the password is not set. Then, you are able to access the virtual machine by using the password that is originally generated by SoftLayer that you can reveal by using the SoftLayer portal.

Amazon AWS EC2

1. Deploy a single virtual server from a public image:

   IBM Cloud Orchestrator does not display public images so you must deploy from the EC2 portal.

   Log in to the AWS portal, open the EC2 application and go to **AMIs**. Select the filter **Public Images**. Choose an available Windows **AMI**, for example Windows_Server-2012-R2_RTM-English-64Bit-Base. Click **Launch**. Select the instance flavor. An EBS-only flavor is recommended to save costs, for example t2.micro.

   Go to the next step to configure the instance. Depending on your requirements, you might also want to enable the public IP assignment. Click **Next** until you reach the security group configuration. There you must make sure that you select a security group that allows RDP access to the virtual machine. Click **Review and Launch** and then **Launch**. The initial administrator password is generated and encrypted using a Keypair. Be sure that you select a Keypair that you have access to. For this, you need the private key pem file from the Keypair creation.

2. Log on to the virtual machine:

   When the instance appears as running with all status checks done in the EC2 portal, select the instance and click **Connect**. Click **Get Password** and select the pem private key file of the Keypair you selected at provisioning time. Click **download remote desktop file** and open it with RDP. Use the displayed password to connect.

3. Install `cloudbase-init` on the virtual machine:

   Run the installer.

   Enter the correct administrator user name for your version of Windows. For example, `Administrator` for the English version. Make sure you that you do not check the `use metadata password` option as EC2 does not provide passwords by using metadata. Click **Next** and wait until the installation completes. Do not select to run `sysprep` or to shut down the virtual machine. Click **Finish** to close the installer.

4. Copy password script to the virtual machine:

   The built-in version of `cloudbase-init` does not support setting the administrator password for EC2 instances. Also, the `EC2ConfigService` provided on Amazon does not yet support execution of Python scripts. Therefore, the `cloudbase-init` installation on your virtual machine must be extended with a small file that runs a script to set the password. Download from `http://<ico_server>:9797/downloads/scripts/ec2/windows/cloudbase-init/`. Copy `setpasswd_ec2.py` to the `localscripts` folder of your `cloudbase-init` installation. The default is: `C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\LocalScripts`.

5. Adjust the configuration settings of `cloudbase-init`:

   Open the file `cloudbase-init.conf` in an editor. The default is: `C:\Program Files (x86)\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf`. Make sure that it contains this line:
   `plugins=cloudbaseinit.plugins.windows.localscripts.LocalScriptsPlugin`.

6. Adjust the configuration settings of `EC2ConfigService`:

   Run `Ec2ConfigServiceSettings`. The default is: `C:\Program Files\Amazon\Ec2ConfigService\Ec2ConfigServiceSettings.exe`. In the ggg**Image** tab make sure that the `random` option is set for the administrator password. This enables the option to access the virtual machine through a Keypair while `cloudbase-init` allows access through password.

7. Adjust the service dependencies of `cloudbase-init`:

   `Cloudbase-init` needs to wait until the `Ec2ConfigService` is finished before it can set the administrator password. Therefore, the service dependencies of `cloudbase-init` must be adjusted. As the administrator user, open a command shell. Run the command:

   ```
   sc config cloudbase-init depend=Winmgmt/Ec2Config
   ```

8. Create a private image from the virtual machine:

   Leave the virtual machine (do not shut it down, run `sysprep` or anything). Go back to the EC2 portal and click the virtual machine to open its details. In the action menu, select the **create image** action and provide an image name. You can now use this private image for provisioning with IBM Cloud Orchestrator.

   Note: The root disk size of the provisioned virtual machine depends on the block devices setting of the image. For example, Block Devices:

   ```
   /dev/sda1=snap-1d9beb9c:10:true:gp2
   ```

   The value of 10 in the example defines the size of the root disk during provisioning. The value is in GB. The root disk size adds to the cost of a virtual machine. The size is defined when the image is created from a provisioned virtual machine.

9. (Optional) Delete the original virtual machine:

   Once the new image has been created, you can safely delete the original virtual machine by using the IBM Cloud Orchestrator UI. Be aware that creating the private image template might take up to

20 minutes depending on the size of the virtual machine. Until the image creation transaction has been completed, the original virtual machine cannot be deleted.

Note: Keep in mind that the password that you enter during provisioning is visible to anyone who can log on to the provisioned virtual machine. So it is best advised to change the password as soon as possible after the first login.

Note: Setting the password during provisioning works only if the chosen password complies with the password policy of the Windows operating system on the image. If the password you chose at provisioning time does not comply with the password policy, the password is not be set. If you chose to use a Keypair for accessing the virtual machine in addition to a password, you are still be able to connect to the virtual machine by using your private key to decrypt the password on the AWS portal.

Configuring flavors

OpenStack API requires flavors for provisioning virtual machines. IBM Cloud Orchestrator must be able to return a flavor list for each region.

The Public Cloud Gateway stores the current list of known flavors in the `flavors.json` file in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

The following capabilities are supported:

- A global flavor list that is provided by the `default` section, if no remote cloud or region-specific information is provided.
- A remote cloud type-specific default flavor that is provided through the following sections:
  - `ec2_default`
  - `softlayer_default`

All the flavor definitions within the `flavors.json` file must be valid for the related remote cloud regions. All the definitions in these sections are snippets and provide configuration examples.

Note:

- Changes to the `flavors.json` file are only active after restarting the Public Cloud Gateway.
- Failure to the assumptions result in provisioning errors.
- A flavor must not be removed if virtual machines with this flavor exist.

Amazon AWS EC2

Amazon AWS EC2 supports only a predefined list of flavors that are published on their website. The Public Cloud Gateway supplies a current list in the `flavor.json` file under the `ec2_default` section. This hardcoded list can be extended or corrected based on the changes Amazon AWS EC2 provides. See http://aws.amazon.com/ec2/instance-types/.

The following rules apply for Amazon AWS EC2:

- You can modify the global list that is supplied in the `ec2_default` section. This affects all Amazon AWS EC2 regions except the ones that have a separate named section.
- You can add a new section for the specific Amazon AWS EC2 region with the name because this region is defined in the `config.json` file.

Note: With the Amazon AWS EC2 API, you cannot query or manage the supported flavors. Any changes to the list of flavors for Amazon AWS EC2 must match the published list on their website or the list of flavors that are shown in the Amazon AWS EC2 management UI. Otherwise, the virtual machine has provisioning errors.

Note: The size of the root disk is highly dependent on the image that is used for provisioning. You might see a difference in the size that is shown in the flavor and the actual size of the root disk in the remote cloud. For more information, see “Creating a supported image” on page 261.

IBM SoftLayer

SoftLayer does natively support flavors during deployment. The `flavors.json` defines the set of flavors that you can use during deployment by using IBM Cloud Orchestrator. SoftLayer supports only a certain list of possible values for CPU, RAM, and disk. These values can change over time. The possible values are visible if you try to create a cloud compute instance through the SoftLayer provided management UI. Only these values can be used for flavor definitions. If other values are used, you might receive deployment errors.

The following rules apply for IBM SoftLayer:

- You can modify the global list that is supplied in `softlayer_default` section. This affects all IBM SoftLayer regions except the ones that have a separate named section.
- You can add a new section for the specific IBM SoftLayer region with the name because this region is defined in the `config.json` file.

Note:

- IBM Cloud Orchestrator requires at least 512 MB of memory to be defined in flavors.
- SoftLayer provides a predefined set of values for CPU, RAM, and disk. Check the possible values in the SoftLayer documentation or in the SoftLayer management portal.
- If the flavor definitions (CPU, RAM and disk) do not match the actual supported sizes within IBM SoftLayer, the results of the quota calculation might not reflect the actual sizes within the remote cloud.

Configuring quotas

The default quotas are configured in `config.json` and the project quotas through the IBM Cloud Orchestrator user interface.

There are two types of quota definitions in the Public Cloud Gateway:

- A default quota set that is used if no project level quotas are defined.
- Project specific quotas.

Quotas are on a per project per region level.

Configuring default quota support

The default quotas setting is stored within the `config.json` file that is located in the `/opt/ibm/ico/wlp/usr/servers/pcg` subdirectory where the Public Cloud Gateway component is installed.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

The file is in JSON format. This is the quota section of the file:

```
{
    "defaultQuota":{
        "instances":"100",
        "cores":"100",
        "ram":"262144",
        "gigabytes":"512",
        "volumes":"2048",
        "key_pairs":"100"
    }
}
```

To modify the `config.json` file, as `root`, open it in a text editor and change the values:

1. Connect to the IBM Cloud Orchestrator Server through SSH. Default location: `/opt/ibm/ico/wlp/usr/servers/pcg`.

2. Restart the Public Cloud Gateway by submitting the following command as `root` on the command line: `service pcg restart`.

Configuring project quota

You can manage project quotas by using the OpenStack Dashboard. For more information, see the following topics:

- "Editing the domain quotas" on page 178
- "Configuring project quotas" on page 185

Note: You cannot delete project quotas from the OpenStack Dashboard. You can only create or modify them. The Public Cloud Gateway only supports a subset of the quotas as described in "Quota support overview" on page 257.

For Public Cloud Gateway regions, because quota values are stored in the Public Cloud Gateway cache directory, if you update the quota on an OpenStack project, to make the changes effective, perform the following steps:

1. In the `/opt/ibm/ico/wlp/usr/servers/pcg/var/cache` directory, delete the quota file named `*<tenant_Id><region_Name>*.quota`, if existing.
2. Restart the Public Cloud Gateway service.

General assumptions

- Quota calculation is done based on the flavors of the virtual machine instances, the additional disks and `sshkey` usage. It is required that the flavor definitions match to the assumptions of the remote clouds. For more information, see "Configuring flavors" on page 265.
- Quota calculation is done at certain intervals configurable in `config.json`. This means that there might be time frames were the actual situation in the remote cloud is different than reported through the quota management. For more information, see "Configuring caching" on page 267.

Configuring caching

Cache management with external clouds is required as the remote clouds have denial of service and API rate limits management in place.

There are two types of caches within Public Cloud Gateway:

- Caching of resources
- Caching of quota actuals

Both are configured within the `config.json` files in different sections.

The cache values describe when the public cloud internal caches are refreshed if no modifying resource requests are performed. Modifying resources requests are create, modify, and delete style requests.

A modifying request would invalidate the caches for the triggering URL (region and project).

The values must be adapted so that the least number of API calls are done against the remote clouds without impacting the responsiveness of the Public Cloud Gateway.

Note: The caching impacts the responsiveness of updates within the IBM Cloud Orchestrator UI. As a result there is a time difference between when the management consoles of the remote clouds show an update of a status or the completion of an action that is compared to the update in the IBM Cloud Orchestrator UI. The difference is at least the time interval that is configured for the cache refresh.

Caching of resources:

```
{
    "cacheTimeout":{
        "serversTimeout":"180",
        "glanceImagesTimeout":"180",
        "availabilityZoneTimeout":"180",
        "volumesTimeout": "180",
        "keypairTimeout": "180"
    }
}
```

All values are in seconds.

serversTimeout
Defines the cache refresh interval for virtual machine instance-related data.

glanceImagesTimeout
Defines the cache refresh interval for image-related data. For example, if you add a new image to the IaaS, this is the time until it shows up in a "glance image-list" for that region.

availabilityZoneTimeout
Defines the cache refresh interval for changes that are related to availability-zones. Changes are normally infrequent because a new data center is added by the IaaS provider.

volumesTimeout
Defines the cache refresh interval for volume-related data.

keypairTimeout
Defines the cache refresh interval for key pair-related data.

Caching of quota actuals:

```
{
    QuotaTimeouts
    {
        "quotaTimeout":{
            "serverQuotaTimeout":"600",
            "volumeQuotaTimeout":"600",
            "keypairQuotaTimeout":"600"
            }
    }
}
```

All values are in seconds. Each entry defines a refresh interval for the quota calculation:

serverQuotaTimeout
Defines the refresh interval in seconds for virtual machine instance-related quota elements.

volumeQuotaTimeout
Defines the refresh interval in seconds for volume-related quota elements.

keypairQuotaTimeout
Defines the refresh interval in seconds for key pair-related quota elements.

Note: The quota timeout determines the maximum cycle where the quota management system refreshes the values from the remote cloud. During the defined timeframe the values between the remote cloud and the reported values from the quota system might be out of sync.

Note: All the clouds have denial of service detection. Each refresh of the quotas is counted in the number of remote API calls. Setting the timeout too low might trigger a denial of service situation with the remote clouds which might disable the account.

Changing the Keystone administrator password

You can change the Keystone administrator password using one of the following scenarios in the Public Cloud Gateway.

- The Keystone administrator password is changed. Public Cloud Gateway stores the credentials to logon to Keystone in the `admin.json` file (`/opt/ibm/ico/wlp/usr/servers/pcg`):

```
{
"auth":
  { "passwordCredentials":
    {
      "username":"xxxx",
      "password":"yyyyy"
    },
    "tenantName":"zzzz",
    "domainName": "ddddd"
  }
}
```

The password is encrypted using the `encryptPassword.sh` located in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory. For information, see “Command-line interface scripts” on page 283 and “Failure to generate admin token” on page 384.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

- The password to access a remote cloud is changed. The access information to the remote clouds is stored in the `credentials.json` file in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory.

  For Amazon AWS EC2, see “Configuring the Public Cloud Gateway for Amazon EC2” on page 272.

  For SoftLayer, see “Configuring the Public Cloud Gateway for SoftLayer” on page 277.

  The password is encrypted using the `encryptPassword.sh` located in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory. For information, see “Command-line interface scripts” on page 283.

Changing a region name

You can change a region name.

Go to the `/opt/ibm/ico/wlp/usr/servers/pcg` directory in the IBM Cloud Orchestrator Server and open the `config.json` property file. Replace the old name with new one.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

For example, to show the change of the region name from `EC2-001` to `EC2region` for an EC2 region. The original EC2 region is:

```
"ec2":[
                    {
                        "name":"EC2-001",
                        "url":"https://ec2.us-east-1.amazonaws.com/",
                        "enabled":true
                    }
```

Change the region name:

```
"ec2":[
                    {
                        "name":"EC2region",
                        "url":"https://ec2.us-east-1.amazonaws.com/",
                        "enabled":true
                    }
```

Configure IBM Cloud Orchestrator for the new region and remove the entry for the old region as documented below.

Restart the Public Cloud Gateway by using the **`service pcg restart`** command. For more information about starting the Public Cloud Gateway, see “Command-line interface scripts” on page 283.

Delete the services of the old region from keystone:

```
source ~/keystonerc

keystone endpoint-list

+----------------------------------+-----------
|                id                |   region
+----------------------------------+-----------
| 0ff9e584b3d04c56af32e7b43ad5324d |  EC2-001
| 11924c78eca949ae939f2309a4e21bf9 |  EC2-001
| 187dbc8c68b74d5f8e098d4c61544d0b | RegionOne
| 19ded8422da445a7b6ceb0ce6d3c5f5e | RegionOne
| 3d8ff61f86f64838be5000b7efd60b89 | RegionOne
| 7f5a578d80684fcaaa473c9012ba7f46 |  EC2-001
| bf8de63072ae453fb5ddf8b3027945cf | RegionOne
| e0a8c3d7c821424e94a0ef17c8c1a383 |  EC2-001
```

```
+----------------------------------+-----------

+---------------------------------------------------------
|                         publicurl
+---------------------------------------------------------
|            http://ico-server:5000/v3
| http://ico-server:9797/EC2-001/v1/%(tenant_id)s
|      http://ico-server:8776/v1/%(tenant_id)s
|      http://ico-server:8774/v2/%(tenant_id)s
|              http://ico-server:9292/
| http://ico-server:9797/EC2-001/v2.0/%(tenant_id)s
|              http://ico-server:5000/v3
|       http://ico-server:9797/EC2-001/v2.0
+---------------------------------------------------------

+---------------------------------------------------------
|                     internalurl
+---------------------------------------------------------
|            http://ico-server:5000/v3
| http://ico-server:9797/EC2-001/v1/%(tenant_id)s
|      http://ico-server:8776/v1/%(tenant_id)s
|      http://ico-server:8774/v2/%(tenant_id)s
|              http://ico-server:9292/
| http://ico-server:9797/EC2-001/v2.0/%(tenant_id)s
|              http://ico-server:5000/v3
|       http://ico-server:9797/EC2-001/v2.0
+---------------------------------------------------------

+---------------------------------------------------------+----------------------------------+
|                       adminurl                          |            service_id            |
+---------------------------------------------------------+----------------------------------+
|           http://ico-server:35357/v3                    | 40a0d00ad6d34cfc8a5c412c61cb3e33 |
|  http://ico-server:9797/EC2-001/v1/%(tenant_id)s        | 372311fb67564e41987038d587c6a539 |
|      http://ico-server:8776/v1/%(tenant_id)s            | 372311fb67564e41987038d587c6a539 |
|      http://ico-server:8774/v2/%(tenant_id)s            | b6155b46d8d1463185fdfbddb05f18b5 |
|             http://ico-server:9292/                     | 1f3d30e2fcf04bdd908969a987722acc |
| http://ico-server:9797/EC2-001/v2.0/%(tenant_id)s       | b6155b46d8d1463185fdfbddb05f18b5 |
|           http://ico-server:35357/v3                    | 40a0d00ad6d34cfc8a5c412c61cb3e33 |
|      http://ico-server:9797/EC2-001/v2.0                | 1f3d30e2fcf04bdd908969a987722acc |
+---------------------------------------------------------+----------------------------------+
```

Delete all the endpoints that are related for the old region EC2-001 with the following example command:

```
keystone endpoint-delete 0ff9e584b3d04c56af32e7b43ad5324d
```

Restarting the Public Cloud Gateway

You might need to restart the Public Cloud Gateway.

The Public Cloud Gateway runs as a service on the IBM Cloud Orchestrator Server. Some configuration tasks require a restart of the Public Cloud Gateway to activate changes:

- Changing a region name
- Changing a keystone administrator password
- Configuring caching
- Configuring default quotas
- Changing flavors
- Changing region settings

To restart the Public Cloud Gateway, complete the following steps:

1. Log on as root by using SSH.
2. Check whether the Public Cloud Gateway is running by executing as `root`: **`service pcg status`**.
3. Restart the Public Cloud Gateway by executing as `root`: **`service pcg restart`**.
4. Check the log of the Public Cloud Gateway for any errors and exceptions: `less /opt/ibm/ico/wlp/usr/servers/pcg/logs/pcg.log`.

   When you run `encryptPassword.sh`, a `trace.log` file is generated at `/opt/ibm/ico/wlp/usr/servers/pcg/logs/`.

> **Note:** The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

Note: For commands to start and stop Public Cloud Gateway server manually, see "Managing the services manually" on page 160.

Remote Cloud API proxy configuration

The Public Cloud Gateway requires connectivity to the remote cloud API endpoints for Amazon AWS EC2 and SoftLayer.

For scenarios where no direct connection to the internet from the IBM Cloud Orchestrator Server is available, the Public Cloud Gateway provides the capability to specify a proxy server in the `config.json` file.

It is possible to define the following proxy servers:

- A default proxy server
- A proxy server for Amazon EC2
- A proxy server for SoftLayer

There is a new main section in the `/etc/config.json` file of the Public Cloud Gateway. This is sample content to describe the structure and properties of the configuration:

```
"proxy":{
         "default":{
                      "host":"proxy.local",
                      "port":"3128",
                      "userid":"xxxxx",
                      "password":"yyyyy"
                   },
         "ec2":{
                      "host":"localhost",
                      "port":"3128"
               },
         "softlayer":{
                      "host":"127.0.0.1",
                      "port":"9090",
                      "userid":"xxxxx",
                      "password":"yyyyy"
                     }
        },
```

The `default` entry within the proxy definition defines the default proxy. This definition is used if there is no proxy definition for the specific remote cloud type:

- The `ec2` entry defines the specific proxy for all regions of type Amazon AWS EC2.
- The `softlayer` entry defines the specific proxy for all regions of type SoftLayer.

Table 19. Parameters that are used in the proxy definition in the `config.json` file

Parameter	Description
`host`	This is a required parameter. It is the host name or IP address of the proxy server. **Note:** If a host name is provided, it is required that the host name can be resolved to an IP address.
`port`	This is a required parameter. It is the port on the host where the proxy server is reachable. The standard port is 3128 in many proxy implementations.

Table 19. Parameters that are used in the proxy definition in the config.json file (continued)

Parameter	Description
userid	This is an optional parameter. If specified, it specifies the user ID that must be used to contact the proxy server. **Note:** If a user ID is specified, the password property is required.
password	This is an optional parameter. If specified, it specifies the password that must be used to contact the proxy server. The value of the parameter must be encrypted with the encryptPassword.sh. The encrypted value must be specified as the value of this parameter. **Note:** If a password is provided, the userid property is required.

Limitations for Amazon EC2

- Amazon EC2 provides only http or https proxy support.
- The capabilities are limited to the support within the Amazon java client binding in version 1.5.8.

Limitation for SoftLayer

- Only http or https proxy support is available.

Managing Amazon EC2

The Public Cloud Gateway is not preconfigured for use with Amazon Elastic Compute Cloud (Amazon EC2) as part of the IBM Cloud Orchestrator. You must complete certain configuration tasks before using the Public Cloud Gateway.

1. Familiarize yourself with the Public Cloud Gateway. See "Public Cloud Gateway overview" on page 249.
2. Check that prerequisites are met. See "Prerequisites" on page 260.
3. Configure the Public Cloud Gateway for Amazon EC2. See "Configuring the Public Cloud Gateway for Amazon EC2" on page 272.
4. Create a supported image. See "Creating a supported image" on page 261.
5. Configure quotas. See "Configuring quotas" on page 266.

For information about post-configuration steps, see "Performing post-configuration tasks" on page 282.

Configuring the Public Cloud Gateway for Amazon EC2

You can configure the Public Cloud Gateway for Amazon EC2.

Some configuration steps are required in the following files to add a region and configure the credentials for a project:

- config.json
- credentials.json

Note: The examples that are shown in this section are only pieces of the config.json and credentials.json files that are required for Amazon EC2-specific configurations. Both files contain extra sections that you must not modify or delete as part of the Amazon EC2 configuration.

Configure regions in the `config.json` file

Go to the `/opt/ibm/ico/wlp/usr/servers/pcg/etc` directory and open the `config.json` file.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

The following code of the `config.json` file is relevant for the Amazon EC2 region configuration:

```
{
      "vcenters":{
        "ec2":[
          {
            "name":"EC2-US-EAST-NORTHERN-VIRGINIA",
            "url":"https://ec2.us-east-1.amazonaws.com",
            "enabled":true
          },
          {
            "name":"EC2-US-WEST-OREGON",
            "url":"https://ec2.us-west-2.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-US-WEST-NORTHERN-CA",
            "url":"https://ec2.us-west-1.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-EU-IRELAND",
            "url":"https://ec2.eu-west-1.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-EU-FRANKFURT",
            "url":"https://ec2.eu-central-1.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-AP-SINGAPORE",
            "url":"https://ec2.ap-southeast-1.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-AP-TOKYO",
            "url":"https://ec2.ap-northeast-1.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-AP-SYDNEY",
            "url":"https://ec2.ap-southeast-2.amazonaws.com",
            "enabled":false
          },
          {
            "name":"EC2-SA-SAOPAULO",
            "url":"https://ec2.sa-east-1.amazonaws.com",
            "enabled":false
          }
        ],

      }
    }
```

The cloud region configuration is described in the vCenters section. Each region is specified by using three key-value pairs: name, url, and enabled.

The parameters in the `config.json` file are explained in the following table. Update the enabled parameter to **true** if you want to specify that a particular region is made available to the users of IBM Cloud Orchestrator.

Parameter	Description
name	Name of the region as it appears in keystone.

Parameter	Description
url	Amazon EC2: The URL of the Amazon EC2 vCenter that must be associated with the region. Default Amazon EC2 endpoints are defined. The data center is part of the URL. For example, `https://ec2.ap-southeast-2.amazonaws.com`, where `ap-sourtheast-2` is the Amazon data center.
enabled	Amazon EC2: Set to `true` if that data center is to be made available to the users of IBM Cloud Orchestrator. Set to `false` if that data center is not available. Do not use quotation marks.

Note: You must add a mapping for the project of the cloud administrator to the `credentials.json` file. The default is `admin`. If this entry is missing, you cannot add the availability zone to the domain through the OpenStack Dashboard.

```
{
            "tenantName":"admin",
            "access_key_ID":"xxx",
            "secret_access_key":"xxx"
          },
```

where *xxx* is a valid set of credentials to access your Amazon EC2 account.

Extra properties for Amazon EC2 on the region level. Example for SAOPAULO region:

```
{
  "name":"EC2-SA-SAOPAULO",
  "url":"https://ec2.sa-east-1.amazonaws.com",
  "enabled":false / true,
  "ImageType" : "cloud-init" or "scp-init",
}
```

Parameter	Description
ImageType	Defines on a region level what image activation type must be returned for images. The value is only used for images that are not already tagged with an image type.

If your account does support the capability to place virtual machines into distinct subnets of a non-default VPC, there are two properties to control that placement:

Parameter	Description
vpc	The ID of the configured non-default VPC where the virtual machines are placed.
privateNetworkOnly	Controls whether the machines get a public IP address. Valid values are **true** or **false**. If true, the virtual machine has no public IP address, otherwise it gets a public IP address from an Amazon AWS EC2 provided pool. If the property is not set in the region definition, the default is false. The property is supported in case of DefaultVPC and non-default VPC.

This capability is available when the only supported platform for you account is VPC. It is not enabled when other supported platforms are listed, for example, EC2. You can check the supported platform of your account on the EC2 dashboard in the **Account Attributes** section.

Be aware that in addition to the configuration in this file, more configuration tasks are necessary in your Amazon VPC account to use the non-default VPC support. See “Configuring subnets and security groups in a non-default VPC region” on page 276.

Configure cloud credentials in the `/opt/ibm/ico/wlp/usr/servers/pcg/etc/credentials.json` file

This file is used to specify Amazon EC2 credentials for each project. For more information about defining projects, see “Managing projects” on page 180. The Amazon EC2 credentials are mapped to specific projects in IBM Cloud Orchestrator. These mappings are specified in the `credentials.json` configuration file.

Go to the `/opt/ibm/ico/wlp/usr/servers/pcg/etc` directory and open the `credentials.json` file:

```
{
      "cred":{
        "ec2":[
          {
            "tenantName":"demo",
            "access_key_ID":"xxx",
            "secret_access_key":"xxx"
          },
          {
            "tenantName":"admin",
            "access_key_ID":"xxx",
            "secret_access_key":"xxx"
          },
          {
            "tenantID":"xxxxxx",
            "access_key_ID":"xxx",
            "secret_access_key":"xxx"
          },
          {
            "tenantName":"*",
            "access_key_ID":"xxx",
            "secret_access_key":"xxx"
          }
        ]
      }
    }
```

The parameters in the `credentials.json` file are explained in the following table. Update these parameters if you want to specify credentials to project mappings and define which credentials must be used for the different projects specified.

Parameter	Description
tenantName	The OpenStack project entity, also known as tenant. The following options to identify a project exist: • OpenStack project ID • OpenStack project name • Wildcard * to match any project **Note:** Due to multi-domain support, a project name might not be unique. If project names are not unique in your environment, you must use the project ID to identify the project by setting the **`tenantID`** parameter in the `credentials.json` file. To get the project ID, use the OpenStack Dashboard identity section.
access_key_ID	The Amazon EC2 access key for the project.

Parameter	Description
secret_access_key	The Amazon EC2 secret access key that is used for the project. This value must be encoded by using the `encryptpassword.sh` script that is available in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory.
region	The region name as defined in the `config.json` file. The region parameter is optional. If this parameter is set, the mapping is restricted to this specific region. If it is not set, the mapping is valid for all regions that are defined for the specific cloud type in the `config.json` file. Replace `yyy` in the following example with the value of the name parameter as it is defined in the `config.json` file. Example statement: `{ "tenantName":"*", "region": "yyy", "access_key_ID":"xxx", "secret_access_key":"xxx" }`

Note: You must add a mapping for the project of the cloud administrator to the `credentials.json` file. The default is `admin`. If this entry is missing, you cannot add the availability zone to the domain by using the OpenStack Dashboard:

```
{
   "tenantName":"admin",
   "region": "yyy",
   "access_key_ID":"xxx",
   "secret_access_key":"xxx"
},
```

where `xxx` is a valid set of credentials to access your Amazon AWS EC2 account.

Procedure to activate configuration changes:

1. Restart the Public Cloud Gateway by using the **`pcg restart`** command. For more information, see "Command-line interface scripts" on page 283.
2. Run the script `refreshEndpoint.sh` in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory to clean up caches that are related to region or endpoint information. See"Command-line interface scripts" on page 283.
3. Check the Public Cloud Gateway log in the`/opt/ibm/ico/wlp/usr/servers/pcg/logs/pcg.log` file for problems.

Configuring subnets and security groups in a non-default VPC region

You can configure subnets and security groups in a non-default VPC region.

If the non-default VPC support is enabled in one of your regions, you must tag at least one subnet in each availability zone to be used as a default subnet in which the virtual machines deployed in that region and availability zone are placed. Do this in the Amazon VPC console of your account by adding a tag to the subnet with key `TenantUUId` and value "`*`". The value "`*`" indicates that this subnet is used for virtual machines for all projects.

It is possible to overwrite the "`privateNetworkOnly` definition on the region level per subnet. If you want to do this, add a tag with the name `privateNetworkOnly` and a value of either `true` or `false` to a subnet. The definition on the subnet has precedence over the definition on the region.

If you want to place virtual machines of a distinct project in another subnet, you can add the OpenStack tenant ID of your project as value for the `TenantUUId` tag. You can only have one of these tags on a given subnet. Multiple subnets tagged with the same OpenStack tenant ID are not allowed within a single availability zone.

Additionally, you can tag one of the existing security groups with the key `TenantUUId` and value "*" to be the default security group for all servers provisioned in this region. If you want to place virtual machines of a distinct project in another security group, you can add the OpenStack tenant ID of your project as a value for the `TenantUUId` tag. You can only have one of those tags on a given security group. Multiple security groups tagged with the same OpenStack tenant ID are not allowed within a single VPC. As opposed to with subnets, you do not have to tag a security group. In this case, the default security group of the VPC is assigned.

Managing SoftLayer

The Public Cloud Gateway is not preconfigured for use with SoftLayer as part of IBM Cloud Orchestrator. You must perform certain configuration tasks before using the Public Cloud Gateway.

