Red Hat Enterprise Linux OpenStack Platform 5
Administration User Guide

Administrating resources in a Red Hat Enterprise Linux OpenStack Platform environment

31 October 2014 Red Hat Documentation Team

Red Hat Enterprise Linux OpenStack Platform 5 Administration User Guide

Administrating resources in a Red Hat Enterprise Linux OpenStack Platform environment

31 October 2014

Red Hat Documentation Team

Legal Notice

Copyright © 2014 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

OpenStack is an open source cloud computing platform for public and private clouds. A series of interrelated projects deliver a cloud infrastructure solution. This guide shows OpenStack admin users how to create and manage resources in an OpenStack cloud with the OpenStack dashboard or OpenStack client commands.

Table of Contents

Chapter 1. How can I administer an OpenStack cloud? 3

Chapter 2. Dashboard 4

1. Log in to the dashboard 4
2. Manage projects and users 9
3. Manage instances 14
4. Manage volumes and volume types 16
5. Create and manage images 17
6. Manage flavors 19
7. View cloud resources 22
8. Create and manage host aggregates 23
9. Launch and manage stacks 26

Chapter 3. OpenStack command-line clients 27

1. Overview 27
2. Discover the version number for a client 28
3. Get help for client commands 28
4. Manage projects, users, and roles 29
5. Manage services 40
6. Manage images 44
7. Manage volumes 53
8. Manage flavors 54
9. Manage the OpenStack environment 56
10. Manage quotas 59
11. Analyze log files 72

Revision History 76

Chapter 1. How can I administer an OpenStack cloud?

As an OpenStack cloud administrative user, you can manage tenants, known as *projects* , users, services, images, flavors, and quotas.

The examples in this guide show you how to perform tasks by using the following methods:

- OpenStack dashboard. Use this web-based graphical interface, code named "horizon", to view, create, and manage resources and services.
- OpenStack command-line clients. Each core OpenStack project has a command-line client that you can use to run simple commands to view, create, and manage resources and services in a cloud and automate tasks by using scripts.

You can modify these examples for your specific use cases.

In addition to these ways of interacting with a cloud, you can access the OpenStack APIs directly or indirectly through cURL commands or open SDKs. You can automate access or build tools to manage resources and services by using the native OpenStack APIs or the EC2 compatibility API.

To use the OpenStack APIs, it helps to be familiar with HTTP/1.1, RESTful web services, the OpenStack services, and JSON or XML data serialization formats.

Chapter 2. Dashboard

As a cloud administrative user, the OpenStack dashboard lets you create and manage projects, users, images, and flavors. You can also set quotas and create and manage services. For information about using the dashboard to perform end user tasks, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

1. Log in to the dashboard

The dashboard is available on the node with the **`nova-dashboard`** server role.

1. Ask the cloud operator for the host name or public IP address from which you can access the dashboard, and for your user name and password.
2. Open a web browser that has JavaScript and cookies enabled.

   > **Note**
   >
   > To use the Virtual Network Computing (VNC) client for the dashboard, your browser must support HTML5 Canvas and HTML5 WebSockets. The VNC client is based on noVNC. For details, see noVNC: HTML5 VNC Client. For a list of supported browsers, see Browser support.

3. In the address bar, enter the host name or IP address for the dashboard.

   ```
   https://ipAddressOrHostName/
   ```

   > **Note**
   >
   > If a certificate warning appears when you try to access the URL for the first time, a self-signed certificate is in use, which is not considered trustworthy by default. Verify the certificate or add an exception in the browser to bypass the warning.

4. On the **`Log In`** page, enter your user name and password, and click **`Sign In`**.

   The top of the window displays your user name. You can also access **`Settings`** or sign out of the dashboard.

   The visible tabs and functions in the dashboard depend on the access permissions, or roles, of the user you are logged in as.

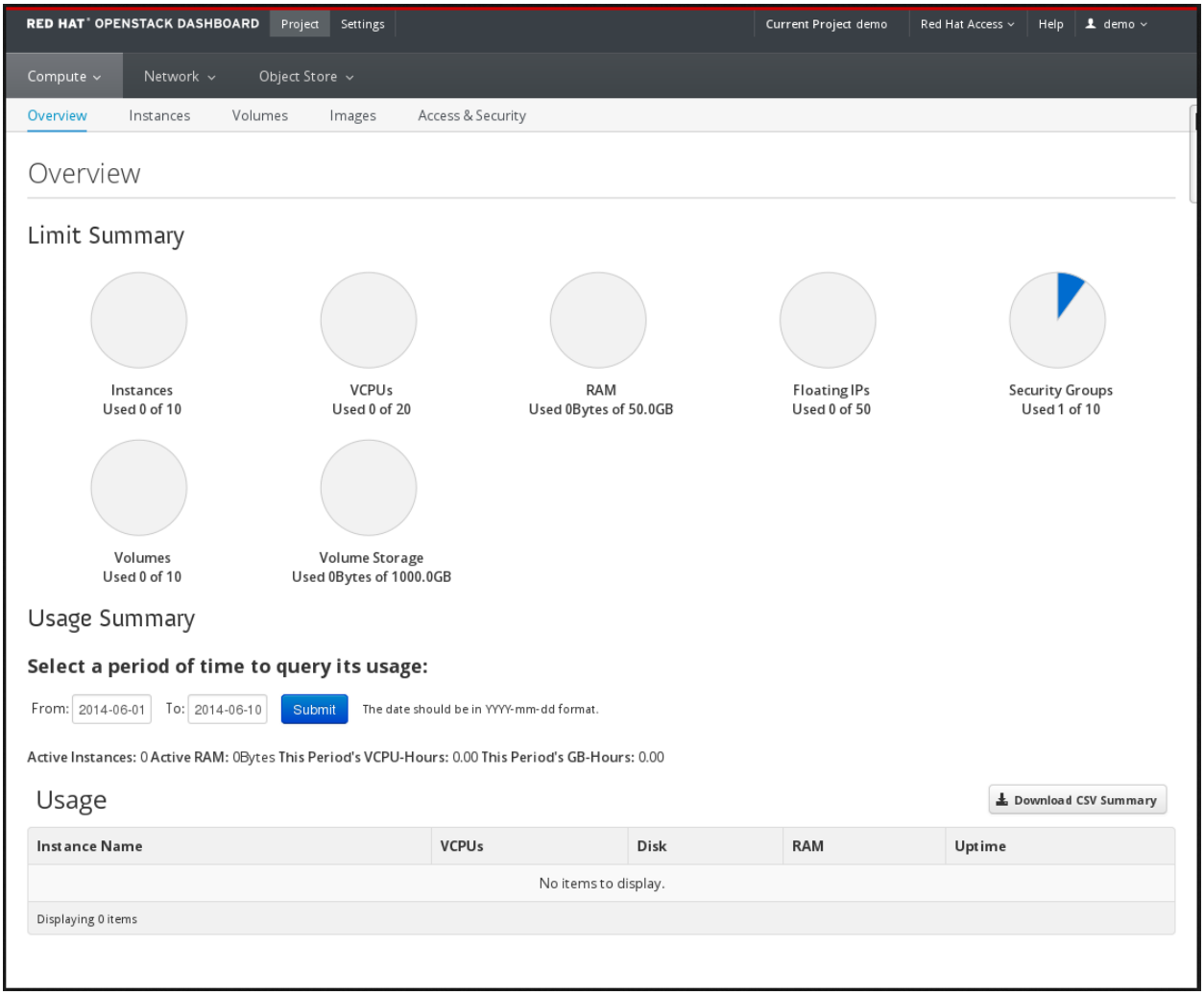
   - If you are logged in as an end user, the Project tab is displayed.
   - If you are logged in as an administrator, the Project tab and Admin tab are displayed.