Before you begin

Familiarize yourself with the Public Cloud Gateway. See "Public Cloud Gateway overview" on page 249.

Procedure

1. Check that prerequisites are met. See "Prerequisites" on page 260.
2. You can integrate SoftLayer using thePublic Cloud Gateway. See "Integrating SoftLayer" on page 277.
3. You can configure the Public Cloud Gateway for SoftLayer. See "Configuring the Public Cloud Gateway for SoftLayer" on page 277.
4. Create a supported image. See "Creating a supported image" on page 261.
5. Configure quotas. See "Configuring quotas" on page 266.

What to do next

For information about post-configuration steps, see "Performing post-configuration tasks" on page 282.

Integrating SoftLayer

You can integrate SoftLayer using the Public Cloud Gateway.

Before you begin

For general information about SoftLayer, see http://www.softlayer.com/.

Procedure

1. Set up an account in SoftLayer and create one or more user IDs. Each ID has its own unique password and API access key. The API access key is required in configuring SoftLayer integration in the Public Cloud Gateway.
2. Create IBM Cloud Orchestrator-ready images.
3. Set up the following configuration files (`admin.json, config.json, credentials.json, flavors.json`) as described in "Configuring the Public Cloud Gateway" on page 253.
4. Start or restart the Public Cloud Gateway.

Configuring the Public Cloud Gateway for SoftLayer

You can configure the Public Cloud Gateway for SoftLayer.

Before you begin

Some configuration steps are required in the following files to add a region and configure the credentials for a project:

- `config.json`.

- credentials.json.

Note: The examples that are shown in this section are only pieces of the config.json and credentials.json files that are required for SoftLayer-specific configurations. Both files contain extra sections that you must not modify or delete as part of the SoftLayer configuration.

Configure regions in the config.json file:

Go to the /opt/ibm/ico/wlp/usr/servers/pcg/etc directory and open the config.json file. The following piece of code of the config.json file is relevant for the SoftLayer region configuration:

Note: The /opt/ibm/ico/wlp/usr/servers/pcg is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

```
{
      "vcenters":{
        "softlayer":[
          {
            "name":"SL-Dallas05",
            "dataCenter" : "Dallas 5",
            "url":"https://api.softlayer.com/",
            "enabled":true
          },
          {
            "name":"SL-Dallas06",
            "dataCenter" : "Dallas 6",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-SanJose",
            "dataCenter" : "San Jose 1",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Amsterdam",
            "dataCenter" : "Amsterdam 1",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Seattle",
            "dataCenter" : "Seattle",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-WashingtonDC",
            "dataCenter" : "Washington 1",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Singapore",
            "dataCenter" : "Singapore 1",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Dallas01",
            "dataCenter" : "Dallas 1",
            "url":"https://api.softlayer.com/",
            "enabled":true
          },
          {
            "name":"SL-HongKong",
            "dataCenter":"Hong Kong 2",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Houston",
            "dataCenter":"Houston 2",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Toronto",
            "dataCenter":"Toronto 1",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-London",
            "dataCenter":"London 2",
            "url":"https://api.softlayer.com/",
            "enabled":false
          },
          {
            "name":"SL-Melbourne",
            "dataCenter":"Melbourne 1",
            "url":"https://api.softlayer.com/",
            "enabled":false
          }
        ]
      }
    }
```

The cloud region configuration is described in the vCenters section. Each region is specified by using three key-value pairs: name, url, and enabled. The parameters in the `config.json` file are explained in the following table. Update the enabled parameter to *true* if you want to specify that a particular region is made available to the users of IBM Cloud Orchestrator.

Table 20. Parameters that are used in the `config.json` file

Parameter	Description
name	Name of the region as it appears in keystone.
dataCenter	Name of the SoftLayer data center the region is connected to.
url	SoftLayer: URL of SoftLayer API server. For SoftLayer API server accessible through public IP addresses, use `https://api.softlayer.com/`. For accessing the SoftLayer API server on the SoftLayer private network, use `https://api.service.softlayer.com/`.
enabled	SoftLayer: Set to `true` if that datacenter is to be made available to the users of IBM Cloud Orchestrator. Set to `false` if that datacenter is not available. The value of `enabled` is either true or false: do not use quotation marks.

Configure cloud credentials in the `/opt/ibm/ico/wlp/usr/servers/pcg/etc` file.

More properties available for SoftLayer regions. Example for Singapore datacenter:

```
{
        "name":"SL-Singapore",
        "dataCenter" : "Singapore 1",
        "url":"https://api.softlayer.com/",
        "enabled":false / true,
        "ImageType" : "cloud-init" or "scp-init",
        "privateNetworkOnly" : false / true,
        "primaryVlanID" : "600516",
        "backendVlanID": "600518"
}
```

Table 21. Parameters that are used in the `config.json` file

Parameter	Description
ImageType	Defines on a region level what image activation type must be returned for images. The value is only used for images that are not already tagged with an image type.
privateNetworkOnly	If it is set to `true`, the virtual machine has a single private NIC (backend). If it is set to `false`, the virtual machine has a private (backend) and a public (primary) NIC. The default value is `false`.
primaryVlanID	VLAN ID that connects the virtual machine to the internet (public VLAN). VLAN ID is the SoftLayer resource ID that describes a VLAN. If **`primaryVlanID`** is not specified, the SoftLayer default is used. For the public network, it is critical to configure the firewall with the appropriate rules for the user ID. The Public Cloud Gateway performs the deployment with the configured user ID (as specified in the `credentials.json` file) so that the firewall rules as defined for that user are applied for the public network of the provisioned virtual machine (for example, only HTTP traffic, port 80 allowed).
backendVlanID	VLAN ID that connects the virtual machine to the management network (private VLAN). VLAN ID is the SoftLayer resource ID that describes a VLAN. If **`backendVlanID`** is not specified, the SoftLayerdefault is used.

Note: To obtain the correct VLAN ID, perform the following steps:

1. Log on to SoftLayer portal at https://control.softlayer.com/.
2. Go to the VLANs page at https://control.softlayer.com/network/vlans.
3. Choose the VLAN that you want to use for provisioning and select it to open the VLAN details.
4. Copy the VLAN ID from the browser URL. For example, if the URL is `https://control.softlayer.com/network/vlans/600516` then `600516` is the correct ID. Do not confuse the VLAN ID with the VLAN number displayed on the web page.

Configure cloud credentials in the `/opt/ibm/ico/wlp/usr/servers/pcg/etc/credentials.json` file:

This file is used to specify SoftLayer credentials for each project. For more information about defining projects, see "Managing projects" on page 191. The SoftLayer credentials are mapped to specific projects in IBM Cloud Orchestrator. These mappings are specified in the `credentials.json` configuration file.

Go to the `/opt/ibm/ico/wlp/usr/servers/pcg/etc` directory and open the `credentials.json` file:

```
{
     "cred":{
       "softlayer":[
         {
           "tenantName":"admin",
           "user_id":"xxx",
           "api_access_key":"xxx"
         },
         {
           "tenantName":"demo",
           "user_id":"xxx",
           "api_access_key":"xxx"
         },
         {
           "tenantName":"tenant1",
           "user_id":"xxx",
           "api_access_key":"xxx"
         },
         {
           "tenantName":"tenant2",
           "user_id":"xxx",
           "api_access_key":"xxx"
         }
       ]
     }
   }
```

The parameters in the `credentials.json` are explained in the following table. Update these parameters if you want to specify credentials to project mappings and define what credentials must be used for the different projects specified.

Table 22. Parameters that are used in the `credentials.json` file

Parameter	Description
`tenantName`	The OpenStack project entity, also known as tenant. The following options to identify a project exist: • OpenStack project ID • OpenStack project name • Wildcard `*` to match any project **Note:** Due to multi-domain support, a project name might not be unique. If project names are not unique in your environment, you must use the project ID to identify the project by setting the **`tenantID`** parameter in the `credentials.json` file. To get the project ID, use the OpenStack Dashboard identity section.
`user_id`	The SoftLayer account user ID used for the project.
`api_access_key`	The SoftLayer API access key. This value must be encoded by using the `encryptpassword.sh` script that is available in the `/opt/ibm/ico/wlp/usr/servers/pcg` directory.

`region`	The region name as defined in `config.json`. The region parameter is optional. If this parameter is set, the mapping is restricted to this specific region. If it is not set, the mapping is valid for all regions that are defined for the specific cloud type in the `config.json` file.
	Replace yyy in the following example with the value of the name parameter as it is defined in the `config.json` file. Example statement:
	`{`
	`            "tenantName":"*",`
	`            "region": "yyy",`
	`            "user_id":"xxx",`
	`            "api_access_key":"xxx"`
	`          }`

Note: You must add a mapping for the project of the cloud administrator to the `credentials.json` file. The default is `admin`. If this entry is missing, you cannot add the availability zone to the domain by using the OpenStack Dashboard.

```
{
            "tenantName":"admin",
            "region": "yyy",
            "user_id":"xxx",
            "api_access_key":"xxx"
          },
```

where *xxx* is a valid set of credentials to access your SoftLayer account.

Activating configuration changes:

Procedure

1. Restart the Public Cloud Gateway by using the **`service pcg restart`** command. For more information about starting the Public Cloud Gateway, see "Command-line interface scripts" on page 283.
2. Run the script `refreshEndpoint.sh` in the `/opt/ibm/ico/wlp/usr/servers/pcg/etc` to clean up caches that are related to region or endpoint information.
3. Check the Public Cloud Gateway log in `/opt/ibm/ico/wlp/usr/servers/pcg/logs/pcg.log` for problems.

Performing post-configuration tasks

You must complete post-configuration tasks after you configure the Public Cloud Gateway.

Procedure

For deployment using a single virtual machine, complete the following steps:

a) Add the newly-defined Public Cloud Gateway managed region or availability zone to:

Domain
See "Assigning a zone to a domain" on page 177.

Project
See "Assigning a zone to a project" on page 184.

b) Register a new SSH key for deployment. See "Registering a key pair" on page 224.

c) If you want to use additional disks during deployment, you must create volumes for the project. You can create volumes using the *Open Stack Cinder Storage Volumes toolkit of Content Development Guide*.

d) Add **`cloud-init`** to Linux operating system images, as described in "Creating Linux base images" on page 241.

e) Deploy the virtual machine as described in "Deploying a virtual machine" on page 213.

Results

You can now deploy a virtual machine by using the Public Cloud Gateway.

Reference

This section provides reference information for the Public Cloud Gateway.

Key pairs

Key pairs are needed to access the virtual machines that you deployed. When you deploy a virtual machine, these keys are injected into the instance to allow password-less SSH access to the instance.

The `default` key pair that is created from the Self-service user interface in Amazon EC2 regions is appended with the user ID of the user that created the key pair. For example, if the user creating the key pair in the Self-service user interface is admin, the name of the key pair that is created in Amazon EC2 is default_admin. For information about managing key pairs, see "Managing key pairs" on page 224.

Command-line interface scripts

Command Line Interface (CLI) scripts are available in the /opt/ibm/ico/wlp/usr/servers/pcg directory. These scripts are used for manual tasks, for example, starting the Public Cloud Gateway, encrypting a password, changing port numbers, and so on.

encryptPassword.sh *plaintext_password*

Writes an encrypted password to stdout. This script is used for encrypting passwords and access keys that are used in the admin.json and credentials.json files, which are located in the /opt/ibm/ico/wlp/usr/servers/pcg directory. The command must be run in the directory where the encryptPassword.sh script is located.

Note: The /opt/ibm/ico/wlp/usr/servers/pcg is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

service pcg start

Starts the Public Cloud Gateway server with the default settings.

service pcg stop

Stops the Public Cloud Gateway server.

service pcg restart

Restarts the Public Cloud Gateway server.

refreshEndpoint.sh *admin_userid admin_password bpm_hostname*:*bpm_port*

Refreshes the IBM Cloud Orchestrator endpoint cache, where

- *admin_userid* is the user name of the Cloud Administrator user.
- *admin_password* is the password of the specified Cloud Administrator user.
- *bpm_hostname* is the host name of the Business Process Manager server. Business Process Manager usually runs on the same host as the Public Cloud Gateway.
- *bpm_port* is the port of the Business Process Manager server. The default port is 9443.

The refreshEndpoint.sh script must be run if region information is changed within the Public Cloud Gateway config.json file. If the command was successful, the command output includes the following line:

```
HTTP/1.1 204 No Content
```

To see any new Regions or Availability Zones in the IBM Cloud Orchestrator user interface lists, users must log out of the user interface and log back in again.

Password authentication on Amazon EC2 images

You can allow password authentication on Amazon EC2 images.

Usually, Amazon Linux images have password and root login disabled by default. Amazon AWS EC2 recommends to use SSH keys to access the images. The images are usually also sudo enabled.

You can enable password and root login using the following procedure:

1. "Update the cloud-init configuration file" on page 284 to allow root access and password login.
2. "Update the authorized_keys file" on page 284.
3. "Update the sshd_config file" on page 284 to enable password authentication and root login.

Note: Future Amazon updates to the images might require changes to the procedure.

Update the cloud-init configuration file

Make sure that the following lines are in the /etc/cloud/cloud.cfg file:

```
disable_root: false
ssh_pwauth:   true
```

These properties enable root login and password authentication in cloud-init. They are required to set the password via user-data.

Update the authorized_keys file

In the authorized_keys file, remove the command prefix and leave only the ssh-rsa statement. For example, change the following default content:

```
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login
as the user \"ec2-user\" rather than the user \"root\".';echo;sleep 10"
ssh-rsa <content of sshkey>
```

to the following content:

```
ssh-rsa <content of sshkey>
```

Update the sshd_config file

Log on to the Amazon EC2 image by using SSH and complete the following steps:

1. Edit the /etc/ssh/sshd_config file.
2. Update the following lines:

   ```
   PasswordAuthentication yes
   PermitRootLogin yes
   ```

3. Save the file.
4. Run the following command:

   ```
   sudo service sshd restart
   ```

Managing Microsoft Azure

You can manage Microsoft Azure as a remote cloud.

Microsoft Azure support consists of:

- A separate set of offerings in the IBM Cloud Orchestrator Self-Service Catalog to manage Microsoft Azure resources.
- A view of the Azure cloud services in **RESOURCES**, including actions.
- A view of the Azure deployments in **RESOURCES**, including actions.
- A view where you can register and manage Microsoft deployment artifacts.
- A view where you can register and manage Microsoft Azure regions.

A Microsoft Azure account with at least one active subscription is required. IBM Cloud Orchestrator communicates to Microsoft Azure using the REST API, and so a management certificate is required. For more information, see Certificates overview for Azure Cloud Services. The subscription must have at least one storage account including a container that is used to store deployment artifacts.

Note: If you want to use Microsoft Azure automation, you must enable it for all the regions in Microsoft Azure that you want to use. For more information, see Azure Automation overview.

You must perform certain configuration tasks before IBM Cloud Orchestrator can use Microsoft Azure subscriptions for provisioning.

Capabilities and limitations

There are the following capabilities and limitations for Microsoft Azure in IBM Cloud Orchestrator.

Capabilities

The following capabilities are supported:

- Add a Microsoft Subscription as Azure region in IBM Cloud Orchestrator.
- Deploy a cloud service using the `Deploy Azure Cloud Service` offering.
- View deployed cloud services using the **RESOURCES** view.
- Lifecycle management of deployed cloud services from the **RESOURCES** view, for example start, stop, and delete an Azure cloud service.
- View deployments in cloud services using the **RESOURCES** view.
- Lifecycle management of deployments in cloud services from the **RESOURCES** view, for example start, stop, and delete a deployment.

Limitations

The following limitations exist:

- Microsoft Azure resources are not visible within OpenStack.
- The OpenStack Dashboard is not supported.
- IBM Cloud Orchestrator dashboards do not show Azure resources. A Microsoft Azure subscription cannot be shared among IBM Cloud Orchestrator projects.

  Note: A Microsoft Azure subscription is uniquely assigned to a single IBM Cloud Orchestrator project.
- Deployed cloud stacks within Azure are only internet accessible using the public IP address of the related cloud service or using a dedicated VPN that requires deployment to a Microsoft Azure virtual network.
- With the functions provided by IBM Cloud Orchestrator, it is not possible to upload and use a certificate for a cloud service.
- The expiry dates of service management certificates are not enforced by Microsoft Azure. For more information, see https://msdn.microsoft.com/en-us/library/azure/ee460782.aspx.

Network planning for Microsoft Azure

The Microsoft Azure support requires a set of network configuration to successfully provision resources within the remote cloud. This topic provides an overview about which network configuration is assumed and required.

Access to Microsoft Azure REST API entry points

During the management lifecycle, the Microsoft Azure support requires access to the remote cloud REST API entry points for:

- The Microsoft Azure Service Management REST API
- The Microsoft Azure Storage Services REST API

Optional: Connectivity from IBM Cloud Orchestrator management stack to the provisioned cloud services

Only access to the Microsoft Azure REST APIs is needed for the management actions that are provided with the Microsoft Azure support.

Depending on the provisioned services, you might need access for ports and protocols that are needed for extra management scenarios on such cloud services.

Managing Microsoft Azure subscriptions

You can perform actions in IBM Cloud Orchestrator against Microsoft Azure subscriptions.

About this task

Microsoft Azure subscriptions are registered as regions in IBM Cloud Orchestrator. A region is a logical partition of cloud space for resource deployment. It is used to separate resources of teams and projects from one another.

You must register a Microsoft Azure subscription as a region in IBM Cloud Orchestrator before you use it for provisioning.

You can find the list of registered Microsoft Azure subscriptions in **CONFIGURATION** > **Domain** > **Regions**.

The **Action** menu shows the available actions that you can run with regions:

- **Register Azure Region**

To register a new region:

1. Go to **CONFIGURATION** > **Domain** > **Regions**.
2. Select the **Register Azure Region** action from the **Action** menu on the left and provide the required information in the action dialog.

To complete the registration process, the following information is required:

- The Azure Subscription ID.
- The management certificate that is used with the Microsoft Azure management API.
- Optionally, a new certificate can be generated.

  Note: The expiry dates of service management certificates that are given in that process are not enforced by Microsoft Azure, refer to: https://msdn.microsoft.com/en-us/library/azure/ee460782.aspx.

During the registration, provide:

- The name of the region. It must be unique across IBM Cloud Orchestrator.
- A description of the region.
- The Microsoft Azure locations that are used for deployment. Each location appears as an availability zone in the new region.
- The Storage Container is the URL of a container in a storage account that belongs to your subscription. It is used to store the templates that are used to create Cloud Service Deployments.

At the end of the registration, the subscription is assigned to the current IBM Cloud Orchestrator project. If you want to assign the subscription to a certain project, you must switch to the context of that project before starting the **Register Azure Region** action.

Note: Azure subscriptions cannot be shared across IBM Cloud Orchestrator projects.

Depending on the selection of Microsoft Azure regions, the list of available management actions that can be run is displayed:

- Removing an Azure Region:
  - This action is done to unregister a previously registered region.

    Note: This action does not remove any of the deployed resources in that region.
- Modifying an Azure Region:
  - This action is done to modify an existing region. This action might be used to add or remove locations as availability zones to the region at a later point in time.
  - Only the description, the container URL, and the list of locations can be modified.
- When you click a region, the details are displayed.

Note: These actions appear only if one or more Microsoft Azure regions are selected. For selected regions of a different type (for example, OpenStack), other actions might be available.

Registering and managing Microsoft Azure deployment package

You can register and manage Microsoft Azure cloud service deployment packages.

A Microsoft Azure cloud service deployment package consists of two files:

- A cloud service configuration file (`.cscfg`) providing the configuration settings for the cloud service and individual roles
- A service package file (`.cspkg`) containing the application code and the service definition

These files can be generated by using the provided Microsoft tool, Microsoft Visual Studio, or Microsoft Azure SDK.

In the IBM Cloud Orchestrator Self-service user interface, you can find the list of Microsoft Azure under **CONFIGURATION** > **Templates** > **Azure Deployment Packages**.

Depending on the selection of deployment packages, there are management actions that can be run:

- **Register Deployment Package**
- **Modify Deployment Package**
- **Delete Deployment Package**

When you click a Microsoft Azure deployment package, you see the details view.

You can register a new deployment artifact that is a Microsoft Azure deployment package that can be deployed by using the offering to deploy an Azure cloud service in the Self-Service Catalog.

To register a new Microsoft Azure deployment package, complete the following steps:

1. Select the action **Register Deployment Package**.
2. In the first screen, specify:
   - The name of the new deployment package to be registered
   - A description
   - The name of the associated cloud service package file
   - The name of the associated cloud service configuration file

   Note: For each Microsoft Azure region, the deployment package is used with the corresponding package file. The configuration file that is specified must be uploaded to the Microsoft Azure storage account that has been specified during creation of the Microsoft Azure region.
3. Complete the tabs on the second screen:

   Azure Cloud Service Deployment Package Source
   You can overwrite the package file name and configuration file name.

   Deployment Details
   Specify for which Microsoft Azure region the deployment package can be used: either for any region or for a selectable list of regions.

   Access Control List
   Specify and modify the access control for the Microsoft Azure deployment package, for example, which role in which project and domain has rights to use the deployment package.

A new Microsoft Azure deployment package is now registered.

Deploying Microsoft Azure resources

You can create a new Microsoft Azure deployment.

To create a new Microsoft Azure deployment, complete the following steps:

1. In the Self-Service Catalog, go to **Deploy cloud services > Deploy Azure Cloud Service**.

2. Select whether to deploy into an existing cloud service or to create a new cloud service and deploy into it.
3. If deployment into a new cloud service is selected, you can select the region and, depending on the region, the availability zone to deploy the new cloud service into.
4. If deployment into an existing cloud service is selected, you can select the cloud service and the deployment slot for the new deployment.
5. Enter the name of the cloud service, the deployment slot (either **Production** or **Staging**), and the name of the deployment if a new cloud service is selected. Select the cloud service template to be deployed.

   Note: You can only see the templates that have been registered either for this region or for all regions and for which the corresponding access rights have been defined.

Finally, a summary screen is displayed.

Viewing and managing Microsoft Azure resources

There are views in IBM Cloud Orchestrator that show deployed resources from Microsoft Azure subscriptions.

In the **RESOURCES** menu of the Self-service user interface, there are categories for Microsoft Azure resources:

- Azure Cloud Service
- Azure Deployments

Azure Cloud Services view

In the **Azure Cloud Services** view, you can see a list of all Microsoft Azure cloud services that can be filtered by the region with the following information:

- Name of the cloud service
- Status
- Last Update
- Description
- Region

Note: The list contains all cloud services for the specified region (subscription) and not just the cloud services that have been created by using IBM Cloud Orchestrator.

Depending on the selection of cloud services, the list of available management actions that can be run is displayed:

Add Deployment
To add a deployment to an existing cloud service.

Delete Azure Cloud Service
To delete a cloud service with all its deployments.

When you click a cloud service, details are displayed.

Azure Deployments view

In the **Azure Deployments** view, there is a list of all deployments that can be filtered by the region:

- Name of the deployment
- Status
- Description
- Deployment Slot
- Region
- Cloud Service the deployment is part of

Note: The list contains all deployments of all cloud services for the specified region (subscription) and not just the cloud services that have been created by using IBM Cloud Orchestrator.

Depending on the selection of deployments, the list of available management actions that can be run is displayed:

Start Azure Deployment
Only available for deployments in status `Suspended`.

Stop Azure Deployment
Only available for deployments in status `Running`.

Delete Azure Deployment
Only available for deployments in status `Suspended`.

When you click a deployment, details are displayed.

Chapter 12. Integrating

Learn how to integrate IBM Cloud Orchestrator with the following IBM products.

Integrating with IBM Tivoli Monitoring

To integrate IBM Cloud Orchestrator with IBM Tivoli Monitoring, you must prepare a base operating system, set up the required databases, install the Tivoli Monitoring components, and deploy the Monitoring Agents to monitor the IBM Cloud Orchestrator environment.

Preparing a base operating system

IBM Tivoli Monitoring 6.3.0.2 supports several operating systems, but the IBM Cloud Orchestrator solution is based on Red Hat Enterprise Linux (RHEL), which makes it an optimal system for setting up Tivoli Monitoring.

Before you begin

For a list of supported operating systems, see Supported operating systems.

For more information about hardware and software requirements for IBM Tivoli Monitoring, see Hardware and software requirements.

Procedure

1. You must install several `rpm` packages that are required by IBM Global Security Toolkit (GSKit). GSKit is deployed automatically with the Tivoli Monitoring installation and requires the following operating system patches:
   - `ksh-20091224-1.el6.x86_64.rpm`
   - `glibc-2.12-1.7.el6.i686.rpm`
   - `libgcc-4.4.4-13.el6.i686.rpm`
   - `nss-softokn-freebl-3.12.7-1.1.el6.i686.rpm`
2. Install the libraries that are required by the OS Monitoring Agent:
   - `libstdc++`
   - `libgcc`
   - `compat-libstdc++`

   Restriction: On a 64-bit system, you must have 32-bit and 64-bit versions of those libraries.

Database setup

IBM Tivoli Monitoring requires two databases, the Tivoli Enterprise Portal Server database and the Tivoli Data Warehouse database.

- The Tivoli Enterprise Portal Server database, or portal server database, stores user data and information that is required for graphical presentation on the user interface. The portal server database is created automatically during configuration of the portal server. It is always on the same computer as the portal server.
- The Tivoli Data Warehouse database, also called the warehouse database or data warehouse, stores historical data for presentation in historical data views. In a single-computer installation, the warehouse database is created on the same relational database management server that is used for the portal server database. In larger environments, it is best to create the warehouse database on a different computer from the portal server.

You can create a TEPS database on an embedded Derby database that is delivered with the Tivoli Monitoring installer. Warehouse database can be set on a DB2 or Oracle server. Thus, the best solution is to install a DB2 server on a Tivoli Monitoring server and use it for TEPS and Warehouse databases.

For more information about installing DB2, see the DB2 documentation.

Installing IBM Tivoli Monitoring

The installation of IBM Tivoli Monitoring requires several mandatory components. You can also install extra ones if you are planning to set up a dashboard environment or use products that support integration using OSLC.

About this task

For more information about Tivoli Monitoring and its components, see Components of the monitoring architecture.

For more information about installing and configuring Tivoli Monitoring, see High-level installation steps.

Procedure

1. You must install the following components of Tivoli Monitoring:
   - Hub Tivoli Enterprise Monitoring Server
   - Tivoli Enterprise Portal Server
   - Tivoli Enterprise Portal desktop client
   - The Warehouse Proxy Agent
   - The Summarization and Pruning Agent
2. If you plan to set up a dashboard environment, you can install extra components. For base installation, these features are not required and can be skipped or installed later:
   - Dashboard Application Services Hub (a Jazz for Service Management component)
   - IBM Infrastructure Management Dashboards for Servers
   - Tivoli Authorization Policy Server
   - tivcmd Command Line Interface for Authorization Policy

   Dashboard and JazzSM are new components of Tivoli Monitoring 6.3. Dashboard does not completely replace Tivoli Portal Client but is used to improve data presentation. TEPS is used for configuration issues.
3. If you plan to use the Performance Monitoring service provider to integrate with the Jazz for Service Management Registry Services component and other products that support integration using OSLC, install the following component. For base installation, this feature is not required and can be skipped or installed later:
   - Tivoli Enterprise Monitoring Automation Server

Packages used for installation

You need several packages to install IBM Tivoli Monitoring 6.3.0.2. All of its components can also be installed in silent mode.

The following packages are required to install IBM Tivoli Monitoring 6.3.0.2:

- CIQ3JEN - IBM Tivoli Monitoring V6.3.0.2 Base, Linux (64-bit Env.), English
- CIQ3PML - IBM Tivoli Monitoring V6.3.0.2 Dashboards for Servers and Authorization Policy Components Assembly Multiplatform, Multilingual
- CIQ3MML - IBM Tivoli Monitoring V6.3.0.2 Language Support Multiplatform Multilingual

All IBM Tivoli Monitoring components can be installed and configured in silent mode. First, you must install all products with one `silent_install` file and then configure each one of them with silent_config response files. For more information about modifying the files, see the following examples.

- IBM Tivoli Monitoring: modify the silent_install.txt file with the following information:

```
INSTALL_PRODUCT=ms
INSTALL_PRODUCT=cq
INSTALL_PRODUCT=hd
INSTALL_PRODUCT=sy
INSTALL_PRODUCT_TMS=all
INSTALL_PRODUCT_TPS=all
INSTALL_PRODUCT_TPW=all
INSTALL_ENCRYPTION_KEY=IBMTivoliMonitoringEncryptionKey
SEED_TEMS_SUPPORTS=true
MS_CMS_NAME=TEMS
DEFAULT_DISTRIBUTION_LIST=NEW
```

- Tivoli Enterprise Monitoring Server: modify the ms_silent_config.txt file with the following information:

```
HOSTNAME=itmsrv1
NETWORKPROTOCOL=ip.pipe
SECURITY=YES
```

- Tivoli Enterprise Portal Server: modify the cq_silent_config.txt file with the following information:

```
CMSCONNECT=YES
HOSTNAME=itmsrv1
NETWORKPROTOCOL=ip.pipe
DB2INSTANCE=db2inst1
DB2ID=itmuser
DB2PW=passw0rd
WAREHOUSEID=itmuser
WAREHOUSEDB=WAREHOUS
WAREHOUSEPW=passw0rd
ADMINISTRATORID=db2inst1
ADMINISTRATORPW=passw0rd
```

- Summarization and Pruning Agent: modify the sy_silent_config.txt file with the following information:

```
CMSCONNECT=YES
HOSTNAME=itmsrv1
NETWORKPROTOCOL=ip.pipe
KSY_WAREHOUSE_TYPE=DB2
KSY_WAREHOUSE_JARS=/opt/ibm/db2/v10.1/java/db2jcc.jar,/opt/ibm/db2/v10.1/java/
db2jcc_license_cu.jar
KSY_DB2_JDBCURL=jdbc:db2://db2srv1:50001/WAREHOUS
KSY_DB2_JDBCDRIVER=com.ibm.db2.jcc.DB2Driver
KSY_WAREHOUSE_USER=itmuser
KSY_DB_COMPRESSION=N
KSY_TIMEZONE_IND=AGENT
KSY_START_OF_WEEK_DAY=0
KSY_SHIFTS_ENABLED=N
KSY_SHIFT1_HOURS=0,1,2,3,4,5,6,7,8,18,19,20,21,22,23
KSY_SHIFT2_HOURS=9,10,11,12,13,14,15,16,17
KSY_VACATIONS_ENABLED=N
KSY_WEEKENDS_AS_VACATIONS=N
KSY_VACATION_DAYS=
SY_MAX_ROWS_PER_TRANSACTION=1000
KSY_FIXED_SCHEDULE=Y
KSY_EVERY_N_DAYS=1
KSY_HOUR_TO_RUN=2
KSY_HOUR_AM_PM=AM
KSY_MINUTE_TO_RUN=0
KSY_EVERY_N_MINS=60
KSY_BATCH_MODE=0
KSY_CNP_SERVER_HOST=localhost
KSY_CNP_SERVER_PORT=1920
KSY_HOUR_AGE_UNITS=1
KSY_DAY_AGE_UNITS=0
```

```
KSY_MAX_WORKER_THREADS=2
KSY_CACHE_MINS=10
```

- Warehouse Agent: modify the `hd_silent_config.txt` file with the following information:

```
CMSCONNECT=YES
HOSTNAME=itmsrv1
NETWORKPROTOCOL=ip.pipe
KHD_DBMS=DB2
KHD_WAREHOUSE_JARS=/opt/ibm/db2/v10.1/java/db2jcc.jar,/opt/ibm/db2/v10.1/java/
db2jcc_license_cu.jar,
/opt/ibm/db2/v10.1/java/db2jcc4.jar,/opt/ibm/db2/v10.1/java/db2policy.jar
KHD_DB2_JDBCURL=jdbc:db2://nc045061:50001/WAREHOUS
KHD_DB2_JDBCDRIVER=com.ibm.db2.jcc.DB2Driver
KHD_WAREHOUSE_USER=itmuser
KHD_WAREHOUSE_PASSWORD=passw0rd
KHD_BATCH_USE=true
KHD_DB_COMPRESSION=false
KHD_SERVER_Z_COMPRESSION_ENABLE=false
KHD_SERVER_DIST_COMPRESSION_ENABLE=true
```

Creating a warehouse database

IBM Tivoli Monitoring supports a remote data warehouse through aliases in a local DB2 server.