1.1. OpenStack dashboard- Project tab

Projects are organizational units in the cloud, and are also known as tenants or accounts. Each user is a member of one or more projects. Within a project, a user creates and manages instances.

From the **Project** tab, you can view and manage the resources in a selected project, including instances and images. You select the project from the **Current Project** ***Project_Name*** list at the top of the tab.

Figure 2.1. Project tab



From the **Project** tab, you can access the following tabs:

Compute tab	
Overview	View reports for the project.
Instances	View, launch, create a snapshot from, stop, pause, or reboot instances, or connect to them through VNC.

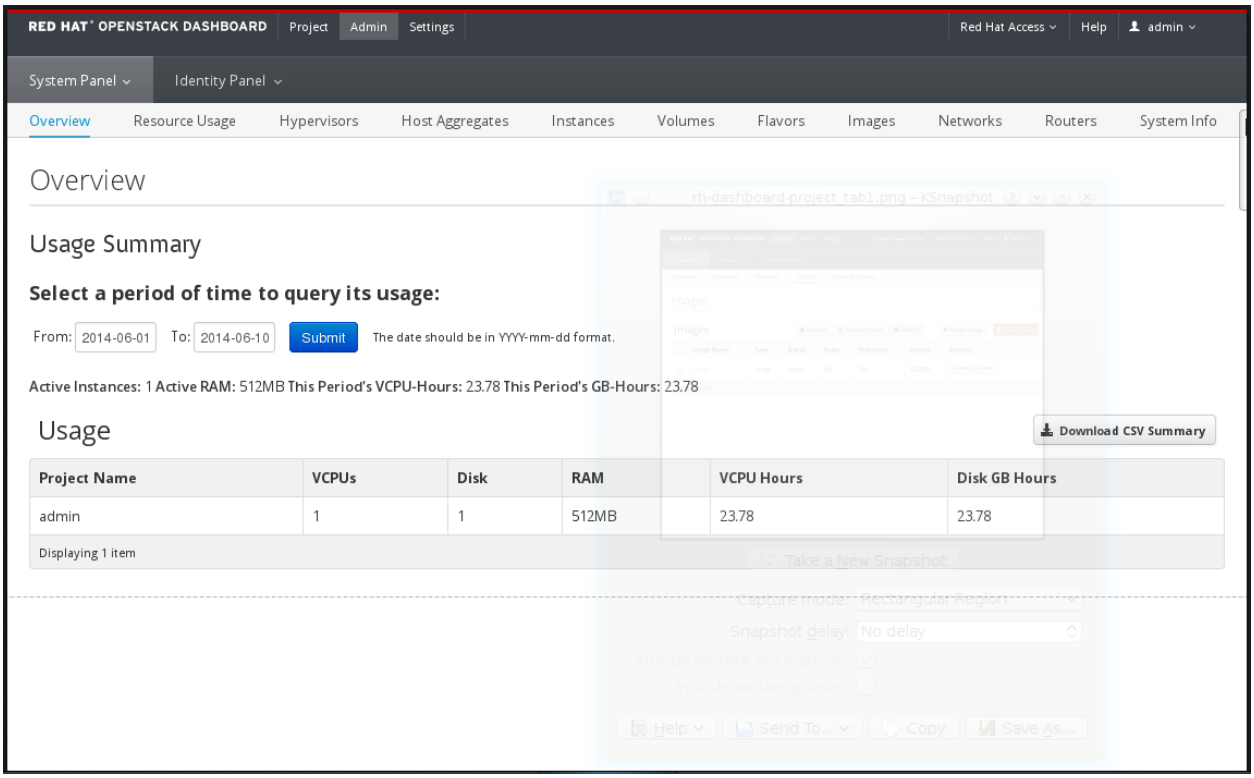
Compute tab	
Volumes	Use the following tabs to complete these tasks: **Volumes** View, create, edit, and delete volumes. **Volume Snapshots** View, create, edit, and delete volume snapshots.
Images	View images and instance snapshots created by project users, plus any images that are publicly available. Create, edit, and delete images, and launch instances from images and snapshots.
Access & Security	Use the following tabs to complete these tasks: **Security Groups** View, create, edit, and delete security groups and security group rules. **Key Pairs** View, create, edit, import, and delete key pairs. **Floating IPs** Allocate an IP address to or release it from a project. **API Access** View API endpoints.
Network tab	
Network Topology	View the network topology.
Networks	Create and manage public and private networks.
Routers	Create and manage subnets.
Object Store tab	
Containers	Create and manage containers and objects.

Compute tab	
Orchestration tab	
Stacks	Use the REST API to orchestrate multiple composite cloud applications.

1.2. OpenStack dashboard- Admin tab

Administrative users can use the **Admin** tab to view usage and to manage instances, volumes, flavors, images, projects, users, services, and quotas.

Figure 2.2. Admin tab



Access the following categories to complete these tasks:

System Panel tab	
Overview	View basic reports.

System Panel tab	
Resource Usage	Use the following tabs to view the following usages: **Daily Report** View the daily report. **Stats** View the statistics of all resources.
Hypervisors	View the hypervisor summary.
Host Aggregates	View, create, and edit host aggregates. View the list of availability zones.
Instances	View, pause, resume, suspend, migrate, soft or hard reboot, and delete running instances that belong to users of some, but not all, projects. Also, view the log for an instance or access an instance through VNC.
Volumes	View, create, edit, and delete volumes and volume types.
Flavors	View, create, edit, view extra specifications for, and delete flavors. A flavor is size of an instance.
Images	View, create, edit properties for, and delete custom images.
Networks	View, create, edit properties for, and delete networks.
Routers	View, create, edit properties for, and delete routers.

System Panel tab	
System Info	Use the following tabs to view the service information: **Services** View a list of the services. **Compute Services** View a list of all Compute services. **Network Agents** View the network agents. **Default Quotas** View default quota values. Quotas are hard-coded in OpenStack Compute and define the maximum allowable size and number of resources.
Identity Panel tab	
Projects	View, create, assign users to, remove users from, and delete projects.
Users	View, create, enable, disable, and delete users.

2. Manage projects and users

As a cloud administrator, you manage both projects and users. Projects are organizational units in the cloud to which you can assign users. Projects are also known as tenants or accounts.You can manage projects and users independently from each other.

Users are members of one or more projects.

During cloud set up, the operator defines at least one project, user, and role. The operator links the role to the user and the user to the project. Roles define the actions that users can perform. As an administrator, you can create additional projects and users as needed.

Learn how to add, update, and delete projects and users, assign users to one or more projects, and change or remove the assignment. To enable or temporarily disable a project or user, update that project or user.

When you create a user account, you must assign the account to a primary project. Optionally, you can assign the account to additional projects. Before you can delete a user account, you must remove the user account from its primary project.

2.1. Consequences of disabling projects and users

When you disable a project, it has the following consequences:

- In the dashboard, users can no longer access the project from the **Current Project** ***Project_Name*** drop-down list on the **Project** tab.
- Users who are members of only the disabled project can no longer log in.
- You cannot launch instances for a disabled project. Instances that are already running are not automatically terminated though—you must stop them manually.
- The data for a disabled project is maintained so that you can enable the project again at any time.

When you disable a user account, the user can no longer log in, but the data for the user is maintained so that you can enable the user again at any time.