About this task

For more information about creating a warehouse database, see Creating the Tivoli Data Warehouse database.

Procedure

1. Create a warehouse database on a remote server.
2. Create a DB2 user on a remote server. Grant the user administrative rights to the database.
3. On local DB2 on which IBM Tivoli Monitoring is installed, catalog a remote data warehouse.

Monitoring Agent for Linux

To monitor the entire IBM Cloud Orchestrator environment, install Monitoring Agent on each Linux computer and configure it with the host name of your Tivoli Enterprise Monitoring Server.

For more information about installing the operating system agents, see Installing monitoring agents.

The operating system agents are delivered with the following package:

- CIQ3QML - IBM Tivoli Monitoring V6.3.0.2 Agents, Multiplatform, Multilingual

If you want to use silent mode to install and configure the agents, you can use the following files:

- `silent_install.txt`. You can use this file without any changes.
- `lz_silent_config.txt`. Modify the file with the following information:

```
        CMSCONNECT=YES
        HOSTNAME=itmsrv1
        NETWORKPROTOCOL=ip.pipe
```

If you want the agents to report to your main Tivoli Enterprise Monitoring Server, configure each one of them with such a configuration file.

Monitoring Agent for Kernel-based virtual machines

The KVM agent is used to monitor the Region Server and it must be installed with other ITM 6.3.0.2 components. The agent requires the `libvirt` library that is used to connect with the monitored KVM hypervisor.

For more information about installing and configuring the agent, see Linux Kernel-based virtual machines agent.

To properly configure the agent, you must provide parameters that describe the hypervisor.

You must add the RSA public keys of host on which the KVM agent is deployed to the hypervisor to enable communication through the SSH protocol. The protocol is configured and enabled for the Kernel services that are created by the Region Server, but you must enable it between the Region Server and the computer on which IBM Tivoli Monitoring is installed.

For more information about configuring the SSH protocol, see SSH protocol.

The KVM agent requires the following packages:

- CIQ4HEN - IBM Tivoli Monitoring for Virtual Environments V7.2.0.2 VMware VI, KVM, NetApp Storage and NMA Agents and Support Files, Windows and Linux, English, Multiplatform
- CIQ4JML - IBM Tivoli Monitoring for Virtual Environments V7.2.0.2 Agent Language Pack, Multiplatform, Multilingual

If you want to use silent mode for installing and configuring the operating system agents, you can use the following files:

- Modify the `silent_install.txt` file with the following information:

```
INSTALL_PRODUCT=v1
INSTALL_PRODUCT_TMS=all
INSTALL_PRODUCT_TPS=all
INSTALL_PRODUCT_TPW=all
INSTALL_ENCRYPTION_KEY=IBMTivoliMonitoringEncryptionKey
SEED_TEMS_SUPPORTS=true
MS_CMS_NAME=TEMS
DEFAULT_DISTRIBUTION_LIST=NEW
```

- Modify the `v1_silent_config.txt` with the following information:

```
CMSCONNECT=YES
HOSTNAME=itmsrv1
NETWORKPROTOCOL=ip.pipe
INSTANCE=RegionServer
DATA_PROVIDER.KV1_LOG_FILE_MAX_COUNT=10
DATA_PROVIDER.KV1_LOG_FILE_MAX_SIZE=5190
DATA_PROVIDER.KV1_LOG_LEVEL=INFO
HOST_ADDRESS.RegionServer=<host name of the region server visible by itmsrv1>
USERNAME.RegionServer=root
PROTOCOL.RegionServer=ssh
PORT.RegionServer=22
CONNECTION_MODE.RegionServer=system
```

OpenStack hypervisors

With the default configuration, you can monitor the Region Server as a KVM hypervisor. To do so, you can use Monitoring Agent for Kernel-based virtual machines from Tivoli Monitoring for Virtual Environments.

OpenStack can use different hypervisors, like KVM or VMware. To monitor different KVM hypervisors, you can use the installed agent and simply add a new instance in the agent configuration.

To monitor a VMware hypervisor, you must install and configure Monitoring Agent for VMware, which is also included in Tivoli Monitoring for Virtual Environments. Then, you can add an instance of configuration every time a new hypervisor is added to OpenStack.

For more information about VMware Agent, see VMware VI User's Guide.

Chapter 13. Reporting

IBM Cloud Orchestrator provides a diverse set of reports that provide specific data you can use for planning purposes.

Tivoli Common Reporting

Tivoli Common Reporting is provided for reporting of monitoring, metering, and billing.

- For information about Tivoli Common Reporting, including installation, see the Jazz for Service Management information center.
- For information about using Tivoli Common Reporting for metering and billing, see the *Administering reports in the metering and billing section of SCCM User's Guide*.
- For information about using Tivoli Common Reporting in IBM Tivoli Monitoring, see the Tivoli Common Reporting in the IBM Tivoli Monitoring information center.

Tip: For information about how to use Tivoli Common Reporting to define users and groups, and set up your system administrator authority, view the "Setting up basic security for your reporting environment" video tutorial on YouTube.

For information about how to use Tivoli Common Reporting to restrict user access to specific reports, and limit certain reporting functions, view the "Restricting user access to specific reports" video tutorial on YouTube.

Chapter 14. Reference

The following topics provide reference information for IBM Cloud Orchestrator.

REST API reference

The representational state transfer (REST) application programming interface (API) is provided by IBM Cloud Orchestrator.

Before you begin

Each product exposes a REST API as there are no special configuration settings to enable or disable this interface. The IBM Cloud Orchestrator REST API is available on the same IP address or host name that is used to access the product GUI and command-line interface. Unlike the GUI, the REST API is only supported over the HTTPS protocol on port 443. The product uses a self-signed certificate for its SSL sessions. The same certificate is used for GUI, command-line interface and REST API sessions. You must configure your HTTPS client to either accept or ignore this certificate during the SSL handshake. You must use an HTTPS client that allows you to set the HTTP headers for each request. This is because there are multiple headers that are required for authentication, authorization, and content negotiation.

To comply with stricter security requirements, IBM Cloud Orchestrator enforces the use of the fully qualified domain name (FQDN) to call the user interfaces. You must use the FQDN to implement single sign-on. The FQDN is also necessary for all HTTP POST and PUT operations, which are used to submit all forms in the user interface, including the login credentials. In emergency cases only: if the FQDN cannot be used, you can disable the security check by removing the `cookie_domain` entry from the `/opt/ibm/ico/wlp/usr/servers/scui/etc/config.json` file.

Note: The `/opt/ibm/ico/wlp/usr/servers/scui` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

When generating HTTP requests to the IBM Cloud Orchestrator REST API, pay special attention to the following headers:

Accept
With a few exceptions, the REST API generates JSON-encoded data in its responses. Include an *"Accept: application/json"* header in your request to indicate the ability of your client to handle JSON responses.

Accept-Language
Use this header on your HTTP request to specify which language or locale must be used by the product when generating the response data. You can specify any of the languages that are supported by the product.

Authentication
The REST API only supports HTTP basic authentication. After successfully authenticating, the server will return two cookies that are named *zsessionid* and *SimpleToken* that must be included with subsequent HTTP requests that are part of the same session. The same user IDs and passwords that are used to access the GUI and the command-line interface are used to access the REST API. The authorization of a user to perform actions on the product is independent of the interface (GUI, command-line interface or REST API) used to request the actions.

Content-Type
All the content included in an HTTP request body sent to the product must be JSON encoded. You must include a *"Content-Type: application/json"* header to indicate this for each request that includes any data.

projectName

When not using the default project, the HTTP request must include in the header `"projectName:<yourProjectName>"` to let the user be authenticated in the *<yourProjectName>* project.

The REST API is only supports the sending and receiving of UTF-8 encoded data. Ensure that your HTTP client is set to encode and decode character data, including JSON data. All responses of REST requests in JSON format are encoded in UTF-8.

Note: Key-value pairs that are only used by user interface clients are optional.

REST API frameworks

This topic describes the REST API frameworks that are used in an IBM Cloud Orchestrator environment for REST calls (client and server).

The following REST API frameworks are used:

- `org.apache.wink 1.1.3`: used by IBM Cloud Orchestrator.
- `javax.net.ssl.HttpsURLConnection`: used by the `genericREST` implemented by IBM Cloud Orchestrator that makes REST calls from Business Process Manager to IBM Cloud Orchestrator.
- `org.apache.http`: used by Business Process Manager to connect to OpenStack.

For customized REST calls that are implemented outside of IBM Cloud Orchestrator, make sure that the compatible REST API framework is used. For example, if you implement a script that makes REST calls against IBM Cloud Orchestrator, you must use the `org.apache.wink 1.1.3` framework or any other compatible REST API framework.

Managing floating IP addresses

Managing floating IP addresses using REST APIs.

Management of floating IP addresses is required when the floating IP address needs to assigned to a deployed virtual machine where NSX plug-in is installed. See "[Optional] Installing the NSX plug-in" on page 24.

- Use this REST API to get a list of floating IP addresses.

Table 23. Get list of all floating IP addresses

HTTP method	GET
Endpoint	http://<ICM_HOST>:9696/v2.0/floatingips
Error response code	– 401

- Use this REST API to delete a specified floating IP address.

Table 24. Delete floating IP addresses

HTTP method	DELETE
Endpoint	http://<ICM_HOST>:9696/v2.0/floatingips/floatingIp_ID
Error response code	– 401 – 404

- Use this REST API to create a floating IP address.

Table 25. Create a floating IP address, method 1

HTTP method	POST

Table 25. Create a floating IP address, method 1 (continued)

Request body	`{`
	`"floatingip":`
	`{ "floating_network_id": "<Network Id>",`
	`"port_id": "<Port Id>",`
	`"subnet_id": "<Subnet Id>",`
	`"fixed_ip_address": "<IP>",`
	`"floating_ip_address": "<Floating IP>",`
	`"description": "<Custom Description>"`
	`}`
	`}`

Endpoint	http://<ICM_HOST>:9696/v2.0/floatingips
Normal response code	– 201
Error response code	– 400 – 401 – 404 – 409

Table 26. Create a floating IP address, method 2

HTTP method	POST
Endpoint	http://<ICM_HOST>:9696/v2.0/floatingips

Request body	`{`
	`"floatingip":`
	`{`
	`"floating_network_id": "<Network Id>",`
	`"tenant_id": "<Tenant Id>"`
	`}`
	`}`

Normal response code	– 201
Error response code	– 400 – 401 – 404 – 409

- Use this REST API to associate a particular floating IP address.

Table 27. Assign floating IP addresses

HTTP method	POST

Request body	`{`
	`"addFloatingIp": {`
	`"address": "<IP>" }`
	`}`

Table 27. Assign floating IP addresses (continued)

Endpoint	http://<ICM_HOST>:9696/v2/{tenant_id}/servers/{server_id}/action
Normal response code	– 202
Error response code	– 400: badRequest – 401: unauthorized – 403: forbidden – 404: itemNotFound – 409: conflict

- Use this REST API to disassociate a floating IP from a certain instance.

Table 28. Dissociate a certain IP from floating IP addresses

HTTP method	POST

Request body	`{`
	`        "disassociate": {`
	`        "instance_id": "<Instance Id>"}`
	`}`

Endpoint	http://<ICM_HOST>:9696/v2.0/floating-ips/{ipid}/action
Normal response code	– 202

Related tasks

"[Optional] Installing the NSX plug-in" on page 24
The NSX plug-in can be installed and configured with IBM Cloud Manager with OpenStack V4.3 Fix Pack 13.

Business Process Manager Invoker REST API

You can use this set of Invoker REST APIs to retrieve information about artifacts that are available in Business Process Manager without accessing them directly.

For detailed information about Business Process Manager REST API, see the Business Process Manager information center.

Retrieve available BPM Business Processes

Use these APIs to retrieve a list of all the BPM Business Processes that are available for the implementation of self-service offerings or orchestration actions.

List all Business Process Manager Business Processes

Use this REST call to retrieve a list of all the Business Process Manager Business Processes that are available for the implementation of self-service offerings or orchestration actions.

Available HTTP method

Table 29. Get list of all Business Process Manager Business Processes

HTTP method	GET
URL pattern	`https://hostname/kernel/bpm/runbook/`

Table 29. Get list of all Business Process Manager Business Processes (continued)

Response	A list of available Business Process Manager Business Processes is returned. If no Business Process Manager Business Processes are available, an empty list is returned with HTTP Code 200. The returned list has the following parameters:
	`{ id:`
	`displayName:`
	`processAppId:`
	`}`

Return values	• 200 - No availableBusiness Process Manager Business Processes found

An entry has the following attributes:

- **`id`** - the unique ID of theBusiness Process Manager Business Process as used in the underlying execution engine. Communication with the underlying engine, for example, to start aBusiness Process Manager Business Process, is typically done by using this ID.
- **`displayName`** - a human-readable name of the Business Process Manager Business Processes, typically used for display in the UI.
- **`processAppId`** - an identifier of a collection of Business Process Manager Business Processes to which the Business Process belongs.

The example shows a response to the following request:

```
GET /kernel/bpm/runbook/
```

```
[
{
  "id": "25.916d4552-9cf4-40c3-89fe-7f7bc43b2435",
    "displayName": "Create Business Object Test"
    "processAppId": "2066.5d35fbc5-6949-4971-8e06-83ca4c3cc760",
  },
{
    "id": "25.2951bb80-e9b2-457a-9097-4443886d1dd5",
    "displayName":"SCO_Process"
    "processAppId": "2066.5d35fbc5-6949-4971-8e06-83ca4c3cc760",
}
]
```

Get entries for a specific Business Process Manager Business Process

Use this REST call to retrieve information about a Business Process Manager Business Process with an indicated ID.

Available HTTP method

Table 30. Get information about a specific Business Process Manager Business Process

HTTP method	GET
URL pattern	`https://hostname/kernel/bpm/runbook/runbook_id`

Response	The following parameters of the Business Process Manager Business Process are retrieved:
	`{ id:`
	`displayName:`
	`processAppId:`
	`}`

Return values	• 404 - TheBusiness Process Manager Business Process not found

This example shows the response to the following request GET `/kernel/bpm/runbook/25.916d4552-9cf4-40c3-89fe-7f7bc43b2435`:

```
{
  "id": "25.916d4552-9cf4-40c3-89fe-7f7bc43b2435",
  "displayName": "Create Business Object Test"
  "processAppId": "2066.5d35fbc5-6949-4971-8e06-83ca4c3cc760",
}
```

.

Retrieve available human services

Use these APIs to retrieve information about the human services that are available for the implementation of self-service offerings and orchestration actions in IBM Cloud Orchestrator.

List all human services

Use this REST API to retrieve a list of all the human services that are available for the implementation of self-service offerings and orchestration actions in IBM Cloud Orchestrator.

Available HTTP method

Table 31. Get list of all human services

HTTP method	GET
URL pattern	`https://hostname/kernel/bpm/humanService`

Response	A list of available human services is returned. If no human services are available, an empty list is returned with HTTP Code 200. The returned list has the following parameters:
	`{ id:`
	`displayName:`
	`runUrl:`
	`}`

Return values	• 200 - No available human services found

An entry has the following attributes:

- **`id`** - the unique ID of the human service as used in the underlying execution engine.
- **`displayName`** - a human-readable name of the human service, which is typically used for display in the UI.
- **`runUrl`** - the URL at which the human service can be started.

The following listing shows an example response that can be retrieved through the REST API:

```
[
  {
    "id": "1.6b0f42d8-0d65-4073-83bd-a3c7cb046f32",
    "displayName": "AddUserToVM",
    "runUrl": "http://xvm192:9080/teamworks/executeServiceByName?
               processApp=SCO_P1&serviceName=AddUserToVM"
  },
  {
    "id": "1.10027723-6b52-46f8-9105-535c75a09970",
    "displayName": "Show_Topology_Data",
    "runUrl": "http://xvm192:9080/teamworks/executeServiceByName?
               processApp=SCO_P1&serviceName=Show_Topology_Data"
  }
]
```

.

Get entries for a specific human service

Use this REST call to retrieve information about a human service with an indicated ID.

Available HTTP method

Table 32. Get information about a specific human service

HTTP method	GET
URL pattern	`https://hostname/kernel/bpm/humanService/<human_service_id>`

Response	The following parameters of the human service are retrieved:
	`{ id:`
	`displayName:`
	`processAppId:`
	`}`

Return values	• 404 - The human service not found

This example shows the response to the following request:

```
GET /kernel/bpm/humanService/1.6b0f42d8-0d65-4073-83bd-a3c7cb046f32
```

```
{
  "id": "1.6b0f42d8-0d65-4073-83bd-a3c7cb046f32",
  "displayName": "AddUserToVM",
  "runUrl": "http://xvm192:9080/teamworks/executeServiceByName?
             processApp=SCO_P1&serviceName=AddUserToVM" }
```

.

Retrieve the Inbox

Use these APIs to retrieve information about the contents of the Inbox.

For detailed information about Business Process Manager task REST API, see the Business Process Manager information center.

List all Inbox items

Use this REST API to retrieve a list of all the items that are contained in the Inbox.

Available HTTP method

Table 33. Get list of all Inbox items

HTTP method	GET
URL pattern	`https://hostname/kernel/bpm/task`

Table 33. Get list of all Inbox items (continued)

Response	A list of Inbox items is returned. If Inbox is empty, an empty list is returned with HTTP Code 200. The returned list has the following parameters:
	`{`
	`id:`
	`assignedTo:`
	`assignedToType:`
	`displayName:`
	`domain:`
	`operationContextId:`
	`project:`
	`relatedTo:`
	`requester:`
	`serviceInstanceId:`
	`taskDueDate`
	`taskOverdue:`
	`taskPriority:`
	`taskStatus:`
	`taskType:`
	`time:`
	`}`

Return values	• 200 - No available pending human activities found

An entry has the following attributes:

- **id** - the unique ID of the pending human activity as used in the underlying execution engine.
- **assignedTo** - the human-readable display name of the group or user to which this request is assigned.
- **assignedToType**:
  - **group** - if the task is still unclaimed.
  - **user** - if the task is claimed.
- **displayName** - a human-readable name of the pending human activity, which is typically used for display in the UI.
- **domain** - the domain of the user who requested this process.
- **operationContextId** - the ID of the operation context or request that is associated to this inbox
- **project** - the project on behalf of which this process was requested.
- **relatedTo** - the name of the process to which the pending human activity belongs.
- **requester** - the requester of the process to which the pending activity belongs.
- **serviceInstanceId**- the ID of the associated virtual system instance, if the approval is part of a triggered event or user action.
- **taskDueDate** - due date of the pending human activity.
- **taskOverdue**:
  - **true** - if the current date is later than the due date of the pending human activity.
  - **false** - otherwise.
- **taskPriority** - the priority of the task as used in the underlying execution engine.
- **taskStatus** - the status of the pending human activity as used in the underlying execution engine.
- **taskType** - the type of human activity:
  - **approval** - for an approval request.
  - **general** - for a general human task.
- **time** - the time at which the process was triggered.

The following listing shows an example response with one pending task:

```
[
   {
      "relatedTo": "Sample_DeleteInstanceApproval",
      "taskStatus": "Received",
      "taskPriority": "Normal",
      "taskOverdue": "true",
      "id": "8",
      "requester": "admin",
      "taskDueDate": "2013-08-19T17:27:26Z",
      "time": "2013-08-19T16:27:26Z",
      "displayName": "Delete Instance Approval: Sample1",
      "taskType": "approval",
      "assignedToType": "group",
      "operationContextId": "1007",
      "domain": "Default",
      "serviceInstanceId": null,
      "assignedTo": "All Users",
      "project": "admin"
   }
]
```

.

Get entries for a specific Inbox item

Use this REST call to retrieve information about an Inbox item with an indicated ID

Available HTTP method

Table 34. Get information about a specific Inbox item

HTTP method	GET
URL pattern	`https://hostname/kernel/bpm/task/`*`<task id>`*

Response	The following parameters of an Inbox item are retrieved:
	`{`
	`overdue:`
	`dueDate:`
	`status:`
	`priority:`
	`id:`
	`displayName:`
	`requester:`
	`time:`
	`type:`
	`assignedToType:`
	`assignedTo:`
	`operationContextId:`
	`domain:`
	`project:`
	`serviceInstanceId:`
	`parameters:[`
	`    {`
	`      "Operation Type":`
	`      "Operation Context ID":`
	`      "Virtual System Pattern ID"`
	`      "Virtual System Name"`
	`      "Virtual System ID"`
	`    }*`
	`  ],`
	`  }`

Return values	• 404 - The Inbox item not found

An entry has the following attributes:

- **`overdue`**:
  - **`true`** - if the current date is later than the due date of the pending human activity.
  - **`false`** - otherwise.

- **duedate** - due date of the pending human activity.
- **status** - the status of the pending human activity as used in the underlying execution engine.
- **priority** - the priority of the task as used in the underlying execution engine.
- **id** - the unique ID of the pending human activity as used in the underlying execution engine.
- **displayName** - a human-readable name of the pending human activity, which is typically used for display in the UI.
- **requester** - the requester of the process to which the pending activity belongs.
- **time** - the time at which the process was triggered.
- **type** - the type of human activity:
  - **approval** - for an approval request
  - **general** - for a general task
- **assignedToType**:
  - **group** - if the task is still unclaimed.
  - **user** - if the task is claimed.
- **assignedTo** - the human-readable display name of the group or user to which this request is assigned.
- **operationContextId** - the ID of the operation context or request that is associated to this inbox
- **domain** - the domain of the user who requested this process.
- **project** - the project on behalf of which this process was requested.
- **serviceInstanceId**- the ID of the associated virtual system instance, if the approval is part of a triggered event or user action.
- **parameters** - a name-value list that holds more information about the task.

  Note: For a general task, the parameters array is typically empty because an additional UI of the underlying process engine is provided which can be started through a URL.

This example shows the response to a request:

```
{
  "overdue": "true",
  "dueDate": "2013-08-19T17:27:26Z",
  "status": "Pending",
  "priority": "Normal",
  "type": "approval",
  "id": "8",
  "requester": "admin",
  "time": "2013-08-19T16:27:26Z",
  "displayName": "Delete Instance Approval: Sample1"
  "assignedToType": "group",
  "operationContextId": "1007",
  "domain": "Default",
  "serviceInstanceId": null,
  "assignedTo": "All Users",
  "project": "admin"
  "parameters": [
     { "Operation Type": "Delete Instance" },
     { "Operation Context ID": "\/kernel\/tasks\/505dcc6d-7a97-4563-8f7b-6569ebc9e94c" },
     { "Virtual System Pattern ID": "\/resources\/patterns\/1" },
     { "Virtual System Name": "Sample1" },
     { "Virtual System ID": "\/resources\/virtualSystems\/2"}
  ],
}
```

.

Core Services REST API

Core Services REST API overview

IBM Cloud Orchestrator uses REST APIs to allow easy, lightweight communication between components and integration with external systems.

Linked Resources versus Collection Resources

REST API implements a linked data concept where relations between resources are provided with the response as metadata. A linked resource is the most basic entity in a response. It represents a resource itself as well as any links to other resources. The following code is a structure of a linked resource:

```
{
"href": "https://<ico_server_fqdn>:443/orchestrator/v2/...",
"item": { ... },
"link_1": {
           "href": "https://<ico_server_fqdn>:443/orchestrator/v2/..."
          },
"link_2": {
           "href": "https://<ico_server_fqdn>:443/orchestrator/v2/..."
          }
}
```

Each linked resource has at least one link to itself, the first `href` property. An item property follows with the actual resource representation. There can also be more links to other resources. A collection resource is a collection of linked resources. In addition to the basic properties of a linked resource, a collection resource also features specific properties for pagination. The following code displays the structure of a collection resource:

```
{
"href": "https://<ico_server_fqdn>:443/orchestrator/v2/collection",
  "start": 10,
  "limit": 10,
  "total": 49,
  "first": {
    "href": "https://<ico_server_fqdn>:443/orchestrator/v2/collection/?_limit=10&_start=0"
  },
  "previous": {
     "href": "https://<ico_server_fqdn>:443/orchestrator/v2/collection/?_limit=10&_start=0"
  },
  "next": {
     "href": "https://<ico_server_fqdn>:443/orchestrator/v2/collection/?_limit=10&_start=20"
  },
  "last": {
     "href": "https://host:9443/orchestrator/v2/collection/?_limit=10&_start=39"
  },
  "items": [ ... ]
```

The start, limit, and total properties enable you to display the correct number of pages in a UI. The number of pages is the total size that is divided by the page size. You can also choose to use the provided first, next, and other links and can call them directly from a UI to navigate the collection easily.

HTTP Status Codes

The following codes apply:

Status Code	Request	Description
200 OK	GET /resource, GET /collection, PUT /resource	General status if the request went OK, no resources were created.
201 Created	POST /collection	New resource was created.
202 Accepted	POST /collection/{id}/launch	A launch request was accepted.
204 No Content	DELETE /resource	A resource was deleted.

Status Code	Request	Description
400 Bad Request	PUT /resource, POST /collection	Request payload was not complete or badly formatted.
401 Unauthorized	Any	Session has expired or no SimpleToken was provided.
403 Forbidden	Any	Session is valid but user was not authorized for the requested operation.
404 Not Found	Any	Requested resource path was not found.
405 Method Not Allowed	PUT /collection, DELETE / collection	Tried to update or delete a collection resource.
406 Not Acceptable	Any	Invalid Accept header. Only application/json is allowed by default.
409 Conflict	POST /collection	A resource with the same identifier already exists.
415 Unsupported Media Type	POST /collection, PUT /resource	Requested invalid content type. Only application/json is allowed by default.
500 Internal Server Error	Any	An internal error occurred. Log files must be checked. The REST API might provide more hints in the response body.

HTTP Media Types

By default, all REST APIs consume and produce application/json as their media type. Other types such as text/xml are not supported unless stated otherwise.

Pagination, Filtering, Sorting and Searching

URL parameters that are used to control the output of a REST API are prefixed with an underscore _ to distinguish them from queries on resource properties. This avoids confusion between

`/apples?sort=goldendelicious` (gives all apples of the "goldendelicious" sort)

and

`/apples?sort=ascending&sortby=id` (sorts apples by id in ascending order).

The following table describes URL keywords controlling REST API behavior:

Keyword	Meaning
`_start=n`	Start pagination at element n.
`_limit=n`	Set page size to n elements per page. There is an upper boundary of 100 items and a lower boundary of 5 items. If omitted, 10 is the default.
`_sortby=abc`	Sort result set by resource attribute abc. If omitted, id (or the respective identifier of the resource) is the default.

Keyword	Meaning
`_sort=asc | desc`	Sort in ascending or descending order. If omitted, as is the default.
`_search=abc`	Do a case-insensitive, full-text search for abc in the searchable parts of a resource. Depends on REST API implementation, usually name and description are searchable.
`property=value`	If no underscore prefix was given, filter for resources with properties containing value. Multiple values can be passed as a separate `property=value` pair.

Examples

- `/orchestrator/v2/categories` - Returns service catalog categories starting at index 0 with a limit of 10, sorted by ID in ascending order.
- `/orchestrator/v2/categories?_start=10` - Returns service catalog categories starting at index 10 with a limit of 10, sorted by ID in ascending order.
- `/orchestrator/v2/categories?_start=10&_limit=20` - Returns service catalog categories starting at index 10 with a limit of 20, sorted by ID in ascending order.
- `/orchestrator/v2/categories?_sortby=name` - Returns service catalog categories starting at index 0 with a limit of 10, sorted by name in ascending order.
- `/orchestrator/v2/categories?_sortby=id&_search=virtual` - Returns service catalog categories containing the word "virtual" starting at index 0 with a limit of 10, sorted by ID in ascending order.
- `/orchestrator/v2/categories?name=OpenStack` - Returns service catalog categories whose name is OpenStack starting at index 0 with a limit of 10.
- `/orchestrator/v2/categories?id=123&id=456` - Returns service catalog categories with the ids 123 and 456 starting at index 0 with a limit of 10.

Offering REST API V2

The following topics cover categories, offering attributes and offering instances of the offering REST API V2.