2.2. Create a project

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Identity Panel** tab and select the **Projects** category.
3. Click **Create Project**.
4. On the **Project Info** tab in the **Create Project** window, enter a name and description for the project. By default, the project is enabled. See Section 2.1, "Consequences of disabling projects and users".
5. On the **Project Members** tab, add members to the project.
6. On the **Quota** tab, edit quota values.
7. Click **Create Project**.

   The **Projects** category shows the project, including its assigned ID.

2.3. Update a project

You can update a project to change its name or description, and enable or temporarily disable it.

1. On the **Admin** tab, click the **Identity Panel** tab and select the **Projects** category.
2. Select the project that you want to update.
3. In the **More** drop-down list, click **Edit Project**.
4. In the **Edit Project** window, you can update a project to change its name or description, and enable or temporarily disable it.

   By default, the project is enabled. To temporarily disable it, clear the **Enabled** check box. To enable a disabled project, select the **Enabled** check box.

5. Click **Save**.

2.4. Modify user assignments for a project

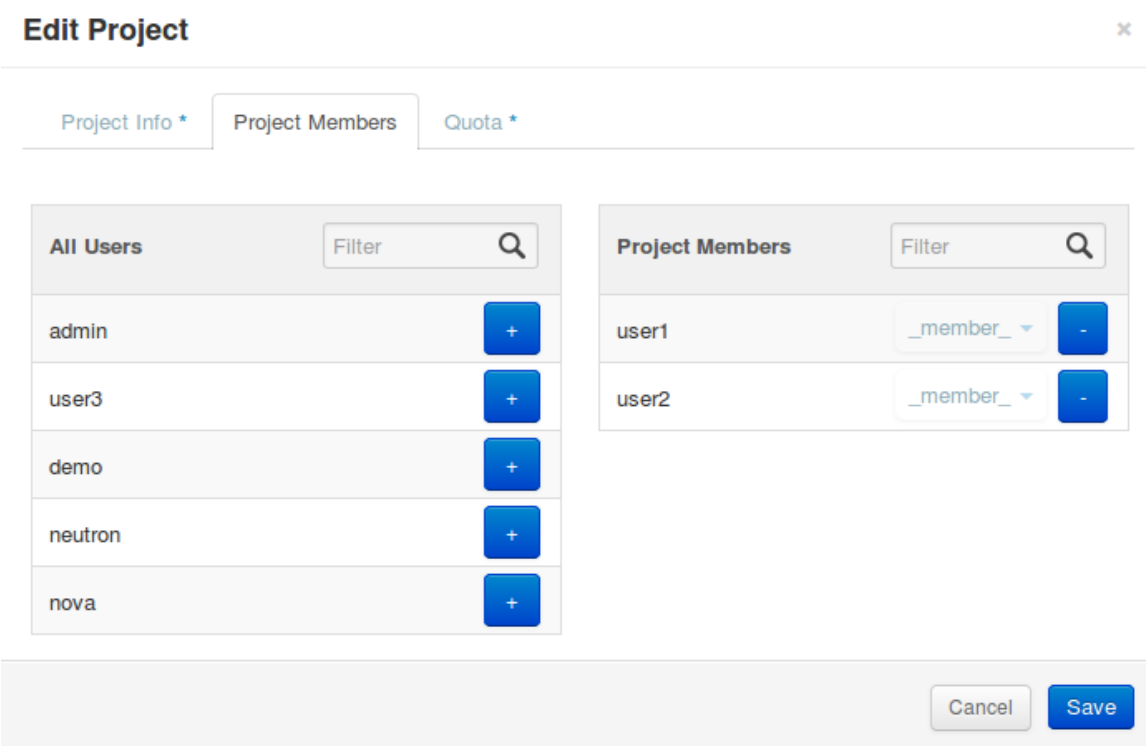
When you create users, you must assign them to a primary project as described in Section 2.6, "Create a user account". You can assign users to additional projects or update and remove assignments.

1. On the **Admin** tab, click the **Identity Panel** tab and select the **Projects** category.
2. Select a project to modify its user assignments.
3. Click the selected project's **Modify Users** button.

   The **Edit Project** window shows the following lists of users:

   - **All Users**. Users that are available to be assigned to the current project.
   - **Project Members**. Users that are assigned to the current project,

   Figure 2.3. Edit the users list



4. To assign a user to the current project, click **+** for the user.

   The user moves from the **All Users** list to the **Project Members** list.
5. To remove a user from the current project, click the **-** button for the user.

   The user moves from the **Project Members** list to the **All Users** list.
6. Click **Save**.

2.5. Delete projects

1. On the **Admin** tab, click the **Identity Panel** tab and select the **Projects** category.
2. Select the projects that you want to delete.
3. Click **Delete Projects** to confirm the deletion.

> **Note**
>
> You cannot undo the delete action.

2.6. Create a user account

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
2. On the **Admin** tab, click the **Identity Panel** tab and select the **Users** category.
3. Click **Create User**.
4. In the **Create User** window, enter a user name, email, and preliminary password for the user. Confirm the password.

   Select a project from the **Primary Project** drop-down list.

   Choose a role for the user from the **Role** drop-down list. Default is **_member_**.
5. Click **Create User** to confirm your changes.

   Respond to the prompt to remember the password for the user.

   The dashboard assigns an ID to the user, and the user appears in the **Users** category.

2.7. Disable or enable a user

1. On the **Admin** tab, click the **Identity Panel** tab and select the **Users** category.
2. Select the user that you want to disable or enable. You can disable or enable only one user at a time.
3. In the **Actions** drop-down list, select **Disable User** or **Enable User**.

   In the **Enabled** column, the enabled value updates to either **True** or **False**.

2.8. Delete users

1. On the **Admin** tab, click the **Identity Panel** tab and select the **Users** category.
2. Select the users that you want to delete.
3. Click **Delete Users**.

4. In the **Confirm Delete Users** window, click **Delete Users** to confirm the deletion.

2.9. Manage project security

Security groups are sets of IP filter rules that are applied to all project instances, and which define networking access to the instance. Group rules are project specific; project members can edit the default rules for their group and add new rule sets.

All projects have a default security group that is applied to any instance that has no other defined security group. Unless you change the default, this security group denies all incoming traffic and allows only outgoing traffic to your instance.

> **Note**
>
> For information about updating global controls on the command line, see Section 4.2, "Manage project security".

2.9.1. Create a security group

1. Log in to the dashboard as a project member.
2. On the **Project** tab, select the appropriate project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
3. On the **Compute** tab, click the **Access & Security** category.
4. On the **Security Groups** tab, click **Create Security Group**.
5. Provide a name and appropriate description for the group, and click **Create Security Group**. By default, the new rule provides outgoing access rules for the group.

2.9.2. Add a security group rule

1. Log in to the dashboard as a project member.
2. On the **Project** tab, select the appropriate project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
3. On the **Compute** tab, click the **Access & Security** category.
4. On the **Security Groups** tab, click **Manage rules** for the appropriate security group.
5. To add a rule, click **Add Rule**. Set the attributes for the rule, and click **Add**:

   IP Protocol

   The IP protocol to which the rule applies:

- **TCP**.Typically used to exchange data between systems, and for end-user communication.
- **UDP**. Typically used to exchange data between systems, particularly at the application level.
- **ICMP**. Typically used by network devices, such as routers, to send error or monitoring messages.

Open

For TCP or UDP rules, the **Port** or **Port Range** to open for the rule. Choose to open a single port or range of ports.

For a range of ports, enter port values in the **From Port** and **To Port** fields.

For a single port, enter the port value in the **Port** field.

Source

The source of the traffic for this rule:

- **CIDR** (Classless Inter-Domain Routing). IP address block, which limits access to IPs within the block. Enter the CIDR in the **Source** field.
- **Security Group**. Source group that enables any instance in the group to access any other group instance.