Categories

Json Formats

Category Request:

```
{

    "isbuiltin": 0,
    "icon": "Web Machine Category Icon:ge100_webcatalog_24",
    "name": "Manage Virtual Machines",
    "description": "Deploy, start, stop and virtual machines based on a single image."
}
```

Category Response:

```
{
  "href": "https://<ico_server_fqdn>:443/orchestrator/v2/categories/4711",
  "item": {

    "id": 4711,

    "isbuiltin": 0,
    "icon": "Web Machine Category Icon:ge100_webcatalog_24",
    "name": "Manage Virtual Machines",
    "description": "Deploy, start, stop and virtual machines based on a single image."
  }

}
```

Categories Response

```
{
  "href": "https://<ico_server_fqdn>:443/orchestrator/v2/categories/",
  "start": 0,
  "limit": 10,
  "total": 9,
  "first": {
    "href": "https://<ico_server_fqdn>:443/orchestrator/v2/categories/?_limit=10&_start=0"
   },
   "previous": null,
   "next": null,
   "last": {
     "href": "https://<ico_server_fqdn>:443/orchestrator/v2/categories/?_limit=10&_start=0"
   },
   "items": [ Category Response, ....,  Category Response ]
  }
```

Attribute	Description	Type	Mandatory	Generated	Comment
id	category id	Number	no	yes	automatically assigned when a new category gets created
icon	icon name	String	no	no	name of the icon that is displayed with the category in the UI
name	category name	String	yes	no	name of the category
description	category description	String	yes	no	description of the category
isbuiltin	built in	Number	no		

GET: Lists offering categories

URL pattern
`/orchestrator/v2/categories`

Accepts
*

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

500 internal server error

Response
Categories Response

Search Attributes
name, description

Filter Attributes
all

Authorization
no authorization needed

POST: Creates a offering category

URL pattern
`/orchestrator/v2/categories`

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
201 created

Error Response Codes
400 bad request if bad JSON was passed or mandatory attributes were missing

401 unauthorized

500 internal server error

Request
Category Request

Response
Category Response

Authorization
role:"admin"

GET: Get category

URL pattern
`/orchestrator/v2/categories/{id}`

Accepts
*

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Category Response

Authorization
no authorization

PUT: Update category

URL pattern
`/orchestrator/v2/categories/{id}`

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes

400 bad request if bad JSON was passed

401 unauthorized

404 not found

500 internal server error

Request

```
{

...
   "name": "Manage Virtual Image",
   "description": "Deploy, start, stop"

...

}
```

Response

Category Response

Authorization

role: "admin"

DELETE: Delete category

URL pattern

`/orchestrator/v2/categories/{id}`

Accepts

*

Normal Response Codes

204 no content

Error Response Codes

401 unauthorized

404 not found

500 internal server error

Authorization

role: "admin"

Offering attributes

Attributes for the offering are displayed in this topic.

Attributes

Attribute	Description	Type	Mandatory	Generated	Comment
id	service id	Number	no	yes	automatically assigned when a new offering is created
icon	icon name	String	no	no	name of the icon that is displayed with the category in the UI
name	offering name	String	yes	no	name of the offering
description	offering description	String	yes	no	description of the offering
category	offering	Number	no	no	category id of this offering

Attribute	Description	Type	Mandatory	Generated	Comment
implementation_type	type of process that gets started	String	no		defaults to "ibm_bpm_process" if not passed
process_app_id	Business Process Manager process application ID containing the linked process (process attribute)	String	yes	no	
process	Business Process Manager process implementing the offering or action	String	yes	no	
human_service_app_id	Business Process Manager process application ID containing the linked human service (`human_service` attribute)	String	no	no	
human_service	Human service implementing the User Interface for the offering or action	String	no	no	
ownerid				no	
operation_type				no	"offering", "singleInstanceAction" , "multiInstanceAction"
instancetype	type of instances the process is working on	String	no	no	name of instance provider
tags	List of service designer tags matching to the ones of the instance type	List of Strings	no	no	subset of tags of instance provider
acl	Access control list	List of ACL JSON	no	no	
acl/domain				no	
acl/project					
acl/role					
acl/use					
acl/modify					
acl/view					

Offering instances
A list of instances for the offering are described in this topic.

Instances

GET: Lists offerings

URL pattern
`/orchestrator/v2/offerings`

Accepts
*

Content-Type
`application/JSON`

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

500 internal server error

Response
Offerings Response

Search Attributes
Name, description

Filter Attributes
id, name, description, icon, human_service, human_service_app_id, priority, created, updated, process, process_app_id, owner_id, category, implementation_type, operation_type, instancetype

Authorization
role: admin or ACL with 'view' set to 'true' for given domain, project, and role

POST: Creates an offering

URL pattern
`/orchestrator/v2/offerings`

Accepts
`application/JSON`

Content-Type
`application/JSON`

Normal Response Codes
201 created

Error Response Codes
400 bad request if bad JSON was passed or mandatory attributes were missing

401 unauthorized

500 internal server error

Request
Offering Request

Response
Offering Response

Authorization
roles: "admin", "domain_admin"

POST: Creates an offering

URL pattern
`/orchestrator/v2/offerings`

Accepts
`application/JSON`

Content-Type
`application/JSON`

Normal Response Codes
201 created

Error Response Codes
400 bad request if bad JSON was passed or mandatory attributes were missing

401 unauthorized

500 internal server error

Request
Offering Request

Response
Offering Response

Authorization
roles: "admin", "domain_admin"

GET: Get an offering

URL pattern
`/orchestrator/v2/offerings/`*`{id}`*

Accepts
*

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Offering Response

Authorization
role: admin or ACL with 'view' set to 'true' for given domain, project, and role

PUT: Update an offering

URL pattern
`/orchestrator/v2/offerings/`*`{id}`*

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes
400 bad request if bad JSON was passed

401 unauthorized

404 not found

500 internal server error

Request
Offering Request (partial)

Response
Offering Response

Authorization
role: admin or ACL with 'modify' set to 'true' for given domain, project, and role

DELETE: Delete an offering

URL pattern
`/orchestrator/v2/offerings/{id}`

Accepts
*

Normal Response Codes
204 no content

Error Response Codes
401 unauthorized

500 internal server error

Authorization
role: admin or ACL with 'modify' set to 'true' for given domain, project, and role

POST: Execute an offering

URL pattern
`/orchestrator/v2/offerings/{id}/launch`

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
202 accepted

Error Response Codes
400 bad request if bad JSON was passed

401 unauthorized

404 not found

500 internal server error

Request
Go to Launching an offering via offering API

Response
TaskResponse

Authorization
role: admin or ACL with 'use' set to 'true' for given domain, project, and role

GET: Get ACL entries for a given offering

URL pattern
`/orchestrator/v2/offerings/{id}/acl`

Accepts
*

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
ACLs Response

Authorization
No authorization is needed but the result is restricted for the given domain, project, and role of the user

PUT: Update given acl for a given offering

URL pattern
`/orchestrator/v2/offerings/{id}/acl`

Accepts
`application/JSON`

Content-Type
`application/JSON`

Normal Response Codes
200

Error Response Codes
400 bad request of bas JSON was passed

401 unauthorized

404 not found

500 internal server error

Request
ACLs Request

Response
ACLS Response

Authorization
No authorization is needed but the given ACL is adjusted to the given domain, project, and role of the user

GET: Get input parameters for a given offering

URL pattern
`/orchestrator/v2/offerings/{id}/parameters`

Accepts
*

Content-Type
`application/JSON`

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Parameters Response

Authorization
No authorization is needed but the result is restricted for the given domain, project, and role of the user

GET: Get graphical representation of a given offering workflow

URL pattern
`/orchestrator/v2/offerings/{id}/graph`

Accepts
*

Content-Type
application/JSON

Normal Response Codes
200 OK

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
`image/jpeg` binary image representing this offering's workflow

Authorization
No authorization is needed but the result is restricted for the given domain, project, and role of the user

Launching an offering through Offering REST API

There are two ways of launching an offering. In both cases, you need to know the offering ID of the offering to launch. You can either use the user interface or pass the data directly with the initial request.

An offering can be retrieved by listing the offerings by using a GET request on the offerings:

```
GET https://<ico_server_fqdn>:443/orchestrator/v2/offerings
```

The ID attribute on each offering contains the offering ID.

1. The offering does not require any input data. In this case, you can start the offering by performing a POST request on the offering you want to launch.

   ```
   POST https://<ico_server_fqdn>:443/orchestrator/v2/offerings/<offering-id>/launch
   ```

2. The offering does require more input data. In this case, you must decide whether you want to use the IBM Cloud Orchestrator User Interface to gather that data and start the process, or if you provide the data as part of the launch request.

Using the User Interface

If you want to use the IBM Cloud Orchestrator User Interface you must perform the following steps:

1. Issue the call to initiate the offering by performing the post request:

   ```
   POST https://<ico_server_fqdn>:443/orchestrator/v2/offerings/<offering-id>/launch
   ```

   with the following header:

   ```
   "Content-Type" : "application/json"
   ```

   and POST body containing the string `{}` (for an empty JSON object).
2. In the JSON response of that request find the `redirect` attribute that contains the path to launch the IBM Cloud Orchestrator User Interface for the offering. For example:

   ```
   "redirect":"\/teamworks\/executeServiceByName?processApp=SCONOVA
   &serviceName=Deploy+Single+Virtual+Machine&tw.local.operationContextId=3059"
   ```

3. Launch the User Interface (which is the Human Service of the offering) with the URI from the previous step by running the following request (on one line):

```
GET https://<ico_server_fqdn>:443/teamworks/executeServiceByName?processApp=SCONOVA
&serviceName=Deploy+Single+Virtual+Machine&tw.local.operationContextId=3059
```

After the data is filled in, the business process of the offering is started automatically.

Pass the data directly with the initial request

Offerings started this way cannot have a human service configured. If the offering has a human service that is configured, create a new one by using the same process as the original offering, with no human service configured. Creating such an offering can be done in the IBM Cloud Orchestrator Self-Service Configuration User Interface. To start the offering, you must do the same POST request as in the other option for creating an offering:

```
POST https://<ico_server_fqdn>:443/orchestrator/v2/offerings/<offering-id>/launch
```

You must also provide a POST body in that request describing the `InputParameterObject` that is passed into the process. Pass the body a JSON document in this format:

```
{"parm":{"OperationParameter":
      "<variable type=\"Sample_BusinessObject\">
<field1 type=\"String\"><![CDATA[Hello]]><\/field1>
<field2 type=\"String\"><![CDATA[Phone]]><\/field2> <\/variable>"}}
```

The `OperationParameter` is the serialized form of the related Business Process Manager business object as returned by the Business Process Manager`tw.system.serializer.toXML(tw.local.inputParameterObject)`

The following methods are some ways of finding out more about the input parameter data type of the process you want to launch. If you need to inspect parameters of the offering workflow, the following methods can be performed:

1. Use REST interface for IBM Cloud Orchestrator to retrieve details of a registered offering:

   ```
   GET https://<ico_server_fqdn>:443/orchestrator/v2/offerings/<offering-id>
   ```

   In this data you can find:

   `process`
   The name of theBusiness Process Manager process that is bound to the offering.

   `process_app_id`
   The ID of the Business Process Manager application in which the process is defined.

   For details, refer to GET entries for a specific self-service offering.

2. Use REST Interface for Business Process Manager - related resources to get details about exposed items:

   ```
   GET https://<ico_server_fqdn>:443/rest/bpm/wle/v1/exposed
   ```

   Find the process with `display=process` and `processAppID=process_app_id`.

   ```
     {"status":"200", "data":{ "exposedItemsList":[ { "type":"process",
       "itemID":"25.8403dd37-e049-46f5-8952-b7a46f0d198f",
       "processAppID":"2066.931b0053-02bd-4f47-ac72-4eb527457383",
       "snapshotID":"2064.73dd1d1a-b533-46ef-ba79-c94cb3b0de87",
       "snapshotName":"version 2.8.1",
       "display":"HR Open New Position", "ID":"2015.204" }
   ```

   For more information, see REST interface for BPD-related resources - Exposed Items Resource in the IBM Business Process Manager Knowledge Center.

3. Use REST Interface for Business Process Manager-related resources to get details about the process model:

   ```
   GET  https://<ico_server_fqdn>:443/rest/bpm/wle/v1/processModel/
   {bpdId}?processAppId={string}&parts=dataModel
   ```

Use `parts=all` for more information about the process. Find the process with `itemID=process` and `processAppID=process_app_id`:

```
{ "status":"200", "data":{ …, "DataModel":{ …} } }
```

For more information, see REST interface for BPD-related resources - Process Model Resource - GET Method in the IBM Business Process Manager Knowledge Center.

Sample Output:

Getting the type of the `inputParameterObject`. Getting the type detailed information, here the parameters to that service are two string parameters that are named `field1` and `field2`.

```
{  "status" : "200",
  "data" : {
  "DataModel" : {
  "properties" : {  "message" : { "type" : "String", "isList" : false },
  "returnFromRest" : { "type" : "String", "isList" : false }
  },
  "inputs" : {
  "operationContext" : { "type" : "OperationContext", "isList" : false },
  "inputParameterObject" : { "type" : "Sample_BusinessObject", "isList" : false }
  },
  …
  "Sample_BusinessObject" : {
  "properties" : { "field1" : { "isList" : false, "type" : "String“ },
                   "field2" : { "isList" : false, "type" : "String“ }
  },
  “type" : "object",
  "ID" : "12.2c079fa7-89a0-426c-a3c1-079be08930ac",
  "isShared" : false
  },
```

Getting an example for the input parameter payload

To get an example of the input parameter payload, trigger a request through the IBM Cloud Orchestrator and query the input parameters of the request. It might either be still running or finished. You must retrieve the request ID and retrieve the input parameters for that request.

1. To get the request ID, select the triggered request in REQUESTS and note the request ID as listed in the address bar.
2. Using the REST call to retrieve details about the request, use:

   ```
   GET https://<ico_server_fqdn>:443/kernel/tasks/{request-id}
   ```

3. Within the response, search for Operation Parameter. The following is a sample excerpt:

   ```
   "OperationParameter" : "<variable type=\"MyRequest\"
    <vpmoNumber type=\"Integer\"><![CDATA[116560]]><\/vpmoNumber>
    <appId type=\"Integer\"><![CDATA[19073]]><\/appId>
    <attuidNo type=\"String\"><![CDATA[dw945f]]><\/attuidNo>
    <serverType type=\"NameValuePair\">
    <name type=\"String\"><![CDATA[Test]]<\/name>
    <value type=\"String\"><![CDATA[T]]&gt;<\/value>
    <\/serverType>
   ```

Getting information on a request

Once a request is started the IBM Cloud Orchestrator system can be queried for the actual status of that request. This is done by using the REST call:

```
GET https://<ico_server_fqdn>:443/kernel/tasks/{id}
```

The response contains information about the status of the request. For details on possible values, refer to GET entries for a specific task. Sample response (excerpt) from

```
REST GET https://<ico_server_fqdn>:443/kernel/tasks/{id}
```

:

```
{
  "updated_iso" : "2014-02-19T17:54:15+0100",
  "description_message" : "PROCESS_COMPLETE",
  "domain" : "Default",
  "created" : 1392828461580,
  "error" : { ... },
  "serviceInstance" : {
  "virtualMachines" : [{
  "memory" : 4096,
  "hypervisorid" : "\/resources\/hypervisors\/PM-1",
  "hostname" : "SC-192-168-0-103.RegionOne.example.com",
  ....
  }
  ],
  ....
  },
  "user" : "admin",
  "parm" : {
  "startPlanByPlugpointEventHandler" : "done",
  "CUSTOM_PARM1": "abc",
  "CUSTOM_PARM2": "xyz",
  "OperationParameter" : "<variable ...<\/variable>",
 "serviceInstanceId" : "282",
  "plan" : { ... },
  "processId" : "1356"
  },
  "created_iso" : "2014-02-19T17:47:41+0100",
  "status_localized" : "TASKSTATUS_COMPLETED",
  "error_message" : "BPM_PROCESS_COMPLETE",
  "status" : "COMPLETED",
  "eventTopic" :  "com\/ibm\/orchestrator\/serviceinstance\/plan\/ibm_bpm_process",
  "delayInSeconds" : 30,
  "project" : "admin",
  ...
  }
  }
```

Resource instances REST API

Instances returned for a specific type are collected using an instance provider. An instance provider is a Java class that can talk to a back-end to query instance information such as VMs, disks, users, networks and other cloud resources.

JSON Formats

Resource Type Request

```
{
    "name" : "myprovider",
    "displayname" : "My Provider",
    "description" : "This is my provider",
    "icon" : "Web Icon:glyphicons_266_flag",
    "provider" : "com.ibm.orchestrator.core.instance.providers.myprovider.MyProvider",
    "type" : "admin",
    "tags" : ["enabled", "disabled"],
    "detailsview" : {
        "application" : "SCOABC",
        "humanservice" : "Show My Provider Details"
    },
    "keyfields" : [{
            "instanceattribute" : "displayname",
            "header" : "Name"
        }, {
            "instanceattribute" : "description",
            "header" : "Description"
        }
    ]
}
```

Resource Type Response

```
{
    "name" : "myprovider",
    "displayname" : "My Provider",
    "description" : "This is my provider",
    "icon" : "Web Icon:glyphicons_266_flag",
    "provider" : "com.ibm.orchestrator.core.instance.providers.myprovider.MyProvider",
    "type" : "admin",
    "tags" : ["enabled", "disabled"],
    "detailsview" : {
        "application" : "SCOABC",
        "humanservice" : "Show My Provider Details"
    },
    "keyfields" : [{
            "instanceattribute" : "displayname",
            "header" : "Name"
        }, {
            "instanceattribute" : "description",
            "header" : "Description"
        }
    ]
}
```

Resource Types Response

```
{
       "href": "http://<hostname:port>/orchestrator/v2/instancetypes",
       "start": 0,
       "limit": 10,
       "total": 3,
       "first": "http://<hostname:port>/orchestrator/v2/instancetypes?_start=0&_limit=10",
       "previous": null,
       "next": null,
       "last": "http://<hostname:port>/orchestrator/v2/instancetypes?_start=0&_limit=10",
       "items":
       [
         Resource Type Response 1,...,Resource Type Response  n
       ]
    }
```

Resource Instance Request

```
{
"parm":
{
.... Instance type dependent paramter JSON object ....
},
"displayname": "mhtest1",
"detailsURL": "<hostname:port>/teamworks/executeServiceByName?processApp=<FOO>
&;serviceName=Show+Server+Details&tw.local.serverId
=b479108c-df8f-4462-be8b-f80af4a59d15&tw.local
.region=RegionOne&tw.local.user=<user>&tw.local.
domain=Default&tw.local.project=<project>",
"status": "ACTIVE",
"region": "RegionOne",
"icon": "Server Category Icon:ge100_servercatalog_24",
"openstackId": "b479108c-df8f-4462-be8b-f80af4a59d15",
"tags":
[
"active"
],
"id": "RegionOne--b479108c-df8f-4462-be8b-f80af4a59d15",
"updated": "2014-03-31T11:04:58Z",
"ipAddresses": "vmnet: 10.0.0.100",
"description": "mhtest1"
}
```

Resource Instance Response

```
{
"href": "<hostname:port>/orchestrator/v2/instancetypes/openstackvms
/instances/RegionOne--b479108c-df8f-4462-be8b-f80af4a59d15",
"created": "2014-03-31T11:04:18Z",
"parm":
{
.... Instance type dependent paramter JSON object ....
},
"displayname": "mhtest1",
"detailsURL": "<hostname:port>/teamworks/executeServiceByName?
processApp=<FOO>&serviceName=Show+Server+Details&tw
.local.serverId=b479108c-df8f-4462-be8b-f80af4a59d15&tw
.local.region=RegionOne&tw.local.user=<user>&tw
.local.domain=Default&tw.local.project=<project>",
"status": "ACTIVE",
"region": "RegionOne",
"icon": "Server Category Icon:ge100_servercatalog_24",
"openstackId": "b479108c-df8f-4462-be8b-f80af4a59d15",
"tags":
[
"active"
],
"id": "RegionOne--b479108c-df8f-4462-be8b-f80af4a59d15",
"updated": "2014-03-31T11:04:58Z",
"ipAddresses": "vmnet: 10.0.0.100",
"description": "mhtest1"
}
```

Resource Instances Response

```
{
"href": "<hostname:port>/orchestrator/v2/instancetypes/openstackvms/instances",
"start": 0,
"limit": 10,
"total": 2,
"first": "<hostname:port>/orchestrator/v2/instancetypes/
openstackvms/instances?_start=0&amp;_limit=10",
"previous": null,
"next": null,
"last": "<hostname:port>/orchestrator/v2/instancetypes
/openstackvms/instances?_start=0&amp;_limit=10",
"items":
[

Resource Instance Response 1, ..., Resource Instance Response n
]
}
```

Instances

GET : Lists all resource types

URL pattern
`/orchestrator/v2/instancetypes`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
500 internal server error

Response
Resource Type Response

Authorization
No authorization needed

POST: Create resource type

URL pattern
`/orchestrator/v2/instancetypes/`

Accepts
application/JSON

Content type
application/JSON

Normal Response Codes
201

Error Response Codes
401 unauthorized

409 conflict

500 internal server error

Request
Resource Type Request

Response
Resource Type Request

Authorization
role: admin

GET: Get one resource type

URL pattern
`/orchestrator/v2/instancetypes/{name}`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
404 not found

500 internal server error

Response
Resource Type Response

Authorization
No authorization needed

PUT: Update a resource type

URL pattern
`/orchestrator/v2/instancetypes/{name}`

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Request
Resource Type Request (partial)

Response
Resource Type Response

Authorization
role: admin

DELETE: Delete a resource type.

URL pattern
`/orchestrator/v2/instancetypes/{name}`

accepts
/

Content-Type
application/JSON

Normal Response Codes
204

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Authorization
role: admin

GET : List instances of a given type.

URL pattern
`/orchestrator/v2/instancetypes/{name}/instances`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Resource Instances Response

Authorization
Instance provider dependent. Generic Provider: Access Control Link with **view** set to **true** for the domain, project and role you are working on.

POST: Creates a instance of a given type.

URL pattern
`/orchestrator/v2/instancetypes/{name}/instances`

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
201 created

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Request
Resource Instance Request

Response
Resource Instance Response

Authorization
Instance provider dependent. Generic Provider: roles: `admin`, `domain_admin`, `catalogeditor`

GET: Gets an instance of a given type.

URL pattern
`/orchestrator/v2/instancetypes/{name}/instances/{id}`

Accepts
application/JSON

Content-Type
/

Normal Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Resource Instance Response

Authorization
Instance provider dependent. Generic Provider: Access Control Link with **view** set to **true** for the domain, project and role you are working on.

PUT: Updates an instance of a given type.

URL pattern
`/orchestrator/v2/instancetypes/{name}/instances/{id}`

Accepts
application/JSON

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Request
Resource Instance Request (partial)

Response
Resource Instance Request

Authorization
Instance provider dependent. Generic Provider: Access Control Link with **modify** set to **true** for given domain, project and role you are working on.

DELETE: Deletes an instance of a given type.

URL pattern: `/orchestrator/v2/instancetypes/{name}/instances/{id}`

Accepts
/

Content-Type
application/JSON

Normal response Codes
204

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Authorization

Instance provider dependent. Generic Provider: Access Control Link with **modify** set to **true** for given domain, project and role of the user.

GET: Lists actions defined on a given instance of a given type

URL pattern: `/orchestrator/v2/instancetypes/{instancetype}/services`

For heat, the URL pattern is `/orchestrator/v2/instancetypes/heat/services`

For VM, the URL pattern is `/orchestrator/v2/instancetypes/openstackvms/services`

Accepts

/

Content-Type

application/JSON

Normal Response Codes

200

Error Response Codes

401 unauthorized

404 not found

500 internal server error

Authorization

Instance provider dependent. Generic Provider: Access Control Link with **view** set to **true** on the instance and services for given domain, project and role of the user.

Request format

Heat service actions:

```
curl -ku admin:openstack1 -X GET -d {} -H
"Content-Type:application/json"
https://ico-09-26-node1.cil.rtp.raleigh.ibm.com/orchestrator/
v2/instancetypes/heat/services
```

OpenStackvms service actions:

```
curl -ku admin:openstack1 -X GET -d {} -H "Content-Type:application/json"
https://ico-09-26-node1.cil.rtp.raleigh.ibm.com/orchestrator/
v2/instancetypes/openstackvms/services
```

POST: Launch given action on a given instance of a given type.

URL pattern: `/orchestrator/v2/instancetypes/openstackvms/services/{service-id}/launch`

Content-Type

application/JSON

Normal Response Codes

202 accepted

Error Response Codes

401 unauthorized

404 not found

500 internal server error

Authorization

Instance provider dependent and ACL with **use** set to **true** on the services for given domain, project and role of the user. Access Control Link with **use** set to **true** on the instance and services for given domain, project and role of the user.

Request format

```
curl -ku admin:openstack1 -X POST -d '{"instances":
["https://BPMSERVER:443/orchestrator/v2/instancetypes/{intance-type}
/instances/{instance-id}"]}'
-H "Content-type:application/json" https://BPMSERVER:443/
orchestrator/v2/instancetypes/openstackvms/
services/{service-id}/launch
```

For example,

```
curl -ku admin:openstack1 -X POST -d '{"instances":
["https://ico-09-26-node1.cil.rtp.raleigh.ibm.com:443/
orchestrator/v2/instancetypes/openstackvms/instances/
kvm-allinone--919fd92f-c358-4c5f-9849-8e897c093db4"]}'
-H "Content-type:application/json"
https://ico-09-26-node1.cil.rtp.raleigh.ibm.com:443/orchestrator/
v2/instancetypes/openstackvms/services/36/launch
```

Resource instance providers

You can use OpenStack, catalog, and generic providers.

OpenStack providers

You can use the following OpenStack providers.

Domain provider

This provider lists OpenStack Keystone domains.

Instance Type

domain

Provider Class

`com.ibm.orchestrator.core.instance.providers.openstack.OpenstackDomainProvider`

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc		
displayname	String	Name of the domain	y	asc/desc	y	y
description	String	Description of the domain	y	asc/desc		y
icon	String	Not used				
detailsURL	String	URI to display details of the domain				
parm	String	JSON result from OpenStack				
enabled	Boolean	Domain enablement status	y	asc/desc	y	
domain	String	Domain ID				
tags	String	Tags to control				

Group provider

This provider lists OpenStack Keystone groups.

Instance Type

group

Provider Class

`com.ibm.orchestrator.core.instance.providers.openstack.OpenstackGroupProvider`

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc		
displayname	String	Name of the group	y	asc/desc	y	y
description	String	Description of the group	y	asc/desc		y
icon	String	Not used				
detailsURL	String	URI to display details of the group				
parm	String	JSON result from OpenStack				
domain	String	Domain ID				
tags	String	Tags to control action availability				

Heat stack provider
This provider lists OpenStack Heat stacks.

Instance Type
heat

Provider Class

```
com.ibm.orchestrator.core.instance.providers.openstack.OpenstackHeatStackProvider
```

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc		
displayname	String	Name of the server	y	asc/desc	y	y
description	String	Description of the server	y			y
icon	String	Not used				
detailsURL	String	URI to display details of the server				
parm	String	JSON result from OpenStack				
status	String	Server status in OpenStack	y	asc/desc	y	

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
openstackId	String	Server ID in OpenStack				
region	String	OpenStack Region	y			
updated	String	Time of last update	y	asc/desc		
created	String	Creation time		asc/desc		
tags	String	Tags to control action availability				

Heat template provider
This provider lists OpenStack Heat Orchestration templates.

Instance Type
stacktemplate

Provider Class
`com.ibm.orchestrator.core.instance.providers.generic.GenericProvider`

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID				
displayname	String	Name of the template	yes	asc/desc	yes	yes
description	String	Description of the template	yes	asc/desc	yes	yes
icon	String	Not used				
detailsURL	String	URI to display details of the template	yes			
parm	String	JSON result from OpenStack				
tags	String	Tags to control action availability				

Project provider
This provider lists OpenStack Keystone projects.

Instance Type
project

Provider Class
`com.ibm.orchestrator.core.instance.providers.openstack.OpenstackProjectProvider`

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc		
displayname	String	Name of the project	y	asc/desc	y	y

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
description	String	Description of the project	y	asc/desc		y
icon	String	Not used				
detailsURL	String	URI to display details of the project				
parm	String	JSON result from OpenStack				
enabled	Boolean	Project enablement status	y	asc/desc	y	
domain	String	Domain ID				
tags	String	Tags to control action availability				

User provider
This provider lists OpenStack Keystone users.

Instance Type
user

Provider Class
`com.ibm.orchestrator.core.instance.providers.openstack.OpenstackUserProvider`

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc		
displayname	String	Name of the user	y	asc/desc	y	y
description	String	Description of the user				y
icon	String	Not used				
detailsURL	String	URI to display details of the user				
parm	String	JSON result from OpenStack				
enabled	Boolean	User enablement status	y	asc/desc	y	
defaultProject Id	String	Default project for this user				

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
email	String	Email address of this user	y	asc/desc	y	
domain	String	Domain ID				
tags	String	Tags to control action availability				

VM provider
This provider lists OpenStack Nova virtual servers.

Instance Type
`openstackvms`

Provider Class
`com.ibm.orchestrator.core.instance.providers.openstack.OpenstackVMProvider`

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchabl e
id	String	Instance ID		asc/desc		
displayname	String	Name of the server	y	asc/desc	y	y
description	String	Description of the server	y			y
icon	String	Not used				
detailsURL	String	URI to display details of the server				
parm	String	JSON result from OpenStack				
openstackId	String	Server ID in OpenStack				
status	String	Server status in OpenStack	y	asc/desc	y	
region	String	OpenStack Region	y			
updated	String	Time of last update	y	asc/desc		
created	String	Creation time		asc/desc		
keyPair	String	SSH key pair	y	asc/desc		
patternInstanceType	String	Instance type of the pattern instance the server belongs				
patternName	String	Name of the pattern instance the server belongs	y			
patternURI	String	URI to display details of the pattern instance the server belongs				
tags	String	tags to control action availability				

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
ipAddresses	String	IP addresses assigned to the server	y	asc/desc		

Self-Service Catalog providers
You can use the following Self-Service Catalog providers.

Offering provider
This provider lists Self-Service Catalog offerings.

Instance Type
offering

Provider Class
`com.ibm.orchestrator.core.instance.providers.catalog.CatalogOfferingProvider`

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc	y	
displayname	String	Name of the offering	y	asc/desc	y	y
description	String	Description of the offering	y	asc/desc	y	y
icon	String	Offering icon		asc/desc	y	
detailsURL	String	URI to display details of the offering				
parm	String	JSON result from Catalog				
type	String	Type of this service		asc/desc		
category	String	Category of this offering	y	asc/desc		
tags	String	Tags to control action availability				

Action provider
This provider lists instance actions. Actions are services that can be run on one or more selected instances.