2.9.3. Delete a security group rule

1. Log in to the dashboard as a project member.
2. On the **Project** tab, select the appropriate project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
3. On the **Compute** tab, click the **Access & Security** category.
4. On the **Security Groups** tab, click **Manage rules** for the appropriate security group.
5. To delete a rule, select the rule and click **Delete Rule**.

2.9.4. Delete a security group

1. Log in to the dashboard as a project member.
2. On the **Project** tab, select the appropriate project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
3. On the **Compute** tab, click the **Access & Security** category.
4. On the **Security Groups** tab, select the appropriate group, and click **Delete Security Group**.

3. Manage instances

As an administrative user, you can manage instances for users in various projects. You can view, terminate, edit, perform a soft or hard reboot, create a snapshot from, and migrate instances. You can also view the logs for instances or launch a VNC console for an instance.

For information about using the dashboard to launch instances as an end user, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

3.1. Create instance snapshots

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Instances** category.

3. Select an instance to create a snapshot from it. From the **Actions** drop-down list, select **Create Snapshot**.

4. In the **Create Snapshot** window, enter a name for the snapshot. Click **Create Snapshot**. The dashboard shows the instance snapshot in the **Images & Snapshots** category.

5. To launch an instance from the snapshot, select the snapshot and click **Launch**.

3.2. Control the state of an instance

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

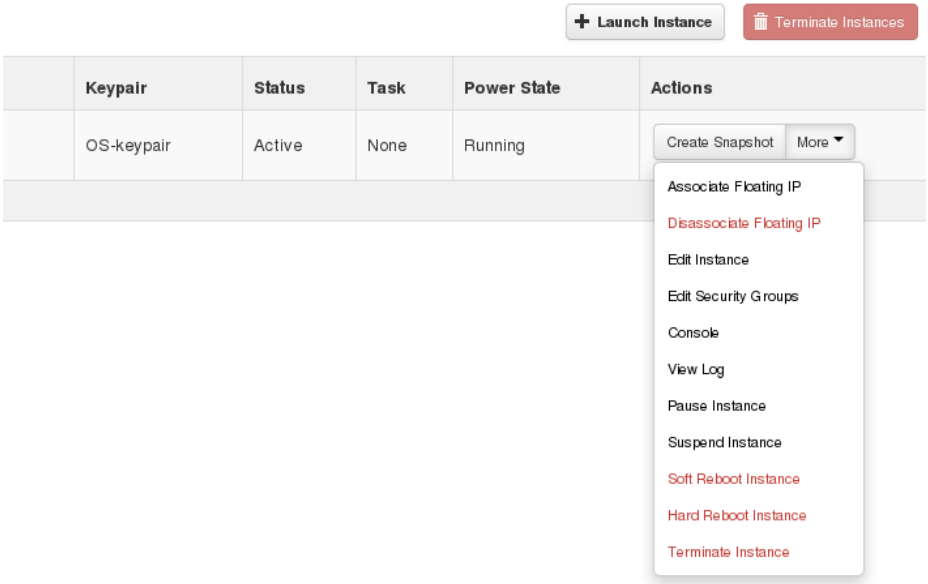
2. On the **Admin** tab, click the **Instances** category.

3. Select the instance for which you want to change the state.

4. In the **More** drop-down list in the **Actions** column, select the state.

   Depending on the current state of the instance, you can choose to pause, un-pause, suspend, resume, soft or hard reboot, or terminate an instance (items in red are disabled).

   Figure 2.4. Dashboard-Instance Actions



3.3. Track usage

Use the **Overview** category to track usage of instances for each project.

You can track costs per month by showing metrics like number of VCPUs, disks, RAM, and uptime of all your instances.

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Instances** category.

3. Select a month and click **Submit** to query the instance usage for that month.

4. Click **Download CSV Summary** to download a CSV summary.

4. Manage volumes and volume types

Volumes are the Block Storage devices that you attach to instances to enable persistent storage. Users can attach a volume to a running instance or detach a volume and attach it to another instance at any time. For information about using the dashboard to create and manage volumes as an end user,see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

As an administrative user, you can manage volumes and volume types for users in various projects. You can create and delete volume types, and you can view and delete volumes.

4.1. Create a volume type

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Volumes** category.
3. Click **Create Volume Type**.

   In the **Create Volume Type** window, enter a name for the volume type.
4. Click **Create Volume Type** to confirm your changes.

4.2. Delete volume types

When you delete a volume type, volumes of that type are not deleted.

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
2. On the **Admin** tab, click the **Volumes** category.
3. Select the volume type or types that you want to delete.
4. Click **Delete Volume Types**.
5. In the **Confirm Delete Volume Types** window, click **Delete Volume Types** to confirm the action.
6. A message indicates whether the action succeeded.

4.3. Delete volumes

When you delete an instance, the data of its attached volumes is not destroyed.

1. Log in to the dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
2. On the **Admin** tab, click the **Volumes** category.
3. Select the volume or volumes that you want to delete.
4. Click **Delete Volumes**.
5. In the **Confirm Delete Volumes** window, click **Delete Volumes** to confirm the action.
6. A message indicates whether the action succeeded.

5. Create and manage images

As an administrative user, you can create and manage images for the projects to which you belong. You can also create and manage images for users in all projects to which you have access.

To create and manage images in specified projects as an end user, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

To create and manage images as an administrator for other users, use the following procedures.

5.1. Create images

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project *Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Images** category. The images that you can administer for cloud users appear on this page.

3. Click **Create Image**.

4. In the **Create An Image** window, enter or select the following values:

Name	Enter a name for the image.
Description	Enter a brief description about the image.
Image Source	Choose the image source from the dropdown list. Your choices are **Image Location** and **Image File**.
Image File or **Image Location**	Based on your selection, there is an **Image File** or **Image Location** field. You can include the location URL or browse for the image file on your file system and add it.
Format	Select the image format.
Minimum Disk (GB) and **Minimum RAM (MB)**	Leave these fields empty.
Public	Select this option to make the image public to all users.

Protected	Select this option to ensure that only users with permissions can delete the image.

5. Click **Create Image**.

   The image is queued to be uploaded. It might take some time before the status changes from queued to active.

5.2. Update images

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Images** category.

3. Select the image that you want to edit. Click **Edit**.

4. In the **Update Image** window, you can change the name for the image. Select the **Public** check box to make the image public. Clear this check box to make the image private. You cannot change the kernel ID, RAM disk ID, or architecture attributes for an image.

5. Click **Update Image**.

5.3. Delete images

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Images** category.

3. Select the images that you want to delete.

4. Click **Delete Images**.

5. In the **Confirm Delete Image** window, click **Delete Images** to confirm the deletion.

   You cannot undo this action.

6. Manage flavors

In OpenStack, a flavor defines the compute, memory, and storage capacity of a virtual server that users can launch. As an administrative user, you can create, edit, and delete flavors.

Only administrative users can create and manage flavors.

A flavor consists of the following parameters:

Table 2.1. Flavor parameters

Parameter	Description
Flavor Name	The flavor name.
VCPUs	Number of virtual CPUs to use.
RAM	Amount of RAM to use, in megabytes.
Root Disk	Amount of disk space (in gigabytes) to use for the root (/) partition.
Ephemeral Disk	Amount of disk space (in gigabytes) to use for the ephemeral partition. If unspecified, the value is 0 by default. Ephemeral disks offer machine local disk storage linked to the life cycle of a VM instance. When a VM is terminated, all data on the ephemeral disk is lost. Ephemeral disks are not included in any snapshots.
Swap Disk	Amount of swap space (in megabytes) to use. If unspecified, the default is 0.
ID	The flavor ID is generated by OpenStack if the option selected is **`auto`**. By default, this field is set to **`auto`**.
Public	Indicates if the flavor is public.

The default flavors are:

- m1.tiny (1 VCPU/1 GB Disk/512 MB RAM)
- m1.small (1 VCPU/20 GB Disk/2048 MB RAM)
- m1.medium (2 VCPU/40 GB Disk/4096 MB RAM)
- m1.large (4 VCPU/80 GB Disk/8192 MB RAM)
- m1.xlarge (8 VCPU/160 GB Disk/16384 MB RAM)

6.1. Create flavors

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Flavors** category.

3. Click **Create Flavor**.

4. In the **Create Flavor** window, enter or select the following values:

Flavor Info tab	
Name	Enter the flavor name.
ID	The flavor ID, generated by OpenStack.
VCPUs	Enter the number of virtual CPUs to use.
RAM MB	Enter the amount of RAM to use, in megabytes.
Root Disk GB	Enter the mount of disk space in gigabytes to use for the root (/) partition.
Ephemeral Disk GB	Enter the amount of disk space in gigabytes to use for the ephemeral partition. If unspecified, the value is 0 by default. Ephemeral disks offer machine local disk storage linked to the life cycle of a VM instance. When a VM is terminated, all data on the ephemeral disk is lost. Ephemeral disks are not included in any snapshots.
Swap Disk MB	Enter the amount of swap space (in megabytes) to use. If unspecified, the default is 0.

5. In the **Flavor Access** tab, you can control access to the flavor by moving projects from the **All Projects** column to the **Selected Projects** column.

   Only projects in the **Selected Projects** column can use the flavor. If there are no projects in the right column, all projects can use the flavor.

6. Click **Create Flavor**.

6.2. Manage flavors

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Flavors** category.

3. **To edit a flavor:**

   a. Select the flavor that you want to edit. Click **Edit Flavor**.

   b. In the **Edit Flavor** window, update the flavor name, VCPUs, RAM, root disk, ephemeral disk, and swap disk values in the **Flavor Info** tab.

   c. In the **Flavor Access** tab, use **+** to control the access by moving projects from the left column to the right column.

   > **Note**
   >
   > You can also perform this action by choosing **Modify Access** from the **More** dropdown list from the **Actions** column.

   d. Click **Save**.

4. **To create new extra spec key-value pair:**

   a. Select the flavor that you want to edit. Click **More**.

   b. Choose **View Extra Specs** from the dropdown list.

   c. Click **Create**.

   d. Select **Keys** from the dropdown list.

   e. Enter values for **Key** and **Value**.

   f. Click **Save**.

6.3. Delete flavors

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Flavors** category.

3. Select the flavors that you want to delete.

4. Click **Delete Flavors**.

5. In the **Confirm Delete Flavors** window, click **Delete Flavors** to confirm the deletion. You cannot undo this action.

7. View cloud resources

7.1. View services information

As an administrative user, you can view information for OpenStack services.

1. Log in to the OpenStack dashboard and choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **System Info** category.

3. Click the:

   - **Services** tab to view services information.

     The page displays the internal name and the public OpenStack name for the service, the host where the service runs, and whether the service is or is not enabled.

   - **Compute Services** tab to view services specific to the Compute Service. Both host and zone are listed for each service, as well as its activation status.

> **Note**
>
> The **System Info** window also offers status information for the following:
>
> - **Availability Zones**
> - **Host Aggregates**
> - **Network Agents**

7.2. View cloud usage statistics

The Telemetry module provides user-level usage data for OpenStack-based clouds, which can be used for customer billing, system monitoring, or alerts. Data can be collected by notifications sent by existing OpenStack components (for example, usage events emitted from Compute) or by polling the infrastructure (for example, libvirt).

> **Note**
>
> You can only view metering statistics on the dashboard (available only to administrators). The Telemetry service must be set up and administered through the **ceilometer** command-line interface (CLI).
>
> For basic administration information, refer to the "Measure Cloud Resources" chapter in the *OpenStack End User Guide*.

7.2.1. View resource statistics

1. Log in to the OpenStack dashboard as a user with Admin privileges.
2. On the **Admin** tab, click the **System Panel** tab and select the **Resource Usage** category.
3. Click the:
   - **Daily Report** tab to view a report of daily usage per project. You can choose the date range and the number of projects.
   - **Stats** tab to view a multi-series line chart with user-defined metrics. You group by project, define the value type (min, max, avg, or sum), and specify the time period (or even use a calendar to define a date range).

8. Create and manage host aggregates

Host aggregates enable administrative users to assign key-value pairs to groups of machines.

Each node can have multiple aggregates and each aggregate can have multiple key-value pairs. You can assign the same key-value pair to multiple aggregates.

The scheduler uses this information to make scheduling decisions. For more information, see the *Configuration Reference Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

Procedure 2.1. To create a host aggregate

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.
2. On the **Admin** tab, click the **Host Aggregates** category.
3. Click **Create Host Aggregate**.
4. In the **Create Host Aggregate** window, enter or select the following values:

Host Aggregate Info tab	
Name	The host aggregate name.

Host Aggregate Info tab	
Availability Zone	The cloud provider defines the default availability zone, such as **us-west**, **apac-south**, or **nova**. **Note** You can target the host aggregate, as follows: - When the host aggregate is exposed as an availability zone, select the availability zone when you launch an instance. - When the host aggregate is not exposed as an availability zone, select a flavor and its extra specs to target the host aggregate.

Host within Aggregate tab	
Selected hosts	To assign a host to the aggregate, click **+** for the host. The host moves from the **All available hosts** list to the **Selected hosts** list. **Note** You can add one host to one or more aggregates. To add a host later, edit the aggregate.

Procedure 2.2. To manage host and aggregates

1. Log in to the dashboard.

   Choose the **admin** project from the **Current Project** ***Project_Name*** drop-down list.

   Here *Project_Name* is the name of the current project.

2. On the **Admin** tab, click the **Host Aggregates** category.

3. To edit host aggregates:

   a. Select the host aggregate that you want to edit. Click **Edit Host Aggregate**

Aggregate.

b. In the **Edit Host Aggregate** window, you can change the name and availability zone for the aggregate.

4. To manage hosts:

   a. Select the host aggregate that you want to edit. Click **More**.

   b. In the **Add/Remove Hosts to Aggregate** window, click **+** to assign a host to the aggregate.

      Click **-** to remove a host that is assigned to an aggregate.

5. To delete host aggregates:

   a. Select the host aggregate that you want to edit. Click **More**.

   b. Click the **Delete Host Aggregate** option.

> **Note**
>
> Availability zones can be created, edited and deleted as a part of managing the host aggregates they are associated with.
>
> In a **packstack --allinone** deployment, **nova** is the default availability zone. In a production deployment with multiple servers, you could have multiple host aggregates associated with different availability zones, based on location and other factors.

9. Launch and manage stacks

The Orchestration service provides a template-based orchestration engine for the OpenStack cloud, which can be used to create and manage cloud infrastructure resources such as storage, networking, instances, and applications as a repeatable running environment.

Templates are used to create stacks, which are collections of resources. For example, a stack might include instances, floating IPs, volumes, security groups, or users. The Orchestration service offers access to all OpenStack core services via a single modular template, with additional orchestration capabilities such as auto-scaling and basic high availability.

For information about:

- Administrative tasks on the command line, see Section 9.4, “Launch and manage stacks”.

  > **Note**
  >
  > There are no administration-specific tasks that can be done through the dashboard.