Instance Type
Offering

Provider Class
`com.ibm.orchestrator.core.instance.providers.catalog.CatalogActionProvider`

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc	y	
displayname	String	Name of the action	y	asc/desc	y	y

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
description	String	Description of the action	y	asc/desc	y	y
icon	String	Action icon		asc/desc	y	
detailsURL	String	URI to display details of the action				
parm	String	JSON result from Catalog				
type	String	Type of this service	y	asc/desc	y	
category	String	Category of this offering		asc/desc	y	
instancetype	String	Type of instance upon which this action can be run	y	asc/desc	y	
tagsAsString	String	Tags that are combined to a single string	y			
tags	String	Tags to control action availability				

Catalog Category provider
This provider lists Self-Service Catalog categories. Categories might contain one or more offerings.

Instance Type
category

Provider Class
`com.ibm.orchestrator.core.instance.providers.catalog.CatalogCategoryProvider`

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc	y	
displayname	String	Name of the category	y	asc/desc	y	y
description	String	Description of the category	y	asc/desc	y	y
icon	String	Category icon		asc/desc	y	
detailsURL	String	URI to display details of the category				
parm	String	JSON result from Catalog				

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
tags	String	Tags to control action availability				

Microsoft Azure providers
These topics describe the Microsoft Azure providers.

Azure Cloud Service provider
This topic describes the Azure Cloud Service provider.

Instance Type
azurecloudservice

Provider jar file
`com.ibm.orchestrator.plugin.azue-1.0.jar`

Provider Class
`com.ibm.orchestrator.plugin.azure.AzureServiceProvider`

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID			y	
displayname	String	Name of the Cloud service	y	asc/desc	y	y
description	String	Description of the Cloud service	y	asc/desc	y	y
created	String	Date of creation				
status	String	Status	y			
region	String	Azure region in IBM Cloud Orchestrator	y			
label	String	Label				
location	String	Azure datacenter location				
affinitygroup	String	Affinity group of the Cloud service				
subscriptionid	String	Subscription ID				
updated	String	Date of the last update of the Cloud service	y			
reverseDnsFqdn	String	Reverse DNS FQDN				

Azure Cloud service deployment provider
This topic describes the Azure Cloud service deployment provider.

Instance Type
azureprovider

Provider jar file
`com.ibm.orchestrator.plugin.azue-1.0.jar`

Provider Class
`com.ibm.orchestrator.plugin.azure.AzureServiceDeployment.Provider`

Attribute Name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID			y	
displayname	String	Name of the Cloud service	y	asc/desc	y	y
description	String	Description of the Cloud service	y	asc/desc	y	y
cloudservice	String	Name of the Cloud Service the Deployment belongs to	y			
deploymentslot	String	Deployment slot	y			
created	String	Date of creation				
status	String	Status	y			
region	String	Azure region in IBM Cloud Orchestrator	y			
label	String	Label				
location	String	Azure datacenter location				
subscriptionid	String	Subscription ID				
updated	String	Date of the last update of the Cloud service	y			

Generic provider
This provider lists generic resources. The provider might be registered multiple times under different instance types.

Instance Type
`<not registered by default>`

Provider Class
`com.ibm.orchestrator.core.instance.providers.generic.GenericProvider`

Attribute name	Type	Description	Displayed in UI	Sortable	Filterable	Searchable
id	String	Instance ID		asc/desc	y	
displayname	String	Name of the instance	y	asc/desc	y	y
description	String	Description of the instance	y	asc/desc	y	y
icon	String	Instance icon		asc/desc	y	
detailsURL	String	URI to display details of the instance				
parm	String	JSON object that is used to store additional information				y
tags	String	Tags to control action availability				

Task engine REST API V2

This topic lists the JSON formats for the task engine REST API V2.

JSON Formats

Task Response

```
{
     "updated_iso" : "2014-03-31T13:05:14+0200",
     "description_message" : "The process is complete.",
     "domain" : "Default",
     "created" : 1396263830875,
     "error" : {
         "resourceBundle" : "com.ibm.orchestrator.messages.orchestratormessages",
         "message" : null,
         "messageKey" : "BPM_PROCESS_COMPLETE",
         "args" :
         [
             "3"
         ]
     },
     "user" : "ksadmin",
     "parm" : {
         .....
     },
     "created_iso" : "2014-03-31T13:03:50+0200",
     "status_localized" : "Completed",
     "error_message" : "CTJCO0002I: Business process instance 3 completed successfully.",
     "status" : "COMPLETED",
     "eventTopic" : "com/ibm/orchestrator/serviceinstance/plan/ibm_bpm_process",
     "delayInSeconds" : 30,
     "project" : "admin",
     "id" : "1003",
     "updated" : 1396263914896,
     "description" : {
         "resourceBundle" : "com.ibm.orchestrator.messages.orchestratormessages",
         "message" : null,
         "messageKey" : "PROCESS_COMPLETE",
         "args" :
         [
         ]
     }
 }
```

Task Response

[Task Response 1,....., Task Response n]

GET: Get all tasks

URL method
`/orchestrator/v2/tasks`

Accepts
/

Content-Type
application/JSON

Normal response Codes
200

Error Response Codes
500 internal server error

Request Parameters
Expand: If set to **serviceInstance** the information service instance referred by the attribute **serviceInstanceId** gets returned in the Task Response in the parameter **serviceInstance**.

Response
Tasks Response

Authorization
No authorization is needed but the output is restricted to tasks from all users within the current project. Users with role admin can see all the tasks.

POST: Get all tasks

URL method
`/orchestrator/v2/tasks`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
201 created

Error Response Codes
401 unauthorized

500 internal server error

Response
Tasks Response

Authorization
Role: admin

GET: Get the task with a given ID

URL method
`/orchestrator/v2/tasks/{id}`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
404 not found

500 internal server error

Response
Task Response

Authorization
No authorization is needed but the output is restricted to role

PUT: Get the task with a given ID

URL pattern
`/orchestrator/v2/tasks/{id}`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Task Response

Authorization
Role: admin or in same project as task

DELETE: Delete the task with a given ID

URL pattern
`/orchestrator/v2/tasks/{id}`

Accepts
/

Content-Type
application/JSON

Normal response Codes
204

Error Response Codes
401 unauthorized

404 not found

500 internal server error

Response
Task Response

Authorization
Role: admin

Configuration providers REST API

Three REST APIs manage the entities under Administration. A REST API consumer can manage the entities in two ways.

- Manage the entities directly
  - To manage the entities by using the core REST APIs, you must use the core REST APIs or OpenStack APIs. If you directly manage entities like domains, users, groups, projects, quotas, you must call the OpenStack APIs. For categories, offerings and actions you must call the core REST APIs. In both cases, the actions are not run and you must handle all the dependencies.

- Process flows
  - If you use the logic of the Business Process Manager processes as previously described, then you need to launch the actions through the core REST API. This REST API is designed for external usage as it involves the business logic that is implemented by the provider. For example, if the REST API provider decides to add an approval process to the **Modify Quota** action of a project, then the provider forces the REST API consumer to launch the action to get the approval logic applied.

Managing entities by using the Core REST APIs

This topic describes how to manage entities by using the Core REST APIs.

Before you begin

The mapping in table 1 shows which REST APIs are used to manage the entities.

Entity	REST API	Endpoint
Domain	OpenStack Keystone API	`v3/domains`
Project	OpenStack Keystone API	`v3/projects`
User	OpenStack Keystone API	`v3/users`
Group	OpenStack Keystone API	`v3/group`
Quota	OpenStack Compute API	`v2.0/{tenant_id}/os-quota-sets`
Category	Core REST API	`orchestrator/v2/categories`
Offering	Core REST API	`orchestrator/v2/offerings`
Action	Core REST API	`orchestrator/v2/offerings`

Managing entities by using actions

Any action as described in the **Managing Entities by using Core Services REST APIs** documentation can be started per API. The action must be started in the Core Services REST API. The following procedure is an example of how to start the **Edit Project** action on a project through API.

Procedure

1. Get the project provider and the URL for its instances in the instances attribute of the response.

   HTTP method:

   GET

   Example:

   `https://ico_server.example.com:443/orchestrator/v2/instancetypes/project`

```
{
  "href": "https://ico_server.example.com:443/
orchestrator/v2/instancetypes/project",
  "item": {
    "provider": "com.ibm.orchestrator.core.instance
.providers.openstack.OpenstackProjectProvider",
    "detailsview": {
      "application": "SCOMT",
      "humanservice": "Show Project Details"
    },
    "keyfields": [
      {
        "instanceattribute": "displayname",
        "header": "Name"
      },
      {
        "instanceattribute": "description",
        "header": "Description"
      },
      {
        "instanceattribute": "enabled",
        "header": "Enabled?"
      }
    ],
    "tags": [
      "enabled",
      "disabled"
    ],
    "icon": "Web Icon:glyphicons_232_cloud",
    "type": "admin",
    "name": "project",
    "description": "Show your OpenStack projects.",
    "displayname": "Projects"
  },
  "instances": {
    "href":
 "https://ico_server.example.com:443/orchestrator
/v2/instancetypes/project/instances"
  },
  "services": {
    "href": "https://ico_server.example.com:443/
orchestrator/v2/instancetypes/project/services"
  }
}
```

2. Get the instance that you want to manage and find the ID and name. You can also use the API filter to search the name attribute.

   HTTP Method:

   GET

   Example:

   `https://ico_server.example.com:443/orchestrator/v2/instancetypes/project/instances`

```
{
  "href": "https://ico_server.example.com:443
/orchestrator/v2/instancetypes/project/instances",
  "start": 0,
  "limit": 10,
  "total": 4,
  "first": {
    "href":
"https://ico_server.example.com:443/orchestrator
/v2/instancetypes/project/instances?_limit=10&_start=0"
   },
   "previous": null,
   "next": null,
   "last": {
     "href":
 "https://ico_server.example.com:443
/orchestrator/v2/instancetypes/project/instances?_limit=10&_start=0"
   },
   "items": [
     {
       "href":
 "https://ico_server.example.com:443/
orchestrator/v2/instancetypes/project/instances/
4ae7ade7e4724c69ab90246ea72965e6",
       "item": {
         "enabled": true,
         "domain": "4ae7ade7e4724c69ab90246ea72965e6",
         "tags": [
           "enabled"
         ],
         "icon": null,
         "id": "4ae7ade7e4724c69ab90246ea72965e6",
         "parm": {
           "enabled": true,
           "domain_id": "default",
           "links": {
             "self":
 "http://192.0.2.35:5000/v3/projects
       /4ae7ade7e4724c69ab90246ea72965e6"   },
           "id": "4ae7ade7e4724c69ab90246ea72965e6",
           "name": "admin",
           "description": "admin Tenant"
         },
         "description": "admin Tenant",
         "detailsURL":
 "https://ico_server.example.com:443/teamworks/
executeServiceByName?processApp=SCOMT&serviceName=
Show+Project+Details&tw.local.projectId=
4ae7ade7e4724c69ab90246ea72965e6&tw.local.domainId=
default&tw.local.authUser=admin&tw.local.authDomain=
Default&tw.local.authProject=admin",
         "displayname": "admin"
       }
     },
     ...
   ]
 }
```

3. Get the actions that are applicable to projects. Get the link to in the services attribute of the response in step 1. Get the services and find the **Edit Project** action by name and remember its ID.

   HTTP method:

   GET

   Example:

   `https://ico_server.example.com:443/orchestrator/v2/instancetypes/project/services`

```
...
{
   "href": "https://ico_server.example.com:443/
orchestrator/v2/instancetypes/project/services/69",
   "item": {
     "human_service": "Edit Project Action",
     "priority": 0,
     "human_service_app_name": "SCOrchestrator Multi-Tenancy Toolkit",
     "implementation_type": null,
     "created": 1401827772,
     "human_service_app_short_name": "SCOMT",
     "process_app_id": "2066.227c57b3-a5e5-4e5b-a283-c920cf9bed50",
     "acl": [
         ...
      ],
     "name": "Edit Project",
     "ownerid": 0,
     "instancetype": "project",
     "process": "Edit Project Action",
     "operation_type": "singleInstanceAction",
     "human_service_app_id": "2066.227c57b3-a5e5-4e5b-a283-c920cf9bed50",
     "tags": [
       "enabled"
     ],
     "process_app_name": "SCOrchestrator Multi-Tenancy Toolkit",
     "icon": "act16_return",
     "updated": 1401827772,
     "id": 69,
     "process_app_short_name": "SCOMT",
     "description": "Edit the project details",
     "category": 31
   }
 },
...
```

4. Start the action with the ID from step 3, passing the ID of the selected instance (project) from step 2 in the request body. The call returns a task that is in the state **NEW** and a new ID.

   Note: For all actions of type "createInstance", the ID of the domain must be passed in the "instances" array of the PUT request.

   HTTP method:

   POST

   Body:

```
{
  "instances": ["default"]
}
```

Example:

https://ico_server.example.com:443/orchestrator/v2/instancetypes/project/services/69/launch

```
{
  "updated_iso": "1970-01-01T01:00:00+0100",
  "description_message": "HS_OFFERING_INVOCATION",
  "domain": "Default",
  "message": "Launched",
  "created": 1402569924045,
  "error": null,
  "user": "admin",
  "parm": {
    "plan": {
      "human_service": "Edit Project Action",
      "priority": 0,
      "human_service_app_name": "SCOrchestrator Multi-Tenancy
                                 Toolkit",
      "implementation_type": null,
      "created": 1401827772,
      "human_service_app_short_name": "SCOMT",
      "process_app_id": "2066.227c57b3-a5e5-4e5b-a283-c920cf9bed50",
      "acl": [
        ...
      ],
      "name": "Edit Project",
      "ownerid": 0,
      "instancetype": "project",
      "process": "Edit Project Action",
      "operation_type": "singleInstanceAction",
      "human_service_app_id": "2066.227c57b3-a5e5-4e5b-a283-
                               c920cf9bed50",
      "tags": [
        "enabled"
      ],
      "process_app_name": "SCOrchestrator Multi-Tenancy Toolkit",
      "icon": "act16_return",
      "updated": 1401827772,
      "id": 69,
      "process_app_short_name": "SCOMT",
      "description": "Edit the project details",
      "category": 31
    },
    "instances": [
      "default"
    ]
  },
  "created_iso": "2014-06-12T12:45:24+0200",
  "status_localized": "New",
  "error_message": null,
  "status": "NEW",
  "eventTopic": "com/ibm/orchestrator/serviceinstance/plan
               /ibm_bpm_process",
  "delayInSeconds": 0,
  "project": "admin",
  "id": "1521",
  "updated": 0,
  "redirect": "/teamworks/executeServiceByName?
processApp=SCOMT&serviceName=Edit+Project+Action&tw.
local.operationContextId=1521",
   "description": {
     "resourceBundle": "com.ibm.orchestrator.messages.
                        orchestratormessages",
     "message": "HS_OFFERING_INVOCATION",
     "messageKey": "HS_OFFERING_INVOCATION",
     "args": [
       "Edit Project"
     ]
   }
 }
```

5. Set the parameters of the task that are the input for the action. Then, set the status of the task to **QUEUED** to queue the task for execution. In the body, the description is set to **test** and the other attributes remain the same.

   HTTP method:

   PUT

   Body:

```
{
"status":"QUEUED",
          "parm":{"OperationParameter":"<variable type=\"Project\">\n
            <name type=\"String\"><![CDATA[admin]]><\/name>\n
            <description type=\"String\"><![CDATA[test]]>
            <\/description>\n
            <enabled type=\"Boolean\"><![CDATA[true]]><\/enabled>\n
            <id type=\"String\"><![CDATA[
                    4ae7ade7e4724c69ab90246ea72965e6]]><\/id>\n
            <domainId type=\"String\"><![CDATA[default]]>
                               <\/domainId>\n<\/variable>"}
    }
```

   Example:

   `https://ico_server.example.com:443/kernel/tasks/1521`

6. Check whether the task succeeded or failed. The status switches to **RUNNING**. If the task succeeds the status says **COMPLETED**. If the task fails the status says **FAILED** and an **error_message** is shown. In the example, the process completed.

   HTTP method:

   GET

   Example:

   `https://ico_server.example.com:443/kernel/tasks/1521`

```
{
   "error_message": "CTJCO0002I: Business process instance 79
                     completed successfully.",
   "status": "COMPLETED",
}
```

7. Verify whether the action applied the changes on the entity. Finally, it is possible to ensure if the change happened on the instance.

   HTTP method:

   GET

   Example:

   `https://ico_server.example.com:443/orchestrator/v2/instancetypes/project/instances/4ae7ade7e4724c69ab90246ea72965e6`

```
{
  "href": "https://ico_server.example.com:443/orchestrator/v2
           /instancetypes/project
           /instances/4ae7ade7e4724c69ab90246ea72965e6",
  "item": {
    "enabled": true,
    "domain": "4ae7ade7e4724c69ab90246ea72965e6",
    "tags": [
      "enabled"
    ],
    "icon": null,
    "id": "4ae7ade7e4724c69ab90246ea72965e6",
    "parm": {
      "enabled": true,
      "domain_id": "default",
      "links": {
        "self": "http://192.0.2.35:5000/v3/projects
                 /4ae7ade7e4724c69ab90246ea72965e6"
      },
      "id": "4ae7ade7e4724c69ab90246ea72965e6",
      "name": "admin",
      "description": "test"
    },
    "description": "test",
    "detailsURL":
 "https://ico_server.example.com:443
/teamworks/executeServiceByName?processApp=SCOMT&serviceName
=Show+Project+Details&tw.local.
projectId=4ae7ade7e4724c69ab90246ea72965e6&tw.local.
domainId=default&tw.local.authUser=admin&tw.local
.authDomain=Default&tw.local.authProject=admin",
     "displayname": "admin"
   }
 }
```

Core REST API for compatibility with earlier versions

The REST APIs described in the following sections were replaced in IBM Cloud Orchestrator but they are still valid for compatibility with SmartCloud Orchestrator 2.3.

Self-service offering REST API

You can use this set of REST API calls to interact with the self-service offerings in IBM Cloud Orchestrator.

Create a self-service offering

Use this REST call to create a self-service offering.

Available HTTP method

Table 35. Create a self-service offering REST API call

HTTP method	POST
URL pattern	`https://hostname/resources/services`

Response	The response of the server contains the specified offering. It has the following set of attributes:
	`{`
	`category:`
	`created:`
	`description:`
	`human_service:`
	`human_service_app_id:`
	`human_service_app_name:`
	`human_service_app_short_name:`
	`icon:`
	`id:`
	`implementation_type:`
	`name:`
	`operation_type:`
	`ownerid:`
	`process:`
	`process_app_id:`
	`process_app_name:`
	`process_app_short_name:`
	`updated:`
	`}`

Return values	• 201 Returns the created service or offering • 500 Internal server error

An entry has the following attributes:

- **`category`** - optional, the category to which the offering belongs.

- **created** - the creation time of the self-service offering, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.
- **description** - optional, a short description of the offering.
- **human_service** - optional, the URL of an IBM Business Process Manager human service (coach), a user interface to provide user input.
- **human_service_app_id** - optional, depending on the `human_service` attribute, the ID of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_name** - optional, depends on if the `human_service` attribute is set, the name of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_short_name** - the short name of the IBM Business Process Manager human service application.
- **icon** - optional, an offering can have an icon assigned that is displayed inside the Self-Service Catalog.
- **id** - the ID of the offering.
- **implementation_type** - the two possible values are 'ibm_bpm_process' and 'script'.
- **name** - the name of the offering.
- **operation_type** - the only possible value is "service".
- **ownerid** - the owner of the user who triggered the offering.
- **process** - the name of the IBM Business Process Manager process that is bound to the offering.
- **process_app_id** - the ID of the IBM Business Process Manager application in which the process is defined.
- **process_app_name** - the name of the IBM Business Process Manager application in which the process is defined.
- **process_app_short_name** - the short name of the IBM Business Process Manager process application.
- **updated** - the time when the self-service offering was last updated, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

The following listing shows an example response that can be retrieved by way of the request:

```
{
   "human_service": "Sample_ReportProblem",
   "human_service_app_name": "SCOrchestrator_Toolkit",
   "implementation_type": "ibm_bpm_process",
   "human_service_app_short_name": "SCOTLKT",
   "process_app_id": "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
   "name": "Problem report",
     "created": 1242965374865,
    "updated": 1242965392870,
   "ownerid": 2,
   "process": "Sample_Report",
   "operation_type": "service",
   "human_service_app_id": "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
   "process_app_name": "SCOrchestrator_Toolkit",
   "icon": "Configuration Icon:ge100_config_24",
   "id": 5,
   "process_app_short_name": "SCOTLKT",
   "description": "Report a problem",
   "category": ""
}
```

List all self-service offerings
Use this REST API method to list all self-service offerings.

Available HTTP method

Table 36. Get list of all self-service offerings

HTTP method	GET
URL pattern	`https://hostname/resources/services`

Response	The response of the server contains a list of available self-service offerings. Each offering has the following set of attributes:
	`{`
	`   category:`
	`   created:`
	`   description:`
	`   human_service:`
	`   human_service_app_id:`
	`   icon:`
	`   id:`
	`   implementation_type:`
	`   name:`
	`   operation_type:`
	`   ownerid:`
	`   process:`
	`   process_app_id:`
	`   updated:`
	`}`

Return values	• 200 Returns the list of offerings • 500 Internal server error

An entry has the following attributes:

- **`category`** - optional, a category to which the offering belongs.
- **`created`** - the creation time of the self-service offering, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.
- **`description`** - optional, a short description for the offering.
- **`human_service`** - optional, the URL of an IBM Business Process Manager human service (coach), a user interface to provide user input.
- **`human_service_app_id`** - optional, depending on the `human_service` attribute. The ID of the IBM Business Process Manager application to which the `human_service` belongs.
- **`icon`** - optional, an offering can have an icon assigned that is displayed inside the Self-Service Catalog.
- **`id`** - ID of the offering.
- **`implementation_type`** - possible values are `ibm_bpm_process` or `script`.
- **`name`** - the name of the offering.
- **`operation_type`** - the only possible value is `service`.
- **`ownerid`** - the owner of the user who triggered the offering.
- **`process`** - the name of the IBM Business Process Manager process that is bound to the offering.
- **`process_app_id`** - the ID of the IBM Business Process Manager application in which a process is defined.
- **`updated`** - the time when the self-service offering was last updated, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

The following listing shows an example response that can be retrieved by way of the request:

```
[
   {
      "human_service": "Sample_ReportProblem",
      "implementation_type": "ibm_bpm_process",
      "process_app_id": "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
      "name": "Problem report",
        "created": 1242965374865,
        "updated": 1242965392870,
      "ownerid": 2,
      "process": "Sample_Report",
      "operation_type": "service",
      "human_service_app_id": "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
      "icon": "Job Icon:ge100_job_24",
      "id": 5,
      "description": "Report a problem",
      "category": 5
   }
]
```

Get entries for a specific self-service offering
Use this REST call to retrieve information about a self-service offering with a specified ID.

Available HTTP method

Table 37. Get entries for a specific self-service offering REST API call

HTTP method	GET
URL pattern	`https://hostname/resources/services/{id}[?acl=true]`
Response	The response of the server contains the specified offering. It has the following set of attributes: `{ acl: category: created: description: human_service: human_service_app_id: human_service_app_name: human_service_app_short_name: icon: id: implementation_type: name: operation_type: ownerid: process: process_app_id: process_app_name: process_app_short_name: updated: }` **Note:** The acl attribute is only returned when the optional query parameter acl is passed with the value `true`.
Return values	• 200 Returns the service or offering that is associated with the given ID • 403 If the client is not on the offering's ACL, they are not authorized to perform this action. • 404 No offering exists with the given ID • 500 Internal server error

An entry has the following attributes:

- **`category`** - optional, the category to which the offering belongs.

- **created** - the creation time of the self-service offering, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.
- **description** - optional, a short description of the offering.
- **human_service** - optional, the URL of an IBM Business Process Manager human service (coach), a user interface to provide user input.
- **human_service_app_id** - optional, depending on the `human_service` attribute, the ID of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_name** - optional, depends on if the `human_service` attribute is set, the name of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_short_name** - the short name of the IBM Business Process Manager human service application.
- **icon** - optional, an offering can have an icon assigned that is displayed inside the Self-Service Catalog.
- **id** - the ID of the offering.
- **implementation_type** - the two possible values are `ibm_bpm_process` and `script`.
- **name** - the name of the offering.
- **operation_type** - the only possible value is `service`.
- **ownerid** - the owner of the user who triggered the offering.
- **process** - the name of the IBM Business Process Manager process that is bound to the offering.
- **process_app_id** - the ID of the IBM Business Process Manager application in which the process is defined.
- **process_app_name** - the name of the IBM Business Process Manager application in which the process is defined.
- **process_app_short_name** - the short name of the IBM Business Process Manager process application.
- **updated** - the time when the self-service offering was last updated, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

The following listing shows an example response that can be retrieved by way of the request:

```
{
   "human_service": "Sample_ReportProblem",
   "human_service_app_name": "SCOrchestrator_Toolkit",
   "implementation_type": "ibm_bpm_process",
   "human_service_app_short_name": "SCOTLKT",
   "process_app_id": "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
   "name": "Problem report",
     "created": 1242965374865,
    "updated": 1242965392870,
   "ownerid": 2,
   "process": "Sample_Report",
   "operation_type": "service",
   "human_service_app_id": "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
   "process_app_name": "SCOrchestrator_Toolkit",
   "icon": "Configuration Icon:ge100_config_24",
   "id": 5,
   "process_app_short_name": "SCOTLKT",
   "description": "Report a problem",
   "category": 5
}
"acl":
   [
{
    "domain": "default",
    "view": true,
    "role": "default",
    "use": false,
    "resourceType": "SCOService",
    "project": "default",
    "resourceId": 101,
    "modify": true,
    "id": 151
}
   ]
```

Delete a specific self-service offering
Use this REST API call to delete a specific self-service offering.

Available HTTP method

Table 38. Delete a self-service offering REST API call

HTTP method	DELETE
URL pattern	`https://hostname/resources/services/{id}`
Response	The self-service offering is deleted.
Return values	• 204 Deletes an offering and its ACL • 401 The client is not authorized to perform this action as they are not on the offering's ACL • 404 No offering has the given ID. • 500 Internal server error

Update a specific self-service offering
Use this REST API call to update a specific self-service offering

Available HTTP method

Table 39. Update a self-service offering REST API call

HTTP method	PUT
URL pattern	`https://hostname/resources/services/{id}`
Response	The self-service offering is updated.
Return values	• 201 - Updates an existing offering • 400 - Decode failure. Request body does not contain valid JSON • 401 - Authorization failure • 404 - Update failure • 500 - Internal server error

Structure of request body:

```
{  "<attribute>": "<attribute_value>" }
```

The following listing shows sample content of request body to update the description of a self-service offering:

```
{    "description": "Changed description" }
```

Self-service catalog REST API
You can use this set of REST API calls to interact with the Self-Service Catalog in IBM Cloud Orchestrator.

Table 40. Categories

ID	The unique identifier of the category

Table 40. Categories (continued)	
Name	The name of the category, that is, the name that appears in the service catalog
Description	The description of the category
Icon	The name of the icon that is used to represent the category
Isbuiltin	"1 or 0" for "true or false" whether the category is provided by the product

Create category
Use this REST call to create a category.

Available HTTP method

Table 41. Create a category REST API call	
HTTP method	POST
URL pattern	`/resources/automationcategories`
Response	The response of the server contains the specified offering. It has the following set of attributes: `{ category: created: description: human_service: human_service_app_id: human_service_app_name: human_service_app_short_name: icon: id: implementation_type: name: operation_type: ownerid: process: process_app_id: process_app_name: process_app_short_name: updated: }`

An entry has the following attributes:

- **`category`** - optional, the category to which the offering belongs.
- **`created`** - the creation time of the self-service offering, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.
- **`description`** - optional, a short description of the offering.
- **`human_service`** - optional, the URL of an IBM Business Process Manager human service (coach), a user interface to provide user input.
- **`human_service_app_id`** - optional, depending on the `human_service` attribute, the ID of the IBM Business Process Manager application to which the human service belongs.
- **`human_service_app_name`** - optional, depends on if the `human_service` attribute is set, the name of the IBM Business Process Manager application to which the human service belongs.
- **`human_service_app_short_name`** - the short name of the IBM Business Process Manager human service application.
- **`icon`** - optional, an offering can have an icon assigned that is displayed inside the Self-Service Catalog.
- **`id`** - the ID of the offering.

- **`implementation_type`** - the two possible values are `ibm_bpm_process` and `script`.
- **`name`** - the name of the offering.
- **`operation_type`** - the only possible value is `service`.
- **`ownerid`** - the owner of the user who triggered the offering.
- **`process`** - the name of the IBM Business Process Manager process that is bound to the offering.
- **`process_app_id`** - the ID of the IBM Business Process Manager application in which the process is defined.
- **`process_app_name`** - the name of the IBM Business Process Manager application in which the process is defined.
- **`process_app_short_name`** - the short name of the IBM Business Process Manager process application.
- **`updated`** - the time when the self-service offering was last updated, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

```
{
"isbuiltin": 0,
"icon": "Web Category Icon:ge100_webcatalog_24",
"name": "Bucket",
"description": ""
}
```

Get the list of categories
Use this REST call to get a list of categories.

Available HTTP method

Table 42. Get the list of categories REST API call

HTTP method	GET
URL pattern	`/resources/automationcategories`

Response	The response of the server contains the specified offering. It has the following set of attributes:
	`{`
	`category:`
	`created:`
	`description:`
	`human_service:`
	`human_service_app_id:`
	`human_service_app_name:`
	`human_service_app_short_name:`
	`icon:`
	`id:`
	`implementation_type:`
	`name:`
	`operation_type:`
	`ownerid:`
	`process:`
	`process_app_id:`
	`process_app_name:`
	`process_app_short_name:`
	`updated:`
	`}`

An entry has the following attributes:

- **`category`** - optional, the category to which the offering belongs.
- **`created`** - the creation time of the self-service offering, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

- **description** - optional, a short description of the offering.
- **human_service** - optional, the URL of an IBM Business Process Manager human service (coach), a user interface to provide user input.
- **human_service_app_id** - optional, depending on the `human_service` attribute, the ID of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_name** - optional, depends on if the `human_service` attribute is set, the name of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_short_name** - the short name of the IBM Business Process Manager human service application.
- **icon** - optional, an offering can have an icon assigned that is displayed inside the Self-Service Catalog.
- **id** - the ID of the offering.
- **implementation_type** - the two possible values are `ibm_bpm_process` and `script`.
- **name** - the name of the offering.
- **operation_type** - the only possible value is `service`.
- **ownerid** - the owner of the user who triggered the offering.
- **process** - the name of the IBM Business Process Manager process that is bound to the offering.
- **process_app_id** - the ID of the IBM Business Process Manager application in which the process is defined.
- **process_app_name** - the name of the IBM Business Process Manager application in which the process is defined.
- **process_app_short_name** - the short name of the IBM Business Process Manager process application.
- **updated** - the time when the self-service offering was last updated, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

```
{
   "isbuiltin": 0,
   "icon": "Web Category Icon:ge100_webcatalog_24",
   "name": "Bucket",
   "id": 8,
   "description": ""
},
        {
   "isbuiltin": 0,
   "icon": "Cloud Category Icon:ge100_virtualfabriccatalog_24",
   "name": "Help Desk",
   "id": 3,
   "description": ""
},
```

Get the details of a single category
Use this REST call to get the details of a single category.

Available HTTP method

Table 43. Get the details of a single category REST API call	
HTTP method	GET
URL pattern	`/resources/automationcategories/8`

Table 43. Get the details of a single category REST API call (continued)

Response	The response of the server contains the specified offering. It has the following set of attributes:
	`{`
	`category:`
	`created:`
	`description:`
	`human_service:`
	`human_service_app_id:`
	`human_service_app_name:`
	`human_service_app_short_name:`
	`icon:`
	`id:`
	`implementation_type:`
	`name:`
	`operation_type:`
	`ownerid:`
	`process:`
	`process_app_id:`
	`process_app_name:`
	`process_app_short_name:`
	`updated:`
	`}`

An entry has the following attributes:

- **category** - optional, the category to which the offering belongs.
- **created** - the creation time of the self-service offering, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.
- **description** - optional, a short description of the offering.
- **human_service** - optional, the URL of an IBM Business Process Manager human service (coach), a user interface to provide user input.
- **human_service_app_id** - optional, depending on the human_service attribute, the ID of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_name** - optional, depends on if the human_service attribute is set, the name of the IBM Business Process Manager application to which the human service belongs.
- **human_service_app_short_name** - the short name of the IBM Business Process Manager human service application.
- **icon** - optional, an offering can have an icon assigned that is displayed inside the Self-Service Catalog.
- **id** - the ID of the offering.
- **implementation_type** - the two possible values are ibm_bpm_process and script.
- **name** - the name of the offering.
- **operation_type** - the only possible value is service.
- **ownerid** - the owner of the user who triggered the offering.
- **process** - the name of the IBM Business Process Manager process that is bound to the offering.
- **process_app_id** - the ID of the IBM Business Process Manager application in which the process is defined.
- **process_app_name** - the name of the IBM Business Process Manager application in which the process is defined.
- **process_app_short_name** - the short name of the IBM Business Process Manager process application.
- **updated** - the time when the self-service offering was last updated, represented as the number of milliseconds since midnight, January 1, 1970 UTC. This value is numeric and is automatically generated by the product.

```
{
     "isbuiltin": 0,
     "icon": "Web Category Icon:ge100_webcatalog_24",
     "name": "Bucket",
     "id": 8,
     "description": ""
  }
```

Update a category
Use this REST call to update a category.

Available HTTP method

Table 44. Update a category REST API call

HTTP method	PUT
URL pattern	`/resources/automationcategories/8`
Response	The self-service offering is updated.

Structure of request body:

```
{  "<attribute>": "<attribute_value>" }
```

```
 {
     "description": "new description of category 8"
  }
```

Delete a category
Use this REST call to delete a category.

Available HTTP method

Table 45. Delete a category REST API call

HTTP method	DELETE
URL pattern	`/resources/automationcategories/8`
Response	The self-service offering is deleted.

Task engine REST API
You can use this set of REST API calls to interact with role parameters of a specific server instance.

Task engine API V1
The following section describes JSON Formats and instances for task engine API V1.

JSON Formats

Task Response

```
{
    "updated_iso" : "2014-03-31T13:05:14+0200",
    "description_message" : "The process is complete.",
    "domain" : "Default",
    "created" : 1396263830875,
    "error" : {
        "resourceBundle" : "com.ibm.orchestrator.messages.orchestratormessages",
        "message" : null,
        "messageKey" : "BPM_PROCESS_COMPLETE",
        "args" :
        [
            "3"
        ]
    },
    "user" : "ksadmin",
    "parm" : {
        .....
    },
    "created_iso" : "2014-03-31T13:03:50+0200",
    "status_localized" : "Completed",
    "error_message" : "CTJCO0002I: Business process instance 3 completed successfully.",
    "status" : "COMPLETED",
    "eventTopic" : "com/ibm/orchestrator/serviceinstance/plan/ibm_bpm_process",
    "delayInSeconds" : 30,
    "project" : "admin",
    "id" : "1003",
    "updated" : 1396263914896,
    "description" : {
        "resourceBundle" : "com.ibm.orchestrator.messages.orchestratormessages",
        "message" : null,
        "messageKey" : "PROCESS_COMPLETE",
        "args" :
        [
        ]
    }
}
```

Tasks Response

[Task Response 1,....., Task Response n]

GET: Get all tasks

URL method
`/kernel/tasks`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Response
Tasks Response

GET: Get the task with a given id

URL method
`/kernel/tasks/{id}`

Accepts
/

Content-Type
application/JSON

Normal Response Codes
200

Response
Task Response

List all currently running and completed tasks
Use this REST API method to list all currently running tasks and completed tasks.

Available HTTP method

Table 46. List all currently running and completed tasks REST API call

HTTP method	GET
URL pattern	`https://hostname/kernel/tasks/`

Table 46. List all currently running and completed tasks REST API call (continued)

<table>
<tr><td>Response</td><td>List all the currently running and completed tasks.<pre>
[<task>]
</pre></td></tr>
<tr><td>Return values</td><td>• 200 - OK
• 500 - Internal Server Error</td></tr>
</table>

Structure of the query string:

- tasks - a comma-separated list of task objects.

Get entries for a specific task
Use this REST call to retrieve information about an active task with an indicated ID.

Available HTTP method

Table 47. Get information about a specific task

<table>
<tr><td>HTTP method</td><td>GET</td></tr>
<tr><td>URL pattern</td><td><code>https://hostname/kernel/tasks/<i>{id}</i></code></td></tr>
<tr><td>Response</td><td>The following parameters of the task are retrieved:<pre>
{
updated_iso:
description_message:
message:
created:
error:
parm: {
       plan:
      }
status_localized:
created_iso:
error_message:
internal status:
status:
eventTopic:
delayInSeconds:
id:
updated:
description: {
       resourceBundle:
       message:
       messageKey:
       args:
             }
}
</pre></td></tr>
<tr><td>Return values</td><td>• 200 - OK
• 401 - The currently logged in user is not authorized to retrieve the task. Only Administrators and creators of the task can see the task.
• 404 – The task does not exist.</td></tr>
</table>

The parameters of the response:

- **`updated_iso`** - the last update to the task in ISO8601 format.
- **`description_message`** - G11N enabled information about the function of the task.
- **`message`** - gives current status, which is displayed to the user.
- **`created`** - the time at which the task was created in java time stamp format.
- **`error`** - a structured object containing an error message if any exists.

- **parm** - a free-form key-value pair object containing all the use-cases specific parameters. It can contain the following parameter:
  - **plan** - contains the self-service offering that was used to create the task. A plan object is only available for tasks that complete self-service offerings.
- **status_localized** - a G11N enabled status for the task.
- **internal status** - one of the following: NEW, QUEUED, RUNNING, SUSPENDED, FAILING, FAILED, COMPLETING, COMPLETED, CANCELING, CANCELED.
- **eventTopic** - identifies the handler that is used to complete the task.
- **description** - contains the internal representation for the **description_message** message.

The following listing shows a sample response that can be obtained through the REST call:

```
{
 "updated_iso" : "2012-08-24T14:06:04+0200",
 "description_message" : "Invoke operation \"Report Problem\" using input parameters entered
                          through a human service.",
 "message" : "The operation was started successfully. For status, check <a
              href=\"/dashboard/appliance/tasks/#e9960f9a-c8cf-499c-8279-73d3a9fbf49e\">the
              task</a> in the task queue."
 "created" : 1345809964681,
 "error" : null,
 "parm" : { "plan" : { "human_service" : "Sample_ReportProblem",
                       "human_service_app_name" : "SCOrchestrator_Toolkit",
                       "implementation_type" : "ibm_bpm_process",
                       "human_service_app_short_name" : "SCOTLKT",
                       "created" : 1345809954675,
                       "process_app_id" : "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
                       "name" : "Report Problem",
                       "ownerid" : 1,
                       "process" : "Sample_AssignProblem",
                       "operation_type" : "service",
                       "human_service_app_id" : "2066.596706e1-2e92-4fb1-a2dd-e0e4bdc4f7fc",
                       "process_app_name" : "SCOrchestrator_Toolkit",
                       "event" : null,
                       "updated" : 1345809954675,
                       "id" : 4,
                       "process_app_short_name" : "SCOTLKT",
                       "description" : "",
                       "category" : "",
                       "apply_to_all_pattern" : 0
                     }
          },
 "status_localized" : "New",
 "created_iso" : "2012-08-24T14:06:04+0200",
 "error_message" : null,
 "status" : "NEW",
 "eventTopic" : "com/ibm/orchestrator/serviceinstance/plan/ibm_bpm_process",
 "delayInSeconds" : 0,
 "id" : "e9960f9a-c8cf-499c-8279-73d3a9fbf49e",
 "updated" : 1345809964685,
 "description" : { "resourceBundle" : "com.ibm.orchestrator.messages.orchestratormessages",
                   "message" : "HS_OPERATION_INVOCATION",
                   "messageKey" : "HS_OPERATION_INVOCATION",
                   "args" : [ "Report Problem" ]
                 }
}
```

Chapter 15. Troubleshooting

Troubleshooting tools have been collected for ease of use when attempting to debug an issue.

Before you begin

You must be assigned the **admin** role to perform these steps.

About this task

The steps to troubleshoot an issue are different for each problem. To help make relevant information available to you as quickly as possible, the log files and other tools for troubleshooting problems have been consolidated together for convenience.

If you are using IBM Cloud Manager with OpenStack in your environment, see Troubleshooting and support for IBM Cloud Manager with OpenStack for more troubleshooting information.

Managing the log information

Use the IBM Cloud Orchestrator log information to troubleshoot the IBM Cloud Orchestrator components.

For information about IBM Cloud Manager with OpenStack log files, see Logging tasks.

Alternate version of file detected

Do not keep alternate file versions under the virtual web server root.

Instead, do the following steps whenever you update the site:

1. Delete or move the files to a directory outside of the virtual root.
2. Edit the files from that directory.
3. Move or copy the files back to the virtual root.

Make sure that only the files that are actually in use reside under the virtual root.

Setting logging levels

Set the logging levels of the IBM Cloud Orchestrator components to increase or decrease the collected troubleshooting information.

When the log data from IBM Cloud Orchestrator components does not provide enough details that are needed to determine the root cause of an error, many of the components have a configurable logging detail setting that you can increase to a debug level.

Note: For some components, the logging level should be changed on a temporary basis only, because log file sizes might increase dramatically when these components are configured to log in debug mode, and the file systems might run out of space.

To increase the logging level of the IBM Cloud Orchestrator components, perform the following procedures on the IBM Cloud Orchestrator Server:

- For Public Cloud Gateway, update the following `-DTRACE_LEVEL` line in `/opt/ibm/ico/wlp/usr/servers/pcg/jvm.options` file:

  ```
  -DTRACE_LEVEL=DEBUG
  ```

  Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

  Restart Public Cloud Gateway with the following command:

  ```
  systemctl restart pcg
  ```

- To enable trace for Self service user interface, see "Enabling trace for the Self-service user interface " on page 363.
- In Business Process Manager, tracing is disabled by default. If you want to troubleshoot or debug any issues, enable tracing. For information about how to enable tracing and how to change the logging level in Business Process Manager, see Configuring trace in the IBM Business Process Manager Knowledge Center.

To increase the logging level for an OpenStack component, complete the following steps.

Important: OpenStack Keystone is a central component that is used very frequently. If you increase the Keystone logging level to debug, the performance of your OpenStack environment will be negatively impacted. After you reproduce the problem and gather the required log information, reverse these changes to return Keystone to normal operating mode.

1. On the OpenStack Controller node, open the configuration file for the component, as listed in the following table:

Component	Configuration files
Keystone	`/etc/keystone/keystone.conf`
Nova	`/etc/nova/nova.conf`
Glance	`/etc/glance/glance-registry.conf` `/etc/glance/glance-api.conf`
Heat	`/etc/heat/heat.conf`
Cinder	`/etc/cinder/cinder.conf`
VMware discovery (VMware only)	`/etc/vmware-discovery.conf`
Neutron	`/etc/neutron/neutron.conf`

2. In the OpenStack component configuration file (in two configuration files for Glance), specify the following values in the `[Default]` section:

```
# Print debugging output (set logging level to DEBUG instead
# of default WARNING level). (boolean value)
#debug=false
debug=true

# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
#verbose=false
verbose=true
```

3. [Debugging Keystone only] Edit the`/etc/keystone/logging.conf` file, and specify the following values in the `[Loggers]` section:

```
###########
# Loggers #
###########

[logger_root]
level=DEBUG
handlers=file
```

Note: To increase the logging level for Keystone, you must edit *both* the `/etc/keystone/keystone.conf` file and the `/etc/keystone/logging.conf` file as described in this section.

4. After you change the logging level of any OpenStack component, you must restart the related OpenStack service. For information about starting the OpenStack services, see your OpenStack documentation. If you are using IBM Cloud Manager with OpenStack, see Managing IBM Cloud Manager with OpenStack services for information about starting the IBM Cloud Manager with OpenStack services.

Important: A Keystone restart must be planned: when you stop Keystone, you disable all IBM Cloud Orchestrator activities that need authorization.

Log file rotation

IBM Cloud Orchestrator uses the `logrotate` mechanism of Linux to manage log file size and rotation settings. For information about adjusting the settings of the log files rotation, see the `logrotate` man page by running the `man logrotate` command. Note that you might be required to run the `logrotate` command by using the `-f` flag after adjusting settings in the configuration files.

The OpenStack Nova, Cinder, Glance, and Keystone log rotation settings are defined in the `/etc/logrotate.d/openstack-*` files on the OpenStack Controller node.

Enabling trace for the Self-service user interface

Update the `log4j.properties` and `server.xml` files to enable trace for the Self-service user interface.

About this task

In this procedure, the example installation directory is `/opt/ibm/ico/wlp/usr/servers/scui`. Replace this value with the appropriate value for your installation.

Procedure

1. Log in to WebSphere Application Server admin console at `https://<ico_server>:9043/ibm/console/logon.jsp` as admin user:
2. Go to **Security** > **Global Security** > **LTPA**.
3. Enter the password to encrypt and decrypt the LPTA key and confirm the password.
4. In **Fully qualified key file name**, enter the full path where you want to export the LTPA key. For example, `/opt/ibm/ico/wlp/usr/servers/scui/LTPA.keys`.
5. Click **Export Keys**.
6. Run the following command to encode both the admin user password and the password that you used when exporting the LTPA key:

   ```
   /opt/ibm/ico/wlp/bin/securityUtility encode <password>
   ```

   For example:

   ```
   # /opt/ibm/ico/wlp/bin/securityUtility encode password
   {xor}Lz4sLCgwLTs=
   ```

7. Edit the `server.xml` file in the `/opt/ibm/ico/wlp/usr/servers/scui` directory and replace the `basicRegistry` tag with the following value:

   ```
   <basicRegistry id="basic" realm="BasicRealm">
   <user name="uid=admin,o=keystone" password="<Admin encoded password>"/>
   </basicRegistry>
   ```

   For example:

   ```
   <basicRegistry id="basic" realm="BasicRealm">
   <user name="uid=admin,o=keystone" password="{xor}Lz4sLCgwLTs="/>
   </basicRegistry>
   ```

8. Add the following line in the `server.xml` file:

   ```
   <ltpa keysFileName="LTPA.keys" keysPassword="<LTPA encoded password>" expiration="120" />
   ```

   For example:

   ```
   <ltpa keysFileName="LTPA.keys" keysPassword="{xor}Lz4sLCgwLTs=" expiration="120" />
   ```

where LTPA.keys is the name of the file where you exported the LTPA key.

9. Change the owner of the LTPA key file:

```
chown scui:scui <LTPA key file>
```

10. Change the following line in the server.xml file:

```
<logging maxFileSize="20" maxFiles="10" traceFileName="${TRACEPATH}"
traceSpecification="n3.app.handler.action.LoginHandler=finest"/>
```

to the following value:

```
<logging maxFileSize="20" maxFiles="10" traceFileName="${TRACEPATH}"
traceSpecification="n3.*=finest:com.ibm.tivoli.*=finest:com.ibm.openstack.api*=fi
nest"
hideMessage="CWWKS4001E" traceFormat="BASIC"/>
```

11. Edit the /opt/ibm/ico/wlp/usr/servers/scui/etc/log4j.properties file and replace all occurrences of INFO with TRACE.
12. Restart the Self-service user interface:

```
service scui restart
```

Finding the log files

To troubleshoot the IBM Cloud Orchestrator components, see the following table to find where the log files are stored in the IBM Cloud Orchestrator Server.

Table 48. Log files on IBM Cloud Orchestrator Server

Component	**Log file default path**
Self-service user interface	/opt/ibm/ico/wlp/usr/servers/scui/logs/scoui.log /opt/ibm/ico/wlp/usr/servers/scui/logs/scoui.trc **Note:** The /opt/ibm/ico/wlp/usr/servers/scui is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.
DB2	Collect the logs by running the following command: `su -db2inst1 -s db2support`
Installer	/var/log/ico_install /var/log/cloud-deployer
IBM HTTP Server	/opt/ibm/ico/HTTPServer/logs
Web Server	/opt/ibm/ico/WebSphere/Plugins/logs
Public Cloud Gateway	/opt/ibm/ico/wlp/usr/servers/pcg/logs/pcg.log **Note:** The /opt/ibm/ico/wlp/usr/servers/pcg is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.
Business Process Manager	/opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/logs

Table 48. Log files on IBM Cloud Orchestrator Server (continued)

Component	Log file default path
Business Process Manager installation	/var/ibm/InstallationManager/logs
IBM Cloud Orchestrator Monitoring	/var/log/ico_monitoring

The following table shows where the OpenStack log files are stored on the OpenStack Controller node or OpenStack compute node by default.

Note: Because the PowerVC node is an OpenStack-based compute node, the related log files are in the default OpenStack directories as specified in the following table.

Table 49. Log files on OpenStack Controller node

Component	Log file default path
OpenStack Dashboard (Horizon) (on the master OpenStack Controller only)	/var/log/httpd
OpenStack	/var/log/nova /var/log/glance /var/log/cinder /var/log/heat /var/log/keystone /var/log/ceilometer /var/log/neutron
IBM Cloud Manager with OpenStack reconfiguration scripts for IBM Cloud Orchestrator	/opt/ico_scripts/

Using the pdcollect tool

Use the pdcollect tool to collect all the log files for debugging IBM Cloud Orchestrator issues that might occur.

About this task

To use the pdcollect tool, run the command /opt/ibm/ico/orchestrator/pdcollect/pdcollect.py on the IBM Cloud Orchestrator Server. It creates the PDcollectlog_*<YYYYMMDDHHMMSS>*_<ICO_Server>.zip file containing log files and other information that can be analyzed and sent to IBM support for problem analysis.

The pdcollect tool for IBM Cloud Orchestrator can also collect log files related to the IBM Cloud Manager with OpenStack environment.

Note: When you run the pdcollect tool for the first time, you are asked to specify the IP address or the fully qualified domain name (FQDN) of the IBM Cloud Manager with OpenStack nodes, and also the components installed on the nodes. This information is stored in the ICM_Environment.json file. You are also asked to specify the password of the IBM Cloud Manager with OpenStack nodes, every time you run the pdcollect tool.

If there is an external database, the pdcollect tool does not collect log files from the external database server. To collect these log files, log in to the external database server as db2inst1 user and run the following command:

```
su -db2inst1 -s db2support
```

Procedure

1. Log on to the IBM Cloud Orchestrator Server.

2. (For the root user only) Change to the directory where the script is located:

```
cd /opt/ibm/ico/orchestrator/pdcollect
```

3. Run the script in one of the following ways:

   - As a root user:

     ```
     ./pdcollect.py [options]
     ```

   - As a nonroot user with sudo permissions:

     ```
     sudo /opt/ibm/ico/orchestrator/pdcollect/pdcollect.py [options]
     ```

     For information about how to create a nonroot user that has the required permissions to run this script, see "Creating a nonroot user to manage the IBM Cloud Orchestrator Server environment" on page 140.

   where the *options* are as follows:

   -h, --help
   Shows this help message and exits.

   -c Components.xml, --componentfile=Components.xml
   Defines the input properties file name. The default name is Components.xml.

   -v Environment.xml, --environmentfile=Environment.xml
   Defines the environment file name. The default name is Environment.xml.

   -o PDCollectlog, --output=PDCollectlog
   Defines the output log file name. The default name is PDCollectlog_*<YYYYMMDDHHMMSS>*_*<ICO_Server>*.zip.

   -n *SYSTEMLIST*, --hostips=*SYSTEMLIST*
   Defines the list of the host IP addresses defined in the environment file to be scanned for log files. The default value is to scan all hosts. The *SYSTEMLIST* format is *hostip1*,*hostip2*,*hostip3*,...

   -p *COMPONENTLIST*, --components=*COMPONENTLIST*
   Lists the components to be scanned for the log files. The *COMPONENTLIST* format is *component1*,*component2*,*component3*,...

   To collect IBM Cloud Manager with OpenStack log files, specify ICM as the component name.

   -s *STARTDATE*, --start=*STARTDATE*
   Defines the first date of the log sequence. The *STARTDATE* format is *YYYY-MM-DD*.

   -e *ENDDATE*, --end=*ENDDATE*
   Defines the day after the last day of the log sequence. The *ENDDATE* format is *YYYY-MM-DD*.

   --version
   Shows the pdcollect tool version and exits.

   Note: A disclaimer is displayed and logged to alert you that the data that is gathered and stored might be confidential and might contain passwords.

Results

The script output is stored in a compressed file. The default output file name is PDCollectlog_*<YYYYMMDDHHMMSS>*_*ICO_Server*.zip.

Known errors and limitations

Check the following sections for information about known errors and limitations in IBM Cloud Orchestrator.

Product limitations

Review the following list of limitations of IBM Cloud Orchestrator.

- The key pairs that are created in IBM Cloud Manager user interface dashboard are not reflected in IBM Cloud Orchestrator offerings during deployment.
- For OpenStack, the service users (for example, nova, `cinder`, `glance`, heat, `ceilometer`) must not be renamed and must be enabled. Also, the service project must not be renamed and must remain enabled. IBM Cloud Orchestrator has more requirements: the `admin` administrator user and the `admin` project must not be renamed or disabled.
- Deploying Windows instances with `cloudbase-init` and multiple NICs might fail because Windows operating system does not honor the order of the devices that are specified at deployment time.
- The following limitations apply to disk resizing in a template with multiple disks:
  - The resize function considers only the boot disk. It does not consider the additional disks or volumes.
  - When deploying on a VMware region a template that has multiple disks, the flavor checks done at deployment time considers the overall space that is needed by all disks and not only the size of the boot disk.
  - When resizing on a VMware region an instance that was deployed from a template with multiple disks, the last disk of the template is resized instead of the boot disk.
- For VMware regions, injecting SSH keys into virtual machines is not supported. When you complete the following OpenStack procedure, the SSH keys are not injected:
  1. Run the **ssh-keygen** command to generate the SSH keys.
  2. Use the **nova** command or the OpenStack Dashboard to add a public key to OpenStack:

     ```
     nova keypair-add --pub_key id_rsa.pub keyname
     ```
  3. Deploy the virtual machine by using the key provided by OpenStack:

     ```
     nova boot --image imageID --key_name keyname --flavor 2 vmName
     ```
  4. You can use this key to access the virtual machine:

     ```
     ssh -i your_private_key user@vm_ip_address
     ```
- In a VMware region, after deploying a virtual machine, if you rename the virtual machine by using vCenter and then you change the flavor by using the Self-service user interface, the name of the virtual machine is set back to the original name.
- PowerVC limitations:
  - The Power NPIV feature requires that all of the hosts in a given system pool have NPIV-capable Fibre Channel adapters.
  - Instances or other resources cannot be moved from one project to another.
  - Each PowerVC region has one availability zone only.
  - Because of an OpenStack limitation, in the OpenStack Dashboard, PowerVC hypervisors are always displayed with an active status. This limitation might cause that IBM Cloud Orchestrator tries to deploy virtual machines to an inactive hypervisor.
  - `cloud-init` is supported starting from PowerVC 1.2.3 and from the following AIX versions:
    - AIX 6.1 TL9 SP5
    - AIX 7.1 TL3 SP5

If you deploy previous versions of AIX images that do not support `cloud-init`, you cannot use password change or ssh key injection functions at deployment time.

  - Shared Storage Pool based images do not support disk resize or extension.
  - Shared Volume Controller FlashCopy® operations can occur only serially per image; if a copy or extending operation is in progress on a boot volume, you cannot start a new operation to make changes to the volume. You can check the Shared Volume Controller UI to see whether a FlashCopy is in progress on the target volume.
  - When attaching a volume to an AIX instance, the **Format volume** option is not supported.
- In a Hyper-V region, resizing an instance makes the instance temporarily invisible in Hyper-V manager. It is readded when the resize operation is completed.
- For z/VM, if you deploy an image to an ECKD disk that is larger than the source ECKD disk, or if you use the OpenStack Dashboard to resize an image to a flavor with a larger disk, the additional space is not usable until you re-partition the disk and resize the file system. If you use a flavor with disk size of 0, the virtual machine is created with the same disk size as the source disk, and therefore avoids the issue. This issue occurs only with ECKD disks.

Security limitations

Check the known security limitations that might expose your IBM Cloud Orchestrator environment to risks.

Toolkit parameters are saved in the log file

The `SCOrchestrator` toolkit saves in the log file all parameters passed by other toolkits used by runbooks. If these parameters contain security sensitive information, they are visible in the `<path_to_BPM_profile>/log/bpm4sco1/SystemOut.log` file

Refer to “Troubleshooting Business Process Manager ” on page 382 for detailed information on this security limitation.

Avoiding data manipulation in Business Process Manager coaches

To prevent manipulation of sensitive data (for example, the domain name) when submitting a Business Process Manager coach in IBM Cloud Orchestrator, when developing a new Business Process Manager human service interface for an offering or action, you must do a sanity check of this sensitive data after the coach has been submitted.

The reason for this sanity check is that all the business objects bound to a Business Process Manager coach are visible in the POST message of the submit request and they could be manipulated by a potential attacker (man-in-the-middle, or logged-in user).

To avoid manipulation, you can store the initially retrieved data in separate variables that are not being exposed to the human interface (coach) itself. After the coach is submitted, the coach data can be validated against these separate variables.

Assuming there is a read-only field for the domain name (for example, `tw.local.exposedField_domainName`) exposed in one of the Business Process Manager coaches visible through an offering, along with this field a separate variable must be created (for example `tw.local.original_domainName`) that holds the domain name that is retrieved at the beginning of the related Business Process Manager workflow (human service). Do not use this variable in any coach. After the coach holding the read-only domain name is submitted, you can check if the domain name has been changed or manipulated by comparing `tw.local.exposedField_domainName` to `tw.local.original_domainName`.

Exposing variables in Business Process Manager business objects

Exposing only one variable which is a parameter of a business object actually exposes the whole business object to the coach.

For example, if you have a coach that displays the name of a virtual machine and the name field is bound to a business object parameter, not only the child variable is exposed to the coach but also the ssh_key variable (which might not be intended). The sample business object tw.local.virtualMachine contains the name and ssh_key parameters. When binding the virtual machine name parameter (tw.local.virtualMachine.name) to the field exposed on the coach, implicitly the whole business object including the ssh_key field is readable in the POST request.

To avoid to expose the whole business object, create an extra variable, for example tw.local.exposedField_VmName, to expose only the virtual machine name in the coach.

Hypervisor errors

You can receive error messages for hypervisors that are defined to IBM Cloud Orchestrator under certain circumstances.

Minus (-) free disk is displayed in OpenStack

Use the following commands to get hypervisor information:

```
 nova hypervisor-list

+----+----------------------+
| ID | Hypervisor hostname  |
+----+----------------------+
| 1  |    computenodeB      |
+----+----------------------+
```

```
 nova hypervisor-show 1

+---------------------------
+-------------------------------------------------------------------------+
| Property                  |
Value                                                                  |
+---------------------------
+-------------------------------------------------------------------------+
| cpu_info_model | ["Intel(R) Xeon(R) CPU X5560 @ 2.80GHz", "Intel(R) Xeon(R) CPU
X5570 @ 2.93GHz"]  |
| cpu_info_topology_cores   |
16                                                                     |
| cpu_info_topology_threads |
32                                                                     |
| cpu_info_vendor           | ["IBM",
"IBM"]                                                         |
| current_workload          |
0                                                                      |
| disk_available_least      |
-                                                                      |
| free_disk_gb              |
25                                                                     |
| free_ram_mb               |
-43934                                                                 |
| host_ip                   |
192.0.2.60                                                             |
| hypervisor_hostname       | domain-c7(HA-
Cluster1)                                                 |
| hypervisor_type           | VMware vCenter
Server                                                  |
| hypervisor_version        |
5001000                                                                |
| id                        |
1                                                                      |
| local_gb                  |
1919                                                                   |
| local_gb_used             |
2171                                                                   |
| memory_mb                 |
89698                                                                  |
| memory_mb_used            |
133632                                                                 |
| running_vms               |
18                                                                     |
| service_host              | vmware-
region1                                                         |
| service_id                |
6                                                                      |
| vcpus                     |
32                                                                     |
| vcpus_used                |
58                                                                     |
+---------------------------
+------------------------------------------------------------------------+
```

`disk_available_least` for hypervisor can be a negative number to indicate the over commitment of hypervisor disk space.

The `qcow2` disk format is used for the virtual machine in the KVM hypervisor, the whole size of disk is not allocated from the beginning to save the disk space, `disk_available_least` comes from the following equation:

```
disk_available_least =  free_disk_gb - disk_overcommit_size
disk_overcommit_size =
    virtual size of disks of all instance instance - used disk size of all instances
```

When the hypervisor instances overcommitted more disk space than free disk space, `disk_available_least` is a negative number.