- The basic creation and deletion of Orchestration stacks, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

Chapter 3. OpenStack command-line clients

1. Overview

You can use the OpenStack command-line clients to run simple commands that make API calls. You can run these commands from the command line or in scripts to automate tasks. If you provide OpenStack credentials, you can run these commands on any computer.

Internally, each client command runs cURL commands that embed API requests. The OpenStack APIs are RESTful APIs that use the HTTP protocol, including methods, URIs, media types, and response codes.

These open-source Python clients run on Linux systems and are easy to learn and use. Each OpenStack service has its own command-line client. On some client commands, you can specify a **`debug`** parameter to show the underlying API request for the command. This is a good way to become familiar with the OpenStack API calls.

The following table lists the command-line client for each OpenStack service with its package name and description.

Table 3.1. OpenStack services and clients

Service	Client	Package	Description
Block Storage	**`cinder`**	python-cinderclient	Create and manage volumes.
Compute	**`nova`**	python-novaclient	Create and manage images, instances, and flavors.
Database Service	**`trove`**	python-troveclient	Create and manage databases.
Identity	**`keystone`**	python-keystoneclient	Create and manage users, tenants, roles, endpoints, and credentials.
Image Service	**`glance`**	python-glanceclient	Create and manage images.
Networking	**`neutron`**	python-neutronclient	Configure networks for guest servers. This client was previously called **`quantum`**.
Object Storage	**`swift`**	python-swiftclient	Gather statistics, list items, update metadata, and upload, download, and delete files stored by the Object Storage service. Gain access to an Object Storage installation for ad hoc processing.
Orchestration	**`heat`**	python-heatclient	Launch stacks from templates, view details of running stacks including events and resources, and update and delete stacks.
Telemetry	**`ceilometer`**	python-ceilometerclient	Create and collect measurements across OpenStack.

An OpenStack **common** client is in development.

For more information on installation instructions and the OpenStack RC file, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

2. Discover the version number for a client

Run the following command to discover the version number for a client:

```
$ PROJECT --version
```

For example, to see the version number for the **nova** client, run the following command:

```
$ nova --version
```

The version number (2.15.0 in the example) is returned.

```
2.15.0
```

3. Get help for client commands

To get usage information, including a list of commands with descriptions, for a client, run the following command:

```
$ CLIENT_NAME help
```

For example, to get help information for the swift client, run the following command:

```
$ swift help
```

> **Note**
>
> Depending on your credentials, you might not have permission to use every command.

After the **help** command, you can enter a command name to get help for that command, as follows:

```
$ CLIENT_NAME help COMMAND_NAME
```

For example, to get help for the glance **image-show** command, enter the following command:

```
$ glance help image-show
```

The command returns a description of the command and its positional and optional arguments:

```
usage: glance image-show [--human-readable] <IMAGE>

Describe a specific image.
```

```
Positional arguments:
  <IMAGE>           Name or ID of image to describe.

Optional arguments:
  --human-readable  Print image size in a human-friendly format.
```

4. Manage projects, users, and roles

As a cloud administrator, you manage projects, users, and roles. Projects are organizational units in the cloud to which you can assign users. Projects are also known as tenants or accounts. Users can be members of one or more projects. Roles define which actions users can perform. You assign roles to user-project pairs.

You can define actions for OpenStack service roles in the **/etc/*PROJECT*/policy.json** files. For example, define actions for Compute service roles in the **/etc/nova/policy.json** file.

You can manage projects, users, and roles independently from each other.

During cloud set up, the operator defines at least one project, user, and role.

Learn how to add, update, and delete projects and users, assign users to one or more projects, and change or remove the assignment. To enable or temporarily disable a project or user, you update that project or user. You can also change quotas at the project level.

Before you can delete a user account, you must remove the user account from its primary project.

Before you can run keystone client commands, you must download and source an OpenStack RC file. See the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

4.1. Services

To look at your service catalog, use these keystone client commands.

4.1.1. service-create

Keyword arguments:

- Name
- Type
- Description

Example:

```
$ keystone service-create \
 --name=nova \
 --type=compute \
 --description="Nova Compute Service"
```

4.1.2. service-list

Arguments

- service_id

Example:

```
$ keystone service-list
```

4.1.3. service-get

Arguments

- service_id

Example:

```
$ keystone service-get 08741d8ed88242ca88d1f61484a0fe3b
```

4.1.4. service-delete

Arguments

- service_id

Example:

```
$ keystone service-delete 08741d8ed88242ca88d1f61484a0fe3b
```

4.1.5. Create a tenant (project)

A tenant is a group of zero or more users. In nova, a tenant owns virtual machines. In swift, a tenant owns containers. In the Dashboard, tenants are represented as projects. Users can be associated with more than one tenant. Each tenant and user pairing can have a role associated with it.

1. To list all projects with their ID, name, and whether they are enabled or disabled:

```
$ keystone tenant-list
```

```
+----------------------------------+--------------------+----
-----+
|                id                |        name        |
enabled |
+----------------------------------+--------------------+----
-----+
| f7ac731cc11f40efbc03a9f9e1d1d21f |       admin        |
True  |
| c150ab41f0d9443f8874e32e725a4cc8 |      alt_demo      |
True  |
| a9debfe41a6d4d09a677da737b907d5e |        demo        |
True  |
```

```
| 9208739195a34c628c58c95d157917d7 | invisible_to_admin |
True  |
| 3943a53dc92a49b2827fae94363851e1 |      service       |
True  |
| 80cab5e1f02045abad92a2864cfd76cb |    test_project    |
True  |
+----------------------------------+--------------------+----
-----+
```

2. Create a project named **new-project**:

```
$ keystone tenant-create --name new-project --description 'my
new project'
```

By default, the project is enabled.

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |          my new project          |
|   enabled   |               True               |
|      id     | 1a4a0618b306462c9830f876b0bd6af2 |
|     name    |           new-project            |
+-------------+----------------------------------+
```

Note the ID for the project so you can update it in the next procedure.

4.1.6. Update a project

Specify the project ID to update a project. You can update the name, description, and enabled status of a project.

1. To temporarily disable a project:

```
$ keystone tenant-update PROJECT_ID --enabled false
```

2. To enable a disabled project:

```
$ keystone tenant-update PROJECT_ID --enabled true
```

3. To update the name of a project:

```
$ keystone tenant-update PROJECT_ID --name project-new
```

4. To verify your changes, show information for the updated project:

```
$ keystone tenant-get PROJECT_ID
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |          my new project          |
```

```
|   enabled   |               True               |
|      id     | 1a4a0618b306462c9830f876b0bd6af2 |
|     name    |           project-new            |
+-------------+----------------------------------+
```

4.1.7. Delete a project

- To delete a project:

```
$ keystone tenant-delete PROJECT_ID
```

4.1.8. Create a user

1. To list all users:

```
$ keystone user-list
```

The output shows the ID, name, enabled status, and e-mail address for each user:

```
+----------------------------------+----------+---------+----
------------------+
|                id                |   name   | enabled |
email         |
+----------------------------------+----------+---------+----
------------------+
| 352b37f5c89144d4ad0534139266d51f |  admin   |   True  |
admin@example.com   |
| 86c0de739bcb4802b8dc786921355813 |   demo   |   True  |
demo@example.com   |
| 32ec34aae8ea432e8af560a1cec0e881 |  glance  |   True  |
glance@example.com  |
| 7047fcb7908e420cb36e13bbd72c972c |   nova   |   True  |
nova@example.com   |
+----------------------------------+----------+---------+----
------------------+
```

2. To create a user, you must specify a name. Optionally, you can specify a tenant ID, password, and email address. It is recommended that you include the tenant ID and password because the user cannot log in to the dashboard without this information.

To create the **new-user** user:

```
$ keystone user-create --name new-user --tenant_id
1a4a0618b306462c9830f876b0bd6af2 --pass PASSWORD
```

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |                                  |
| enabled  |               True               |
```

```
|    id    | 6e5140962b424cb9814fb172889d3be2 |
|   name   |              new-user             |
| tenantId | 1a4a0618b306462c9830f876b0bd6af2 |
+----------+----------------------------------+
```

4.1.9. Update a user

You can update the name, email address, and enabled status for a user.