Minus (-) `free_ram_mb` or `current_workload` is displayed in OpenStack

Use the following command to get hypervisor information:

```
nova hypervisor-show 1

+---------------------------
+------------------------------------------------------------------------+
| Property                  |
Value                                                                  |
+---------------------------
+------------------------------------------------------------------------+
| cpu_info_model | ["Intel(R) Xeon(R) CPU X5560 @ 2.80GHz", "Intel(R) Xeon(R) CPU
X5570 @ 2.93GHz"]  |
| cpu_info_topology_cores   |
16                                                                     |
| cpu_info_topology_threads |
32                                                                     |
| cpu_info_vendor           | ["IBM",
"IBM"]                                                         |
| current_workload          |
0                                                                      |
| disk_available_least      |
-                                                                      |
| free_disk_gb              |
25                                                                     |
| free_ram_mb               |
-43934                                                                 |
| host_ip                   |
192.0.2.60                                                             |
| hypervisor_hostname       | domain-c7(HA-
Cluster1)                                                 |
| hypervisor_type           | VMware vCenter
Server                                                  |
| hypervisor_version        |
5001000                                                                |
| id                        |
1                                                                      |
| local_gb                  |
1919                                                                   |
| local_gb_used             |
2171                                                                   |
| memory_mb                 |
89698                                                                  |
| memory_mb_used            |
133632                                                                  |
| running_vms               |
18                                                                      |
| service_host              | vmware-
region1                                                          |
| service_id                |
6                                                                       |
| vcpus                     |
32                                                                      |
| vcpus_used                |
58                                                                      |
+---------------------------
+-------------------------------------------------------------------------+
```

The `free_ram_mb` for hypervisor can be a negative number to indicate the over commitment of hypervisor memory. The default memory overcommit rate is 1.5 that means you can use memory overall `memory_mb * 1.5` memories. The default cpu overcommit rate is 16 that means you can use memory overall `vcpus * 16 vcpus`.

To configure the overcommit rate, you must modify the following attribute in `nova.conf` and restart the `openstack-nova-scheduler` and the `openstack-nova-compute` services.

```
# virtual CPU to Physical CPU allocation ratio (default: 16.0)
cpu_allocation_ratio=16.0

# virtual ram to physical ram allocation ratio (default: 1.5)
ram_allocation_ratio=1.5
```

High-availability errors

You might encounter errors in an high-availability installation.

Internal error occurs when using the Self-service user interface

During the automated recovery in an high-availability installation, when you are trying to access a page in the Self-service user interface, the following internal error might occur:

```
Ask the administrator to check the SystemOut.log file for more information
to resolve the error (<id of the request>)
```

Causes

The high-availability management uses periodical heartbeats to check for the health of the high-availability cluster. In case of an outage, it triggers an automated recovery. Until this recovery is finished and all services are working correctly again, the error might occur.

Resolving the problem

Log out and log on again to IBM Cloud Orchestrator. Then redo the action.

Troubleshooting IBM Cloud Orchestrator Keystone topology

You might encounter errors in a high-availability keystone topology installation.

IBM Cloud Orchestrator High Availability UI login error

The following internal error might occur when you log into IBM Cloud Orchestrator user interface:

```
CTJCP0027E: An internal error occurred. The requested page cannot be displayed.
Please contact your administrator.
```

Perform the following resolution steps on both primary and secondary nodes:

1. Run the following command in a single line to check whether the virtual IP certificate exists in Self-service user interface keystore:

```
source /etc/profile.d/jdk.sh;keytool -list -v -noprompt -trustcacerts -storepass <password> -
keystore
/opt/ibm/ico/wlp/usr/servers/scui/resources/security/keystore
```

2. If the virtual IP certificate does not exist, then run the following command in a single line to add the certificate to the Self-service user interface keystore:

   ```
   source /etc/profile.d/jdk.sh;keytool -noprompt -import -file /tmp/ico/openstack.crt -
   keystore
   /opt/ibm/ico/wlp/usr/servers/scui/resources/security/keystore -storepass <password> -alias
   <alias>
   ```

3. Restart the HA services.For the procedure to restart, see "Managing the services in a high-availability environment" on page 159.

Instance errors

There are some known errors that might occur when managing instances in IBM Cloud Orchestrator.

Error occurs when deleting an instance

The following rpc error is displayed in the Nova log file when you try to delete an instance.

```
TRACE nova.openstack.common.rpc.amqp
File "/usr/lib/python2.6/site-packages/nova/compute/manager.py", line 923,
in _delete_instance nova.openstack.common.rpc.amqp     instance["uuid"])
TRACE nova.openstack.common.rpc.amqp
File "/usr/lib/python2.6/site-packages/nova/consoleauth/rpcapi.py", line 68,
in delete_tokens_for_instance
TRACE nova.openstack.common.rpc.amqp     instance_uuid=instance_uuid))
TRACE nova.openstack.common.rpc.amqp
File "/usr/lib/python2.6/site-packages/nova/openstack/common/rpc/proxy.py", line 80,
in call
```

Resolving the problem

Install the `openstack-nova-console*.rpm` and then run the following command:

```
/etc/init.d/openstack-nova-consoleauth restart
```

.

Unable to add disk to SLES instance

SLES hotplug of virtual disk is not fully supported.

Symptoms

The `Default add disk` add-on failed or the device requested in the `Default raw disk` add-on is not present in the `fdisk -l` output.

Resolving the problem

Manually request a reboot from the virtual machine or stop and restart it from the user interface, and click **Execute Now** in the virtual machine script packages section.

Unable to change the flavor of VMware virtual machines

Attempts to change the flavor of VMware virtual machines by using OpenStack fail with an error.

Symptoms

If you try to change the flavor of VMware virtual machines by using OpenStack, the OpenStack server object is in an error state. The virtual machine itself is not affected.

Resolving the problem

You must verify the following settings for `/etc/nova/nova.conf`:

- If there is only one compute node, set **allow_resize_to_same_host** to true.
- Set **multi_host** to false.

You must also make sure that the virtual machine uses an **SCSI** disk.

Unable to correctly display virtual machines

If you disabled the VNC in the nova.conf file by setting vnc_enable=false, some virtual machines might not display correctly.

Symptoms

You can see a message like boot from harddisk for the instances in the console log.

Causes

The problem occurs because OpenStack does not prepare the graphic device for the instance if the VNC is disabled. Therefore, if the virtual machine depends on the graphic during the boot time, it hangs and it does not boot up.

Resolving the problem

Enable the VNC by setting vnc_enable=true in the nova.conf file.

Unable to deploy a Heat stack after migration

After migrating from an IBM Cloud Orchestrator V2.4.0.2 environment, Heat stack deployment might time out or fail.

Resolving the problem

Perform the following steps:

1. Restart the Heat services by running the following commands:

   ```
   systemctl restart openstack-heat-engine
   systemctl restart openstack-heat-api
   ```

2. Delete the failed deployment and deploy the Heat stack again.

Unable to deploy an instance with No valid host was found error

When you deploy an instance, an error occurs. If you run nova show <instance name>, the No Valid host was found error is displayed.

Causes

This problem occurs because the Nova scheduler is not able to find a hypervisor to provision the virtual machine. The problem usually occurs for the following reasons:

- You do not clean unused virtual machine for a long time.
- You are provisioning more virtual machines than the cloud capacity.
- You disable or remove hypervisor from cloud like, for example, you remove an ESXi host from VMware vCenter, or a KVM compute node is offline.

To understand if the cloud is out of capacity, you can use the **nova hypervisor-show <hypervisor id>** command to check CPU, memory, and disk that are used by the virtual machines on each hypervisor.

Resolving the problem

To debug the problem, you can increase the logging level of the Nova component. For more information, see “Setting logging levels” on page 361.

Restart the openstack-nova-scheduler and after a failure, you might see the following messages in the /var/log/nova/scheduler.log file:

```
2014-09-16 10:15:37.672 28695 DEBUG nova.scheduler.filters.ram_filter
[req-ef6b203c-e0fa-4ba0-8f9f-b2d71ca3deb9 e417380bc10b48c1b5fb4296d6fa470d
```

```
26e0b2a9767848f88cd62d7580682bf0] (cil017110014, domain-c19(cluster110))
ram:-42019 disk:267230208 io_ops:0 instances:24 does not have 2048 MB usable ram,
it only has -27444.5 MB usable ram. host_passes
/usr/lib/python2.6/site-packages/nova/scheduler/filters/ram_filter.py:60
2014-09-16 10:15:37.673 28695 INFO nova.filters
[req-ef6b203c-e0fa-4ba0-8f9f-b2d71ca3deb9 e417380bc10b48c1b5fb4296d6fa470d
26e0b2a9767848f88cd62d7580682bf0] Filter RamFilter returned 0 hosts
2014-09-16 10:15:37.673 28695 WARNING nova.scheduler.driver
[req-ef6b203c-e0fa-4ba0-8f9f-b2d71ca3deb9 e417380bc10b48c1b5fb4296d6fa470d
26e0b2a9767848f88cd62d7580682bf0] [instance: a4086e77-389e-4eb8-9431-de5eb6da07f0]
 Setting instance to ERROR state.
```

The errors indicate that the RAM is not enough to host one more virtual machine with 2 GB required.

Unable to reach a deployed virtual machine

You cannot reach a virtual machine after deployment.

Causes

In the `/etc/sysconfig/network` file, the `GATEWAYDEV=<some-if>` statement might cause the deployed virtual machine routing table to be set incorrectly and therefore the virtual machine might result unreachable.

Resolving the problem

Comment or remove the `GATEWAYDEV=<some-if>` statement.

Unable to reach one of the addresses if multiple NICs of a Linux virtual machine are deployed

The router of the second IP address is not set in the virtual machine automatically.

Causes

The problem occurs because in Linux there is only one default gateway, which means that even if the network packet can reach the second NIC, the response packet still uses the default gateway. At that point, the response packet is not able to reach the sender.

Resolving the problem

Manually add another routing table by performing the following steps:

1. Determine which is the default gateway and which NIC needs to add an additional route table. Run the command:

   ```
   ip addr show
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet6 ::1/128 scope host
   valid_lft forever preferred_lft forever

   2: eth0
   <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1454 qdisc pfifo_fast state UP qlen 1000
   link/ether fa:16:3e:cd:c3:17 brd ff:ff:ff:ff:ff:ff
   inet 192.0.1.145/24 brd 192.0.1.255 scope global eth0
   inet6 fe80::f816:3eff:fecd:c317/64 scope link
   valid_lft forever preferred_lft forever
   eth1:
   <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1454 qdisc pfifo_fast state UP qlen 1000
   link/ether fa:16:3e:f8:a4:f2 brd ff:ff:ff:ff:ff:ff
   inet 192.0.2.4/24 brd 192.0.2.255 scope global eth1
   inet6 fe80::f816:3eff:fef8:a4f2/64 scope link
   valid_lft forever preferred_lft forever
   ```

   Now, the virtual machine has two NICs: `eth0` has `192.0.1.145`, `eth1` has `192.0.2.4`.

   Check the route table:

   ```
   route -n

    Kernel IP routing table  Destination Gateway  Genmask          Flags Metric Ref
   ```

```
Use Iface
 169.254.169.254          192.0.2.3            255.255.255.255  UGH   0       0
0   eth1
 192.0.1.0                0.0.0.0              255.255.255.0    U     0       0
0   eth0
 192.0.2.0                0.0.0.0              255.255.255.0    U     0       0
0   eth1
 169.254.0.0              0.0.0.0              255.255.0.0      U     1002    0
0   eth0
 169.254.0.0              0.0.0.0              255.255.0.0      U     1003    0
0   eth1
 192.0.2.1                0.0.0.0                               UG    0       0
0   eth1
```

so the NIC `eth1` has a default gateway that can be reached from the outside, while `eth0` does not have a gateway so it cannot be reached from other networks.

2. You must add another route table for `eth0`. Use following command (`eth0` is the name of the route table or you can provide your own meaningful name):

```
echo "1 eth0" >> /etc/iproute2/rt_tables
```

3. Configure the routing rules for table `eth0`:

```
ip route add 192.0.1.0/24 dev eth0 src 192.0.1.145 table eth0
ip route add default via 192.0.1.1 dev eth0 table eth0
ip rule add from 192.0.1.145/32 table eth0
ip rule add to 192.0.1.145/32 table eth0
```

Unable to start a virtual system

The `resource could not be found` message might be displayed when a user from a newly created project tries to start a virtual system.

Symptoms

An example of this situation is when you run the following command to run a virtual machine from OpenStack:

```
nova boot --image rhelova --flavor m1.large --nic net-id=e325a701-ab07-4fb9-a7df-621e0eb31c9b
test1vm2
```

The following message is displayed:

```
ERROR: The resource could not be found. (HTTP 404) (Request-ID:
req-4d801615-5fb3-49fe-8830-860de3f3f7db)
```

Causes

The reason for the error is that the user cannot access the network that is defined in the environment profile for the virtual system. This problem occurs when a user who is associated with one project (for example, Project A) tries to access a network that is associated with another project (for example, Project B).

The problem can be caused because of the following issues:

- The network was manually associated with Project B.
- The network was automatically associated with Project B when an admin user from Project B booted a virtual machine from OpenStack without specifying the network:
  - Using the API: Omitted the **`networks`** parameter
  - Using the CLI: Omitted the **`--nic`** parameter

  If the network is not specified in the boot command, and if the project of the current user is not associated with any network, OpenStack associates the network with the project of the current user: in the example, Project B.

If a network is associated with a project, that network cannot be accessed by users from any other project.

Resolving the problem

To resolve this problem, you must associate the network with the appropriate project to ensure that the users assigned to that project can access that network.

You can associate a network with only one project. If you want users from multiple projects to access the same network, you must disassociate all projects from the network, which sets the `project_id` value for the network to None.

General errors

There are some known errors that might occur when using IBM Cloud Orchestrator.

IBM Cloud Orchestrator UI login failure with browser language switch

If no available zones are assigned to the current project and you access the IBM Cloud Orchestrator UI with the browser language set to Russian (ru), then the UI fails with an error message.

Error message:

```
The server encountered an internal error or misconfiguration and was unable to complete your
request.
Please contact the server administrator, you@your.address and inform them of the time the error
occurred, and anything you might have done that may have caused the error.
More information about this error may be available in the server error log.  IBM_HTTP_Server at
ico.cn.ibm.com Port 443
```

Ensure that the available zones are assigned to the current project and then change the browser language setting to Russian.

SCOrchestrator.py fails for defusedxml package

Troubleshooting details for SCOrchestrator.py whenever it fails for defusedxml package

If the setup is not pointing to the repo that has python package, the following error occurs:

```
./SCOrchestrator.py --status
Traceback (most recent call last):
  File "./SCOrchestrator.py", line 12, in <module>
    import defusedxml.ElementTree as xml
ImportError: No module named defusedxml.ElementTree
```

As a resolution, update the repos and run the below commands to install defusedxml:

```
python /usr/bin/easy_install pip

pip install defusedxml
```

Users with member permission cannot see migrated IP details

User with member role cannot see the IP details of on-boarded virtual machines.

To resolve the issue, do the following steps:

1. Create an offering in Business Process Manager to provision a virtual machine with the option of selecting IP address.
2. After the virtual machine gets provisioned, replace the `vmdk` file of the discovered virtual machine with the `vmdk` file of the IBM Cloud Orchestrator provisioned virtual machine.
3. Restart the IBM Cloud Orchestrator provisioned virtual machine.

Errors in ICM_configure_ico.sh and ./ICM_configure_ico_horizon_extensions.sh scripts

Use this section to solve known issues that might occur when you run **ICM_configure_ico.sh** and **./ICM_configure_ico_horizon_extensions.sh** scripts.

- Errors might occur whenever you run **ICM_configure_ico.sh** script post the update of HTTPS on IBM Cloud Manager with OpenStack. Go to `<location where scripts are copied>/ICO-reconfigure_<timestamp>.log` to analyze the error. Sometimes, the error might get resolved automatically if you rerun the **ICM_configure_ico.sh** script. If script rerun does not resolve the error, contact IBM support.
- Whenever you run `ICM_configure_script.sh` on Red Hat Enterprise Linux 7.4, the script fails with the following error, which is associated to GnomeKeyring:

  ```
  Log of the reconfiguration is stored in
  /rpms/scripts2505/ICO-reconfigure_20171122-0418.log
  sourcing openrc
  ICM_config_ico.sh: Configure master ICM controller for ICO
  Check ico extension file exists
  /rpms/scripts2505/sco_horizon.zip exists
  Stopping OpenStack services
  Creating ICO users, roles and assignments
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6:
  PyGIWarning: GnomeKeyring was imported without specifying a version first.
  Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  openstack 1.0.4
  /rpms/scripts2505/script_functions.sh: line 98: [: PyGIWarning:: integer expression expected
  /rpms/scripts2505/script_functions.sh: line 102: PyGIWarning:: syntax error in expression (error token is
  ":")
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.
  from gi.repository import GnomeKeyring
  /usr/lib/python2.7/site-packages/keyring/backends/Gnome.py:6: PyGIWarning: GnomeKeyring was imported
  without specifying a version first. Use gi.require_version('GnomeKeyring', '1.0')
  before import to ensure that the right version gets loaded.2:55:05 PM
  ```

  As a resolution, upgrade the package python keyring. Alternatively, you can also run the local fix that is provided at https://bugzilla.redhat.com/show_bug.cgi?id=1259747.

Errors in provisioning virtual machines

The provisioning of virtual machines from IBM Cloud Orchestrator can fail with an error message.

For all such provisioning failures, go to IBM Cloud Manager with OpenStack user interface and delete the virtual machine. For the steps to delete the virtual machine, see IBM Cloud Manager with OpenStack knowledge center at https://www.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_kc_welcome.html.

Issues with Internet Explorer 11 browser

The general known issues that might occur when you use Internet Explorer 11 browser.

- Once you log off from the IBM Cloud Orchestrator Self service UI, the logon panel does not appear on the screen. If you refresh the browser with the Self service user interface URL, an "HTTP 500 internal error" occurs. If you log off and want to relogin, then close and reopen the Internet Explorer browser.
- If a logged in user is defined in different projects and you switch (change) the project in IBM Cloud Orchestrator UI, you might experience problems in using the user interface. Use another browser (Mozilla Firefox or Google Chrome) to resolve this issue.

Member role not able to use Self service Catalog

Errors might occur when you submit the service coach from **Self service Catalog** of IBM Cloud Orchestrator user interface as a user with member role.

As an administrator, check the `SystemOut.log` file for more information to resolve the error. The following errors get recorded in the `SystemOut.log` file:

- `AjaxControlle E CWLLG0010E: HTTP Session not found, unexpected errors will occur as a result.`
- `ControllerServlet. Error: The following session is not valid! pY85Qb26uYcxWAEjQldQfqA`
- `00000156 AjaxControlle E com.lombardisoftware.servlet.AjaxControllerServlet handleError Logging original exception`

As a resolution, do the following steps:

1. Log off from the current session and close the browser.
2. Open new session with new browser instance.

Microsoft Active Directory integration errors

There exists a IBM Cloud Orchestrator UI authentication limitation whenever the OpenStack is integrated with Microsoft Active Directory.

When you log in to IBM Cloud Orchestrator UI, remember to enter the user ID in the same character case as of Microsoft Active Directory. If the case is different, then you might observe issues while you work with IBM Cloud Orchestrator features.

UAC setting issues in Windows virtual machine

If the UAC setting is enabled in a Windows virtual machine, a business process instance error might occur whenever you run an IBM Cloud Orchestrator offering to attach volume with the virtual machine.

To disable the UAC setting in windows virtual machine, change the system policy **`EnableLUA`** from `1` to `0`. For the steps to deactivate, see your Windows operating system documentation.

32-bit library files were not found

In the `db2prereqcheck.log` file, the following errors are not critical and can be ignored.

```
Validating "32 bit version of "libstdc++.so.6" " ...
Found the 64 bit "/usr/lib64/libstdc++.so.6" in the following directory "/usr/lib64".
DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library file:
 "libstdc++.so.6".
Validating "/lib/libpam.so*" ...
DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library file:
 "/lib/libpam.so*".
WARNING : Requirement not matched.
Requirement not matched for DB2 database "Server" . Version: "10.5.0.2".
Summary of prerequisites that are not met on the current system:
DBT3514W  The db2prereqcheck utility failed to find the following 32-bit library file:
 "/lib/libpam.so*".
```

Empty quota values on a domain in the OpenStack Dashboard

When you edit a domain in the OpenStack Dashboard, some of the fields in the quota tab are empty.

Causes

This problem might be caused by creating a domain using the CLI or API and not assigning a project with the name `Default`. The IBM Cloud Orchestrator Horizon extension implicitly creates a `Default` project when a domain is created.

Resolving the problem

Create a **Default** project for the domain.

Internal error occurs when using the Self-service user interface

When you are trying to access a page in the Self-service user interface, the following internal error might occur.

```
CTJCP0027E: An internal error occurred. The requested page cannot be displayed.
Please contact your administrator.
```

Resolving the problem

If the same problem occurs on different pages or for different users, check if all the IBM Cloud Orchestrator services are running by using the following command:

```
SCOrchestrator.py -status
```

Also check the status of the OpenStack services and middleware. This might also be caused by any services in error in the underlying OpenStack installation.

For IBM Cloud Manager with OpenStack, see Checking status of OpenStack services.

For a generic OpenStack distribution, see the documentation of your distribution.

You might also find additional information about the error in the Self-service user interface log file. For information about log files, see “Finding the log files” on page 364.

Unable to list all the existing resources

The number of items returned in a single response from resources, like virtual machines or volumes, is limited to 1000.

Causes

The API or user interface only lists 1000 resources. This is an intentional limit as a larger result sets require greater cost to derive and manage.

Resolving the problem

If you want to see a larger result sets, increase the maximum number of instances returned in a single response by setting the `osapi_max_limit` property in the `/etc/nova/nova.conf` file in the Compute Nodes of the related region. This setting impacts several interfaces, including the Nova `list` interface, the OpenStack Dashboard, and the Self-service user interface. To manage all instances in a region, the recommended setting is the maximum number of instances for the region and a growth buffer. For example, if the region can contain 2000 instances, and you want a 10% growth buffer, use a limit of 2200 instances.

Unable to retrieve availability zone data after migration

After migrating from an IBM Cloud Orchestrator 2.4.0.2 environment, when editing a domain or a project, the following error is displayed in the OpenStack Dashboard.

```
Error: Unable to retrieve availability zone data.
```

In the httpd log file, the following error is displayed:

```
==> openstack-dashboard-error.log <==
[Tue Dec 08 18:37:29.258353 2015] [:error] [pid 9343] Recoverable error:
('Connection aborted.', error(113, 'No route to host'))
```

Resolving the problem

Restart the pcg and httpd services.

Errors when you run prereq-checker.sh

Errors might occur whenever you run prereq-checker.sh.

To resolve the error, check the error type and take appropriate action:

- The following error appears in prereq-checker.sh when you do a fresh installation of IBM Cloud Orchestrator 2.5.0.9 or when you upgrade your operating system before IBM Cloud Orchestrator upgrade.

  ```
  - Message text: Some prerequisite packages are missing.
  The following prerequisite packages are missing: libibcm
  - User response: No action required. The missing prerequisite packages will be installed
  automatically by the installation program.
  ```

  To resolve this issue, configure both RHEL 7.5 and RHEL 7.6 repos.

- If the following error occurs, verify whether the SIMPLE_TOKEN_SECRET value is valid. If the error persists even if the SIMPLE_TOKEN_SECRET is valid, then the issue might be because of time lag. Synchronize the time difference between IBM Cloud Orchestrator and IBM Cloud Manager with OpenStack:

  ```
  Checking that the parameter SIMPLE_TOKEN_SECRET can be used
  to authenticate with <ICM_server>
  Status: Failed
  Message text: The value of SIMPLE_TOKEN_SECRET must be correct for the host <ICM_server>
  ERROR: 401
  Message: KS-58299FC The request you have made requires authentication.
  SIMPLE_TOKEN_SECRET is not valid
  User response: Check that the value of the parameter SIMPLE_TOKEN_SECRET is correct.
  ```

- If the following error occurs, then check whether the **PROTOCOL** value in the ico_install.rsp file is compliant with the OpenStack instance configuration:

  ```
  Checking that the parameter SIMPLE_TOKEN_SECRET can be used to
  authenticate with <openstack_server> -
  Status: Failed   -  Message text: The value of SIMPLE_TOKEN_SECRET must be correct
  for the host <openstack_server>
  ERROR: Could not send authentication request to <openstack_server>   -  User
  response: Check that the value of the parameter OPENSTACK_HOST_NAME is correct
  ```

  The value of **PROTOCOL** must be HTTPS only if the OpenStack is configured with HTTPS. For **PROTOCOL** details, see *Mandatory deployment parameters* table of "Setting the deployment parameters" on page 60.

Error occurs when attaching a volume to an existing Windows virtual machine

The following error is displayed in the Business Process Manager logs when attaching and formatting a volume to an existing Microsoft Windows virtual machine.

```
Virtual Disk Service error: The operation is not allowed on a disk that is offline. Problem is
with the DISKPART configuration of the windows 2012 image.
```

Resolving the problem

Change the virtual machine policy by using the following command:

```
diskpart> san policy=OnlineAll
```

For more information, see the following link: https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2000767

Troubleshooting a high-availability environment

This topic describes how to troubleshoot problems that might occur when running a high-availability environment of IBM Cloud Orchestrator.

Troubleshooting IBM System Automation for Multiplatforms controlled services

To provide high availability, the services on the IBM Cloud Orchestrator server are clustered on two virtual machines and are controlled by IBM System Automation for Multiplatforms. A service might be in an error state. You can check the state of these clusters in the following ways:

- Log on to the command line of a cluster virtual machine and run the **lssam** command. The following is a sample output:

```
Online IBM.ResourceGroup:central-services-rg Nominal=Online
                      |- Online IBM.Application:bpm-node
                              |- Online IBM.Application:bpm-node:primaryiconode
                              '- Online IBM.Application:bpm-node:secondaryiconode
                      |- Online IBM.Application:bpm
                              |- Online IBM.Application:bpm:primaryiconode
                              '- Online IBM.Application:bpm:secondaryiconode
                      |- Online IBM.Application:ihs
                              |- Offline IBM.Application:ihs:primaryiconode
                              '- Online IBM.Application:ihs:secondaryiconode
                      |- Online IBM.Application:scui
                              |- Online IBM.Application:scui:primaryiconode
                              '- Online IBM.Application:scui:secondaryiconode
                      '- Online IBM.ServiceIP:cs-ip
                              |- Offline IBM.ServiceIP:cs-ip:primaryiconode
                              '- Online IBM.ServiceIP:cs-ip:secondaryiconode
              Online IBM.ResourceGroup:pcg-rg Nominal=Online
                      '- Online IBM.Application:pcg
                              '- Online IBM.Application:pcg:primaryiconode
              Online IBM.Equivalency:cs-network-equ
                      |- Online IBM.NetworkInterface:ens192:primaryiconode
                      '- Online IBM.NetworkInterface:ens192:secondaryiconode
```

If a resource is not in the correct state (either offline or online), check the actual state of the service:

1. Log on, as a `root` user, to the virtual machine where the service runs. The host name of the VM is listed the **lssam** output as property of the resource.
2. Run the following command:

   ```
   systemctl status <servicename>
   ```

   Tip: The service registered in `systemctl` might differ from the name of the application resource in IBM System Automation for Multiplatforms. The output should match the state in the **lssam** command output.
3. If a service is in an error state or in unknown state, try to manually recover the service by running the following command on the virtual machine where the service should run:

   ```
   systemctl stop <service>; systemctl start <service>
   ```

4. If the service does not start, check the log files of the service and correct any problems found there.
5. You might need to suspend the automation to resolve an error condition. In a normal state, the automation automatically restarts a service even if the service is stopped manually. To suspend the automation, log on to one of the cluster virtual machines and run the following command:

   ```
   samctrl -M t
   ```

   After the error condition is resolved, start the service and verify the correct status. If the status is correct, run the following command to resume the automation:

   ```
   samctrl -M f
   ```

6. If the resource is still in error, run the following command to reset the resource in IBM System Automation for Multiplatforms:

   ```
   resetrsrc -s 'Name == "<service>"' IBM.Application
   ```

Managing the placement of services

During a system failure, certain services are moved in the cluster. To identify where a service is running, use the **lssam** command. If a service is configured as active/standby, the command output shows that the server where the service is running is online, and the standby server is offline. For maintenance or certain recoveries, you might need to manually move a service in the cluster. You can move a service by running the following command on one of the servers in the cluster:

```
rgreq -n <fqdn_of_the_node_to_be_moved_from> -o move
<name_of_resource_group_for_service>
```

Troubleshooting Business Process Manager

Use this topic to solve known issues that might occur when using Business Process Manager.

Log files are stored in the `/opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/logs` directory.

In Business Process Manager, tracing is disabled by default. If you want to troubleshoot or debug any issues, enable tracing. For information about how to enable tracing and how to change the logging level in Business Process Manager, see Configuring trace in the IBM Business Process Manager Knowledge Center.

Synchronizing security changes on WebSphere Application Server nodes

The security-related changes like certificate changes that are made from the WebSphere Application Server Administrative Console takes time to reflect in all the WebSphere Application Server nodes.

For any synchronization errors due to this delay, see https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/rxml_syncnode.html to run commands on your operating system.

Visible security-sensitive information

The `SCOrchestrator` toolkit identifies all parameters that are passed by other toolkits that are used by runbooks, and saves the parameters in the `SystemOut.log` file. If these parameters contain security-sensitive information, that security-sensitive information is visible in the `SystemOut.log` file. The `SystemOut.log` file is in the *path_to_BPM_profile*`/logs/SingleClusterMember1` directory (for example, `/opt/ibm/ico/BPM/v8.5/profiles/Node1Profile/logs/SingleClusterMember1/SystemOut.log`).

To disable logging in the `SCOrchestrator` toolkit, complete the following steps on the IBM Cloud Orchestrator Server:

1. Create a script file that is named `disableLogging.py` with the following content:

   ```
   AdminTask.setTraceSpecification('[-persist true -traceSpecification
   *=info:WLE.wle_javascript=audit ]');
   AdminConfig.save();
   ```

2. From the directory where you saved the `disableLogging.py` file, run the following command:

   ```
   path_to_BPM_profile/bin/wsadmin.sh  -host `uname -n` -username admin
             -password admin_password -f disableLogging.py
   ```

To re-enable logging in the `SCOrchestrator` toolkit, complete the following steps on the IBM Cloud Orchestrator Server:

1. Create a script file that is named `enableLogging.py` with the following content:

   ```
   AdminTask.setTraceSpecification('[-persist true -traceSpecification *=info ]');
   AdminConfig.save();
   ```

2. From the directory where you saved the `enableLogging.py` file, run the following command:

   ```
   path_to_BPM_profile/bin/wsadmin.sh -host `uname -n` -username admin
       -password admin_password -f enableLogging.py
   ```

Database size increases continuously

The size of the Performance Data Warehouse database increases continuously (greater than 100 GB) when IBM Cloud Orchestrator is running, and eventually results in out-of-space errors.

The database size increases because error messages similar to the following message are written to the `LSW_DATA_TRANSFER_ERRORS` table in the Performance Data Warehouse database:

```
undefined tracking group with external ID 5cde1ff9-d1ab-4a88-8810-b0e7dcbe571e
```

To resolve the problem, perform the following steps:

1. Open the Business Process Manager Process Designer.
2. Click the **Toolkits** tab, and select the following toolkit:

   ```
   SCOrchestrator_Support_vSys_Toolkit (SCOVSYS)
   ```

3. Click **File > Update Tracking Definitions**.

Troubleshooting the Public Cloud Gateway

Use this section to solve known issues that might occur when you use Public Cloud Gateway.

Public Cloud Gateway server related log messages are stored in the `/opt/ibm/ico/wlp/usr/servers/pcg/logs/pcg.log` file.

Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.

For Public Cloud Gateway, update the following `-DTRACE_LEVEL` line in `/opt/ibm/ico/wlp/usr/servers/pcg/jvm.options` file:

```
-DTRACE_LEVEL=DEBUG
```

Restart Public Cloud Gateway with the following command:

```
systemctl restart pcg
```

Modify Availability Zone Action of domains

If you have non-accessible Public Cloud Gateway regions in IBM® Cloud Orchestrator, then an error might occur whenever you modify Availability Zones' of domains.

As a solution, delete the old non-working Public Cloud Gateway regions before you continue to manage availability zones for domains.

Debugging image templates

When you build new images from the base images available in the remote clouds, you must take some precautions to allow the images to support login through a user ID and a password.

Symptoms

The provisioned virtual machine from an image template is not accessible through the specified credentials.

Causes

This problem might happen for the following reasons:

- `cloud-init` or `cloudbase-init` is not installed.
- `cloud-init` or `cloudbase-init` is not configured for root and password login.
- The remote clouds have strict default password rules and the password that is provided in the DeploySingleServer offering is not compliant with the rules in the image.

Resolving the problem

Perform the following steps:

1. Add a debug user to the image so that you can access the image after provisioning if problems occur. Remove the debug user in the final version of the image.
2. Make sure that you performed all the image setup steps for the deployment scenario that you chose.
3. Check the password rules that are active on the base image and ensure that the provided password is compliant.
4. Make sure that `cloud-init` or `cloudbase-init` is installed and configured, and that the Public Cloud Gateway provided extensions are installed.

Deploying an instance with an additional disk to SoftLayer fails due to timeout

Deploying an instance with an additional disk to SoftLayer fails with error `java.net.SocketTimeoutException: Read timed out`.

Symptoms

When a new instance is deployed with an additional disk, Public Cloud Gateway waits until the virtual machine is running before it attaches the disk. Depending on the load of the cloud, this might take up to several minutes on SoftLayer, during which the provisioning workflow might run in a timeout.

Resolving the problem

You can increase the socket timeout by running the following steps:

1. Log in to WebSphere console as `bpm_admin` at `https://<host>:9043/admin`.
2. Navigate to **Servers > All Servers > SingleClusterMember1 > Server Infrastructure >Java and Process Management >Process definition >Additional Properties > Java Virtual Machine > Custom properties**.
3. Add new property: `vmm_OpenStack_SocketTimeout` and set it to `3600000` (1 hour).
4. Save and restart the Business Process Manager server by running the following command:

   ```
   systemctl restart bpm
   ```

Failure to generate admin token

The Public Cloud Gateway fails with `Unable generate admin token` error.

Symptoms

The Public Cloud Gateway startup fails with `Unable generate admin token` and a HybridUnauthorizedException errors.

During startup of Public Cloud Gateway an admin token is generated based on the following configuration information in the etc directory of the Public Cloud Gateway:

```
* admin.json    :
* config.json
```

Admin.json content:

```
{
    "auth":{
        "passwordCredentials":{
            "username":"xxxx",
            "password":"yyyyy"
        },
        "tenantName":"zzzz",
        "domainName": "ddddd"
    }
}
```

The username must be a user ID that has admin rights. The password must be encrypted through the encryptPassword.sh. The tenantName must be set to the tenant name of the admin user. The domainName is optional and defaults to the Default domain. Set domainName to the domain of the admin user.

Note: Required if the user is in a non-default domain.

Resolving the problem

Make sure the values in admin.json match to your admin user ID in the system.

Config.json content:

```
"auth":{
        "provider":"keystone",
        "service_url":"http://KeystoneHost:5000",
        "admin_url":"http://KeystoneHost:35357"
    }
```

If you manually changed the content of the service_url or the admin_url, the admin token cannot be created.

Make sure that the KeystoneHost is set to the host name where keystone is installed in your IBM Cloud Orchestrator environment. During installation the values are configured based on your topology selection.

Loss of functionality in Public Cloud Gateway cloud groups

Loss of functionality might occur in Public Cloud Gateway cloud groups in IBM Cloud Orchestrator, where there has been heavy load on the Public Cloud Gateway cloud groups.

Causes

This problem is due to exceeding the Amazon Web Service (AWS) API Request Rate limit. The Public Cloud Gateway has addressed this issue by introducing a caching mechanism.

Resolving the problem

See “Configuring caching” on page 267.

Problem configuring privateNetworkOnly on Amazon EC2 subnets

The following problem might occur configuring privateNetworkOnly on Amazon EC2 subnets. Amazon provided a new out of the box feature to auto assign public IP addresses on a subnet base. This feature allows to control on a subnet base if a public IP address must be assigned. The feature is available for default and non-default VPC as subnets are assigned through the VPC and availability zone.

Symptoms

The problem occurs configuring privateNetworkOnly according to what documented at "Configuring subnets and security groups in a non-default VPC region" on page 276.

During provisioning the result is not as expected, whether a public IP address is assigned or not.

Causes

Amazon added new support for auto-assign public IP address on the subnet configuration for default and non-default VPC.

If you see a **Modify Auto-Assign Public IP** button on the subnets page in your VPC Dashboard, you might face the problem.

If **Auto-assign Public IP** is set to yes, this setting takes precedence over what you configured in Public Cloud Gateway.

Resolving the problem

The **Auto-assign Public IP** flag on the subnet that is used for provisioning must be set to no for default or non-default VPC.

Setting the **Auto-assign Public IP** flag on the subnet to yes has precedence over the Public Cloud Gateway related configuration.

Related concepts

"Configuring subnets and security groups in a non-default VPC region" on page 276
You can configure subnets and security groups in a non-default VPC region.

Quota troubleshooting

Resolve any quota issues that you encounter when using the Public Cloud Gateway.

Default quota is not defined large enough

Without any customization, a default quota definition exists in the config.json. There are situations in which this default quota definition is too small.

To resolve the problem either create a project level quota in the **Quota** tab of the **Project** page in the OpenStack Dashboard or increase the default quota definition in config.json.

Project quota definition is too small

If a project level quota definition exists, the values can only be changed in the **Quota** tab of the **Project** page in the OpenStack Dashboard.

Existing virtual machine instances already consume more resources than the quota allows

To resolve the problem, count the number of instances, cores, RAM, and volume usage and update the corresponding quota values. Volumes is the sum of the VM instance volume and the additional disks. There is a gigabytes value in the quotas that defines the largest possible virtual machine instance possible. It is possible that you have reached this limit.

Too many key-pairs already exist

Key pairs are stored on a per project base post-fixed with the project ID. To resolve the problem, sum up all the key pairs that have the same project ID and adjust the quota definition for key pairs accordingly.

Too much storage is already consumed than is defined in the quota

Volumes are the sum of the virtual machine instance volume and the additional disks. To resolve the problem, adjust the quota definition for volumes accordingly

Provisioning failed even though quota has not been reached yet

There might be situation where the capacity of the region (EC2 or SoftLayer) is already exhausted before the defined quotas.

To resolve the problem:

- Check whether you have set the quota for the region and projects higher than the capacity of the region. Either lower the quota for the related projects or increase the capacity of the region.
- Check whether you have more out-of-bound-created resources within the region. If so, make sure, using ACL, that these resources are not visible to the user ID you have configured for the region access in `credentails.json`.

Region names displayed incorrectly in the Virtual Image window

A known issue exists where IBM Cloud Orchestrator removes the name after the underscore ("_") in the region name when registering images.

Symptoms

`EC2-US-WEST_NORTHERN-CA` and `EC2-US-WEST_OREGON` regions are displayed as `EC2-US-WEST` when registering an image. This error prevents you from selecting images from both regions.

Resolving the problem

1. Edit `/opt/ibm/ico/wlp/usr/servers/pcg` and replace `"_"` (underscore) in region names with a `"–"` (dash).

   Note: The `/opt/ibm/ico/wlp/usr/servers/pcg` is the default directory path. However, if you have a customized IBM Cloud Orchestrator installation directory, use that path instead of the default path.
2. Restart the Public Cloud Gateway to allow the changes to take effect:

   ```
   service pcg restart
   ```
3. Log in to IBM Cloud Orchestrator and wait for the new regions to display. Once the images are displayed, delete the old `EC2-US-WEST_NORTHERN-CA` and `EC2-US-WEST_OREGON` cloud groups and hypervisors. You can also delete any registered images that belong to these regions.
4. Log in to the OpenStack Controller and run the following command:

   ```
   keystone endpoint-list
   ```

   and identify the old endpoints by their region name.
5. Delete the old endpoints for `EC2-US-WEST_NORTHERN-CA` and `EC2-US-WEST_OREGON`, by running the following command:

   ```
   keystone endpoint-delete <endpoint-id>
   ```

   Note: Be careful not to delete any valid endpoints.
6. Images can now be registered for both regions.

SSH key deployment failures

SSH keys have different scopes and assumptions depending on the remote cloud.

Symptoms

IBM SoftLayer is the most restrictive cloud. You can only register a given SSH key once within an account. This is because the fingerprint of the SSH key is used as a unique key.

When multiple IBM SoftLayer regions are defined in the Public Cloud Gateway and more than one of these regions are backed by a single IBM SoftLayer account, an SSH key can be deployed in only one of these regions.

For information about SSH key management, see "SSH key management" on page 254.

Resolving the problem

The Public Cloud Gateway has caches for SSH keys per region. For this scenario, you must set the `keypairTimeout` and the `keypairQuotaTimeout` in `config.json` to 0. This disables the caches for the SSH key and so the SSH key in the IBM SoftLayer account is immediately visible in each of the regions that map to a single IBM SoftLayer account.

Note: Failure to perform this configuration results in SSH key deployment errors during the first virtual machine deployment or a newly registered SSH key.

Timeouts during resource modification processing

For regions managed by the Public Cloud Gateway, there is a possibility that timeouts might happen during the following actions: Create, Delete, Attach, or Detach of a resource.

Causes

Root cause is that there are multiple levels of timeout handling in such a management scenario.

Public Cloud Gateway is frequently updating its internal caches with the live data from the remote clouds. These refresh times are configurable through configuring caching (for details, refer to: "Configuring caching" on page 267).

The management workflows that orchestrate the modification actions have also a timeout handling. As default they query the status of the modification action once per minute and do it for a certain number of retries. The retry value is configurable in the related toolkits.

Depending on how the cache refresh in the Public Cloud Gateway is configured related to the workflow retries, there might be situations where the workflow does not wait long enough for a status change depending on the time the modification action requires.

In Public Cloud Gateway scenarios the time a remote request lasts might have a huge variation time depending on:

- The size of the resource that must be modified.
- The remote cloud that is the target of the modification (Amazon AWS EC2 or SoftLayer).
- The time of the day the request is performed.
- The day within a week or month the request is performed.

The retries within the workflows are defaulted based on variation tests. There might be situations where these values are too small.

If such a situation happens that the retry count is too short, you would see messages in the workflow log that the workflow waited for all the retries without success.

Resolving the problem

Check in the remote cloud if the intended modification action was completed successful. If this is true ignore the workflow error. If the modification action was not successful rerun the request / offering. If the situation happens frequently, get in touch with IBM Customer Support to get instructions to increase the retry count in the workflows.

Unable to connect to a public cloud due to missing credentials

In the Public Cloud Gateway, you might receive the error `Unable to connect to public cloud due to missing credentials`.

Causes

This problem is due to tenants or projects being present in IBM Cloud Orchestrator that are not accounted for in the `credentials.json` file.

Resolving the problem

You can resolve this problem in one of the following ways:

- Add credentials for each tenant in IBM Cloud Orchestrator to the `credentials.json` file.
- Add credentials for a specific tenant by ID in IBM Cloud Orchestrator to the `credentials.json` file.
- Add credentials, where the `tenantName` is * as stated in step 5 in the related configuring topic. This ensures that these credentials are applied to each tenant that is not explicitly stated in `credentials.json` file.

Related tasks

Configuring the Public Cloud Gateway

The Public Cloud Gateway is deployed as part of the IBM Cloud Orchestrator installation. However, the Public Cloud Gateway is not enabled by default and certain updates to the configuration files are required before you can use the Public Cloud Gateway.

Troubleshooting a VMware region

Use this topic to solve issues that might occur in a VMware region.

Region creation failure

During region creation, the following error might occur in `SystemOut.log`:

```
"VMwareConnect E com.ibm.orchestrator.plugin.vmware.VMwareConnection connect
ERROR : Connection failed javax.xml.ws.soap.SOAPFaultException:
javax.net.ssl.SSLHandshakeException: General SSLEngine problem
[3/20/18 5:36:51:680 EDT]
0000013b VMwareService E
com.ibm.orchestrator.plugin.vmware.VMwareServiceUtilities getVMResources
ERROR : java.lang.NullPointerException "
```

As a resolution, delete the respective vCenter's node trust and cell trust certificates from WebSphere Application Server console and create them again.

Incorrect VMware Region affects Business Process Manager toolkit

If you register an incorrect VMware region, then you cannot open any Business Process Manager toolkit. As a resolution, restart Business Process Manager service every time you register an incorrect VMware region.

Cannot find attached volume after `volume-attach`:

The nova **`volume-attach`** can be used to attach a volume to an instance. Sometimes, the **`volume-attach`** command runs successfully, but when you run the **`fdisk -l`**, you cannot find the attached volume. After you restart the virtual machine, the volume is found. It is a known issue for a VMware hosted system. There are some workarounds that you can use to discover the attached volume without rebooting the guest operating system, such as logging in to the guest operating system and running the following command:

```
echo "- - -" > /sys/class/scsi_host/host#/scan
```

A VMware instance is shut down every time it is restarted:

OpenStack has a feature where power states are synced between the OpenStack database and the managed hypervisors. If OpenStack records a virtual machine as shutdown, it ensures that the virtual machine is shut down on the hypervisor as well. So, if a cluster inside a vCenter is managed by more than one OpenStack VMware region, when a virtual machine is started from a VMware region, the other region tries to stop it. To avoid this, make sure that each cluster of vCenter is managed by only one OpenStack VMware Region. When you configure a VMware Region, consider the following issues:

- You cannot share a vCenter cluster across IBM Cloud Orchestrator installations. When you do, and then stop a virtual machine, whenever you start it, the other OpenStack ensures that it is stopped within 60 seconds.

- You cannot host the managed-from environments on the managed-to vCenter.
- Avoid out of band operations.

VMwareDriverException: The object has already been deleted or has not been completely created {u'obj': <MOREF id>}

Together with name and UIID , VMware uses a Managed Object Reference Identifier (MOREF id) to identify a specific instance of a virtual machine in the vCenter. Be aware that the identifier is renewed every time the image is added to the inventory, like when the virtual machine is removed from vCenter and then re-added, also if its identity does not change. This might happen for example if the virtual machine is downloaded from a datastore and then reloaded, or when the entire virtual machine is backed up using Tivoli Storage Manager for Virtual Environments. The IBM Cloud Orchestrator OpenStack VMware driver uses the MOREF id to index virtual machine instances in a cache. If it changes, like in the cases mentioned before, the entry in the cache becomes invalid. In that case an error message is shown in the `compute.log` when you try to run an action on the instance:

```
VMwareDriverException: The object has already been deleted or
                       has not been completely created {u'obj': <MOREF id>}
```

to avoid such error the administrator must apply the following practices when an activity must be run in vCenter that may result in a change of the MOREF id:

1. Stop the `openstack-nova-compute` service with:

   ```
   systemctl stop <openstack-nova-compute service name>
   ```

2. Run the activity that causes the MOREF change.
3. Start the `openstack-nova-compute service`:

   ```
   systemctl start <openstack-nova-compute service name>
   ```

If the error already occurred and the tasks run on the virtual machine fail, do not try to delete the virtual machine from IBM Cloud Orchestrator interfaces, but instead restart the Nova compute to refresh the cache.

Troubleshooting a PowerVC region

Use this topic to solve issues that might occur in a PowerVC region.

The following log files for PowerVC are available in the `/var/log/powervc/` directory:

- `glance-powervc.log`
- `neutron-powervc.log`
- `nova-powervc.log`
- `cinder-powervc.log`

The logging level of each log file is controlled by the configuration file of the base OpenStack components. For more information, see "Setting logging levels" on page 361.

PowerVC images/volumes cannot synchronize with IBM Cloud Manager with OpenStack

Whenever the following scenario occurs, there might be issues around the driver functionality because the images/volumes from PowerVC cannot synchronize with IBM Cloud Manager with OpenStack:

- Deploy Power cloud with IBM Cloud Manager with OpenStack that configures keystone v2 endpoint.
- In IBM Cloud Orchestrator, configure v3 endpoint for keystone on the controller.

The older configurations of IBM Cloud Orchestrator removed v2 endpoints, but it is not the case now. Though the support got included for VMware use cases, the existence of v2 and v3 endpoints break the PowerVC driver functionality in PowerVC cloud deployments.

To resolve this problem, run the following commands to manually remove the legacy v2 endpoints for keystone:

```
source /root/v3rc
openstack endpoint list
openstack endpoint delete <id>
```

Missing PowerVC services

When `ICM_config_ico.sh` script tries to invoke some of the missing PowerVC services, the following warning message is displayed:

```
"unary operator expected"
```

You can ignore this warning in PowerVC as it does not affect the installation process and the `ICM_config_ico.sh` script completes successfully.

Unable to create or delete volumes in PowerVC

When the `openstack-cinder-powervc` service fails, due to lack of SSL certificate or another issue, the `openstack-cinder-volume` service might also fail. To solve the problem, restart the `openstack-cinder-volume` service on the OpenStack Controller.

Authentication errors communicating with either keystone of PowerVC

- Confirm OpenStack credentials are correct.
- Confirm PowerVC credentials are correct.
- Confirm PowerVC SSL certificate is at `/etc/pki/tls/certs/powervc.crt` on the OpenStack Controller.
- Confirm that the PowerVC can communicate with the PowerVC server by the means specified by the ssl certificate. This can be by IP address, host name, or FQDN, depending on the installation of PowerVC. This is resolved by either ensuring that both OpenStack Controller and PowerVC server are set up on the same DNS server correctly. Alternatively, put an appropriate entry in the `/etc/hosts` file of the OpenStack Controller.

Authentication errors regarding staging user/tenant

Confirm that the staging user is a member of the staging tenant.

Live migration does not occur

- When performing a live migration, the log file states that the resource monitoring and control is down on the virtual machine. The resource monitoring and control is a program that hardware management console uses to communicate with the LPARS. This is standard on AIX deployments, but is not on Linux on Power.
- Confirm that the resource monitoring and control is running on the virtual machine to be migrated. This can be confirmed by an **OK** health status by the nova show command or by viewing the PowerVC user interface. If the resource monitoring and control is not running on an AIX deployment, reset it. The image must also be recaptured as the resource monitoring and control often gets disabled by the Network Installation Manager deployment process.

  ```
  # /usr/sbin/rsct/bin/rmcdomainstatus -s ctrmc  -check status
  ```

  - Stopping and starting resource monitoring and control without erasing configuration:
    - `# /usr/sbin/rsct/bin/rmcctrl -z` - Stops the daemons
    - `# /usr/sbin/rsct/bin/rmcctrl -A` - Adds entry to `/etc/inittab` and it starts the daemons
    - `# /usr/sbin/rsct/bin/rmcctrl -p` - Enables the daemons for remote client connections

  If it is an image for Linux on Power, there are instructions on how to install the resource monitoring and control in the PowerVC documentation.

Volume resize operations do not occur or fail

Shared Storage Pool backed storage does not support resize of disk:

```
2014-11-11 04:49:34.916 4750 ERROR oslo.messaging.rpc.dispatcher [-]
Exception during message handling: Get error: RPCException: Cinder API error:
NV-37CDE0F The host 228b02eb0e47e611e4b5060000c9f82a76 for boot volume
991b9732-041d-4dab-8ef8-75006216920d  does not support extend volume.
```

Shared Volume Controller FlashCopy operations can only occur serially per image. If it is doing a copy or extending operation on a boot volume, you cannot start a new operation to make changes to the volume. You can check the Shared Volume Controller user interface to see whether a flashcopy is in progress to the target volume or the current target volume is being extended:

```
2014-11-11 01:24:58.006 2095 TRACE nova.openstack.common.loopingcall
ResizeError: Get error: PVCExpendvdiskFCMapException: Flashcopy is in
progress for boot volume, volume size didn't change. Please try again later.
(HTTP 409) (Request-ID: req-29350427-32b3-4721-baca-504c5216b041)
during resizing the instance in the PowerVC
```

Changing the PowerVC user name and password on the OpenStack Controller if the PowerVC user name or password changes

1. On the OpenStack Controller, edit the following file:

   ```
   /etc/powervc/powervc.conf
   ```

   Note: Take a backup of the file before making any changes.
2. Navigate to [powervc] and locate admin_user and admin_password.
3. Change the user name and password by using admin_user and admin_password.

   Note: The password must be encrypted so use openstack-obfuscate <password> to generate.
4. Click **Save**.
5. Restart the following services:

   ```
   service openstack-glance-powervc restart
   service openstack-neutron-powervc restart
   service openstack-nova-powervc restart
   service openstack-cinder-powervc restart
   service openstack-cinder-volume restart
   ```

Unable to attach volumes to Instances on Shared Storage Pool backed environment

Volume attach on Shared Storage Pool backed environments requires that the Resource Monitoring and Control (RMC) software is installed and running on deployed instances. If the particular instance that you are deploying never appears to have RMC running, in other words the health of the virtual machine is warning, it is necessary to reset it (relevant to all operating system types) or install it, if it was not already installed (relevant to Linux on Power). AIX must have RMC included by default.

To install the RMC for Linux on Power, follow the instructions at http://www.ibm.com/support/customercare/sas/f/lopdiags/home.html to install the service and productivity tools.

To reset RMC, perform the following commands on a running instance and then recapture by using the usual process:

```
  # /usr/sbin/rsct/bin/rmcctrl -z
  # /usr/sbin/rsct/bin/rmcctrl -A
  # /usr/sbin/rsct/bin/rmcctrl -p
```

After the previous commands are run, the status of the virtual machine in the PowerVC UI must change from warning to OK. Changing the status might take several minutes.

`rstrip` error in `nova-powervc.log` and some actions such as starting or stopping the virtual machine is not responsive

Stop and start the services in the OpenStack Controller by using `SCOrchestrator.py`, if the following error is displayed in the `nova-powervc.log`:

```
2014-12-17 02:57:10.348 26453 ERROR powervc.nova.driver.compute.manager [-]
Exception: Exception: 'NoneType' object has no attribute 'rstrip'
```

OpenStack Controller does not support secure connection to PowerVC server with mixed-case or uppercase host name

This problem affects only secure connections; insecure connections are unaffected. The following error is displayed:

```
Host "hostname" does not match x509 certificate contents: CommonName "hostNAME",
subjectAltName "DNS:hostNAME, DNS IP"
```

To fix this problem, complete the following steps:

1. Reconfigure the PowerVC server so that the host name is specified in lowercase letters. Include the fully qualified domain name.
2. Reconfigure the PowerVC application by using the **`powervc-config`** command.
3. Run the **`powervc-replace-cert`** command to generate a new `powervc.crt` file that is based on the new host name.
4. Restart the `httpd` service on the PowerVC server.
5. Replace the `powervc.crt` file on the OpenStack Controller with the new `powervc.crt` file generated in step "3" on page 393.
6. Restart the PowerVC services on the OpenStack Controller.

Deploy single virtual server offering always produces a `no valid host found` error

PowerVC has a mechanism known as Storage Connectivity Groups, it is a way of grouping storage in PowerVC. Some images only support certain Storage Connectivity Groups depending on the storage types they were created from. The **Deploy single virtual server** offering has no knowledge of the Storage Connectivity Group and thus it always attempts to deploy an AIX or Linux on Power server to the first Storage Connectivity Group that is listed in the `/etc/powervc.conf` file on the PowerVC OpenStack Controller. If this Storage Connectivity Group is not supported by the image being deployed, the deployment fails with the `no valid host found` error message. See "Heat template examples" on page 234 for an example of how to deploy an AIX or Linux on Power server using Heat that is able to leverage Storage Connectivity Groups.

Accessibility features for IBM Cloud Orchestrator

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features of IBM Cloud Orchestrator are described in this topic.

Accessibility features

The following list includes the major accessibility features in IBM Cloud Orchestrator:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices

Note: The default configuration of JAWS screen reader does not read tooltips. JAWS users must enable their current mode to read tooltips by selecting **Utilities > Settings Center > Speech Verbosity > Verbosity Level > Configure Verbosity Levels**.

User documentation is provided in HTML and PDF format. Descriptive text is provided for all documentation images.

The knowledge center, and its related publications, are accessibility-enabled.

Related accessibility information

You can view the publications for IBM Cloud Orchestrator in Adobe Portable Document Format (PDF) using the Adobe Reader. PDF versions of the documentation are available in the knowledge center.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of IBM Cloud Orchestrator. This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM Cloud Orchestrator. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking: *Programming Interface information*.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user’s user name, or other personally identifiable information for purposes of session management, enhanced user usability, single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Glossary

This glossary includes terms and definitions for IBM Cloud Orchestrator.

The following cross-references are used in this glossary:

- See refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- See also refers you to a related or contrasting term.

To view glossaries for other IBM products, go to www.ibm.com/software/globalization/terminology (opens in new window).

A

account code
A code that uniquely identifies an individual, billing, or reporting entity within chargeback and resource accounting.

account code conversion table
An ASCII text file that contains the definitions that are required to convert the identifier values defined by the account code input field to the user-defined output account codes.

account report
A report that is used to show account level information for usage and charge.

availability zone
A logical group of OpenStack Compute hosts. It provides a form of physical isolation and redundancy from other availability zones, such as by using separate power supply or network equipment.

B

Bill program
A program that performs cost extensions within SmartCloud Cost Management and summarizes cost and resource utilization by account code. The Bill program uses the rate code table that is assigned to the client to determine the amount to be charged for each resource consumed.

building block
The model of an image that is created by combining models of a base operating system and software bundles. Each building block contains a semantic and functional model that describes the contents of the components, for example, the installed products, supported operating systems, prerequisites, and requirements.

business object
A software entity that represents a business entity, such as an invoice. A business object includes persistent and nonpersistent attributes, actions that can be performed on the business object, and rules that the business object is governed by.

business process
A defined set of business activities that represent the required steps to achieve a business objective. A business process includes the flow and use of information and resources.

C

chargeback identifier
A label, which is often tied to an algorithm or set of rules, that is not guaranteed to be unique, but is used to identify and distinguish a specific chargeback item or chargeback entity from others.

compute node
A node that runs a virtual machine instance, which provides a wide range of services, such as providing a development environment or performing analytics.

consolidation process
A process during which the data collectors process the nightly accounting and storage files that were created by the data collection scripts and produce an output CSR file.

conversion mapping
An entry in a mapping table which allows you to map identifiers to accounts or other identifiers.

custom node
A virtual image part that provides an unconfigured node for a pattern that has a deployment manager or a control node as its base.

E

exception file
A file that contains a list of records with identifier names that do not have a matching Parameter IdentifierName attribute value.

exception processing
A process in which the system writes all records that to do match an entry in the account code conversion table to an exception file.

H

human service
An activity in the business process definition that creates an interactive task that the process participants can perform in a web-based user interface.

hypervisor
Software or a physical device that enables multiple instances of operating systems to run simultaneously on the same hardware.

K

kernel
The part of an operating system that contains programs for such tasks as input/output, management and control of hardware, and the scheduling of user tasks.

P

parameter (parm)
A value or reference passed to a function, command, or program that serves as input or controls actions. The value is supplied by a user or by another program or process.

parm
See parameter.

performance counter
A utility that provides a way for software to monitor and measure processor performance.

primary key
In a relational database, a key that uniquely identifies one row of a database table.

process application
A container in the Process Center repository for process models and supporting implementations. A process application typically includes business process definitions (BPDs), the services to handle

implementation of activities and integration with other systems, and any other items that are required to run the processes. Each process application can include one or more tracks.

proration

A process that distributes the overall or individual resources of an account and the cost of those resources across multiple accounts at a specified percentage.

proration table

An ASCII text file that defines the identifier values and rate codes that are used in the proration process.

R

rate code

The identifier of a rate that is used to link a resource unit or volume metric with its charging characteristics.

rate group

A group of rate codes that is used to create rate subtotals in reports, graphs, and spreadsheets.

registry

A repository that contains access and configuration information for users, systems, and software.

S

service operation

A custom operation that can be run in the context of the data center. These operations are typically administrative operations and are used to automate the configuration. Service operations can also be used to enhance the catalog of available services with extra functionality.

software bundle

A collection of software installation files, configuration files, and metadata that can be deployed on a virtual machine instance.

T

toolkit

A container where artifacts can be stored for reuse by process applications or other toolkits.

V

virtual machine (VM)

An instance of a data-processing system that appears to be at the exclusive disposal of a single user, but whose functions are accomplished by sharing the resources of a physical data-processing system.

VM

See virtual machine.

Product Number: 5725-H28