1. To temporarily disable a user account:

   ```
   $ keystone user-update USER_ID --enabled false
   ```

   If you disable a user account, the user cannot log in to the dashboard. However, data for the user account is maintained, so you can enable the user at any time.

2. To enable a disabled user account:

   ```
   $ keystone user-update USER_ID --enabled true
   ```

3. To change the name and description for a user account:

   ```
   $ keystone user-update USER_ID --name user-new --email new-
   user@example.com
   ```

   ```
   User has been updated.
   ```

4.1.10. Delete a user

- To delete a specified user account:

  ```
  $ keystone user-delete USER_ID
  ```

4.1.11. Create and assign a role

Users can be members of multiple projects. To assign users to multiple projects, define a role and assign that role to a user-project pair.

1. To list the available roles:

   ```
   $ keystone role-list
   ```

   ```
   +----------------------------------+---------------+
   |                id                |      name     |
   +----------------------------------+---------------+
   | 71ccc37d41c8491c975ae72676db687f |     Member    |
   | 149f50a1fe684bfa88dae76a48d26ef7 | ResellerAdmin |
   ```

```
| 9fe2ff9ee4384b1894a90878d3e92bab |    _member_   |
| 6ecf391421604da985db2f141e46a7c8 |     admin     |
| deb4fffd123c4d02a907c2c74559dccf |  anotherrole  |
+----------------------------------+---------------+
```

2. To create the **`new-role`** role:

```
$ keystone role-create --name new-role
```

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | bef1f95537914b1295da6aa038ef4de6 |
|   name   |             new-role             |
+----------+----------------------------------+
```

3. To assign a user to a project, you must assign the role to a user-project pair. To do this, you need the user, role, and project IDs.

   a. To list users:

```
$ keystone user-list
```

```
+----------------------------------+----------+--------
-+----------------------+
|                id                |   name   | enabled
|        email         |
+----------------------------------+----------+--------
-+----------------------+
| 352b37f5c89144d4ad0534139266d51f |  admin   |   True
|  admin@example.com   |
| 981422ec906d4842b2fc2a8658a5b534 | alt_demo |   True
| alt_demo@example.com |
| 036e22a764ae497992f5fb8e9fd79896 |  cinder  |   True
|  cinder@example.com  |
| 86c0de739bcb4802b8dc786921355813 |   demo   |   True
|   demo@example.com   |
| 32ec34aae8ea432e8af560a1cec0e881 |  glance  |   True
|  glance@example.com  |
| 7047fcb7908e420cb36e13bbd72c972c |   nova   |   True
|   nova@example.com   |
+----------------------------------+----------+--------
-+----------------------+
```

   Note the ID of the user to which you want to assign the role.

   b. To list role IDs:

```
$ keystone role-list
```

```
+----------------------------------+---------------+
|                id                |      name     |
+----------------------------------+---------------+
| 71ccc37d41c8491c975ae72676db687f |     Member    |
```

```
| 149f50a1fe684bfa88dae76a48d26ef7 | ResellerAdmin |
| 9fe2ff9ee4384b1894a90878d3e92bab |    _member_   |
| 6ecf391421604da985db2f141e46a7c8 |     admin     |
| deb4fffd123c4d02a907c2c74559dccf |  anotherrole  |
| bef1f95537914b1295da6aa038ef4de6 |    new-role   |
+----------------------------------+---------------+
```

Note the ID of the role that you want to assign.

c. To list projects:

```
$ keystone tenant-list
```

```
+----------------------------------+------------------
-+---------+
|                id                |        name
| enabled |
+----------------------------------+------------------
-+---------+
| f7ac731cc11f40efbc03a9f9e1d1d21f |       admin        |
True  |
| c150ab41f0d9443f8874e32e725a4cc8 |      alt_demo
|   True  |
| a9debfe41a6d4d09a677da737b907d5e |        demo
|   True  |
| 9208739195a34c628c58c95d157917d7 | invisible_to_admin |
True  |
| caa9b4ce7d5c4225aa25d6ff8b35c31f |      new-user
|   True  |
| 1a4a0618b306462c9830f876b0bd6af2 |    project-new
|   True  |
| 3943a53dc92a49b2827fae94363851e1 |      service
|   True  |
| 80cab5e1f02045abad92a2864cfd76cb |    test_project
|   True  |
+----------------------------------+------------------
-+---------+
```

Note the ID of the project to which you want to assign the role.

4. Assign a role to a user-project pair. In this example, you assign the **`new-role`** role to the **`demo`** and **`test-project`** pair:

```
$ keystone user-role-add --user USER_ID --role ROLE_ID --
tenant TENANT_ID
```

5. To verify the role assignment:

```
$ keystone user-role-list --user USER_ID --tenant TENANT_ID
```

```
+------------------------+----------+------------------------
+------------------------+
|           id           |   name   |        user_id
|       tenant_id        |
+------------------------+----------+------------------------
```

```
+-------------------------+
| bef1f95537914b1295.etc | new-role | 86c0de739bcb4802b8.etc |
80cab5e1f02045abad.etc |
+-------------------------+----------+-------------------------
+-------------------------+
```

6. To get details for a specified role:

```
$ keystone role-get ROLE_ID
```

```
+----------+-----------------------------------+
| Property |               Value               |
+----------+-----------------------------------+
|    id    | bef1f95537914b1295da6aa038ef4de6 |
|   name   |              new-role             |
+----------+-----------------------------------+
```

7. To remove a role from a user-project pair:

```
$ keystone user-role-remove --user USER_ID --role ROLE_ID --
tenant TENANT_ID
```

8. To verify the role removal, run the following command:

```
$ keystone user-role-list --user USER_ID --tenant TENANT_ID
```

If the role was removed, the command output omits the removed role.

4.2. Manage project security

Security groups are sets of IP filter rules that are applied to all project instances, which define networking access to the instance. Group rules are project specific; project members can edit the default rules for their group and add new rule sets.

All projects have a "default" security group which is applied to any instance that has no other defined security group. Unless you change the default, this security group denies all incoming traffic and allows only outgoing traffic to your instance.

> **Note**
>
> For information about updating rules using the dashboard, see Section 2.9, “Manage project security”.

You can use the **`allow_same_net_traffic`** option in the **`/etc/nova/nova.conf`** file to globally control whether the rules apply to hosts which share a network.

If set to:

- **`True`** (default), hosts on the same subnet are not filtered and are allowed to pass all types of traffic between them. On a flat network, this allows all instances from all projects unfiltered communication. With VLAN networking, this allows access between instances within the same project. You can also simulate this setting by configuring the default security group to allow all traffic from the subnet.

- **False**, security groups are enforced for all connections.

Additionally, the number of maximum rules per security group is controlled by the **security_group_rules** and the number of allowed security groups per project is controlled by the **security_groups** quota (see Section 10, “Manage quotas”).

Procedure 3.1. List and view current security groups

From the command line you can get a list of security groups for the project, using the nova command:

1. Ensure your system variables are set for the user and tenant for which you are checking security group rules for. For example:

```
export OS_USERNAME=demo00
export OS_TENANT_NAME=tenant01
```

2. Output security groups, as follows:

```
$ nova secgroup-list
+---------+-------------+
| Name    | Description |
+---------+-------------+
| default | default     |
| open    | all ports   |
+---------+-------------+
```

3. View the details of a group, as follows:

```
$ nova secgroup-list-rules groupName
```

For example:

```
$ nova secgroup-list-rules open
+-------------+-----------+---------+-----------+------------
--+
 | IP Protocol | From Port | To Port | IP Range  | Source
Group |
 +-------------+-----------+---------+-----------+-----------
---+
 | icmp        | -1        | 255     | 0.0.0.0/0 |
|
 | tcp         | 1         | 65535   | 0.0.0.0/0 |
|
 | udp         | 1         | 65535   | 0.0.0.0/0 |
|
 +-------------+-----------+---------+-----------+-----------
---+
```

These rules are allow type rules as the default is deny. The first column is the IP protocol (one of icmp, tcp, or udp). The second and third columns specify the affected port range. The third column specifies the IP range in CIDR format. This example shows the full port range for all protocols allowed from all IPs.

Procedure 3.2. Create a security group

When adding a new security group, you should pick a descriptive but brief name. This name shows up in brief descriptions of the instances that use it where the longer description field often does not. For example, seeing that an instance is using security group "http" is much easier to understand than "bobs_group" or "secgrp1".

1. Ensure your system variables are set for the user and tenant for which you are checking security group rules for.

2. Add the new security group, as follows:

```
$ nova secgroup-create Group Name Description
```

For example:

```
$ nova secgroup-create global_http "Allows Web traffic
anywhere on the Internet."
+---------------------+-------------+-----------------------
----------------------+
| Id                  | Name        | Description
|
+---------------------+-------------+-----------------------
----------------------+
| 1578a08c-...dd9f23b | global_http | Allows Web traffic
anywhere on the Internet. |
+---------------------+-------------+-----------------------
----------------------+
```

3. Add a new group rule, as follows:

```
$ nova secgroup-add-rule secGroupName ip-protocol from-port
to-port CIDR
```

The arguments are positional, and the "from-port" and "to-port" arguments specify the local port range connections are allowed to access, not the source and destination ports of the connection. For example:

```
$ nova secgroup-add-rule global_http tcp 80 80 0.0.0.0/0
+-------------+-----------+---------+-----------+------------
--+
| IP Protocol | From Port | To Port | IP Range  | Source
Group |
+-------------+-----------+---------+-----------+------------
--+
| tcp         | 80        | 80      | 0.0.0.0/0 |
|
+-------------+-----------+---------+-----------+------------
--+
```

You can create complex rule sets by creating additional rules. For example, if you want to pass both http and https traffic, run:

```
$ nova secgroup-add-rule global_http tcp 443 443 0.0.0.0/0
+-------------+-----------+---------+-----------+------------
--+
| IP Protocol | From Port | To Port | IP Range  | Source
```

```
Group |
+-------------+-----------+---------+-----------+------------
--+
| tcp         | 443       | 443     | 0.0.0.0/0 |
|
+-------------+-----------+---------+-----------+------------
--+
```

Despite only outputting the newly added rule, this operation is additive (both rules are created and enforced).

4. View all rules for the new security group, as follows:

```
$ nova secgroup-list-rules global_http
+-------------+-----------+---------+-----------+------------
--+
| IP Protocol | From Port | To Port | IP Range  | Source
Group |
+-------------+-----------+---------+-----------+------------
--+
| tcp         | 80        | 80      | 0.0.0.0/0 |
|
| tcp         | 443       | 443     | 0.0.0.0/0 |
|
+-------------+-----------+---------+-----------+------------
--+
```

Procedure 3.3. Delete a security group

1. Ensure your system variables are set for the user and tenant for which you are deleting a security group for.

2. Delete the new security group, as follows:

```
$ nova secgroup-delete GroupName
```

For example:

```
$ nova secgroup-delete global_http
```

Procedure 3.4. Create security group rules for a cluster of instances

Source Groups are a special, dynamic way of defining the CIDR of allowed sources. The user specifies a Source Group (Security Group name), and all the users' other Instances using the specified Source Group are selected dynamically. This alleviates the need for individual rules to allow each new member of the cluster.

1. Make sure to set the system variables for the user and tenant for which you are deleting a security group for.

2. Add a source group, as follows:

```
$ nova secgroup-add-group-rule secGroupName source-group ip-
protocol from-port to-port
```

For example:

```
$ nova secgroup-add-group-rule cluster global_http tcp 22 22
```

The **cluster** rule allows ssh access from any other instance that uses the **global_http** group.

5. Manage services

5.1. Create and manage services and service users

The Identity Service enables you to define services, as follows:

- Service catalog template. The Identity Service acts as a service catalog of endpoints for other OpenStack services. The **etc/default_catalog.templates** template file defines the endpoints for services. When the Identity Service uses a template file back end, any changes that are made to the endpoints are cached. These changes do not persist when you restart the service or reboot the machine.
- A SQL back end for the catalog service. When the Identity Service is online, you must add the services to the catalog. When you deploy a system for production, use the SQL back end.

The **auth_token** middleware supports the use of either a shared secret or users for each service.

To authenticate users against the Identity Service, you must create a service user for each OpenStack service. For example, create a service user for the Compute, Block Storage, and Networking services.

To configure the OpenStack services with service users, create a project for all services and create users for each service. Assign the admin role to each service user and project pair. This role enables users to validate tokens and authenticate and authorize other user requests.

5.1.1. Create a service

1. List the available services:

```
$ keystone service-list
```

```
+----------------------------------+----------+----------+---
------------------------+
|                id                |   name   |   type   |
description        |
+----------------------------------+----------+----------+---
------------------------+
| 9816f1faaa7c4842b90fb4821cd09223 |  cinder  |  volume  |
Cinder Volume Service   |
| da8cf9f8546b4a428c43d5e032fe4afc |   ec2    |   ec2    |
EC2 Compatibility Layer  |
| 5f105eeb55924b7290c8675ad7e294ae |  glance  |  image   |
Glance Image Service   |
| dcaa566e912e4c0e900dc86804e3dde0 | keystone | identity |
```

```
Keystone Identity Service |
| 4a715cfbc3664e9ebf388534ff2be76a |   nova   | compute  |
Nova Compute Service   |
| 6feb2e0b98874d88bee221974770e372 |    s3    |    s3    |
S3            |
+----------------------------------+----------+----------+---
------------------------+
```

2. To create a service, run this command:

```
$ keystone service-create --name service_name --type
service_type --description service_description
```

The arguments are:

- ***service_name***. The unique name of the new service.
- ***service_type***. The service type, such as **identity**, **compute**, **network**, **image**, **object-store** or any other service identifier string.
- ***service_description***. The description of the service.

For example, to create a **swift** service of type **object-store**, run this command:

```
$ keystone service-create --name swift --type object-store --
description "object store service"
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       object store service       |
|      id     | 84c23f4b942c44c38b9c42c5e517cd9a |
|     name    |              swift               |
|     type    |           object-store           |
+-------------+----------------------------------+
```

3. To get details for a service, run this command:

```
$ keystone service-get service_ID
```

For example:

```
$ keystone service-get 84c23f4b942c44c38b9c42c5e517cd9a
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       object store service       |
|      id     | 84c23f4b942c44c38b9c42c5e517cd9a |
|     name    |              swift               |
|     type    |           object-store           |
+-------------+----------------------------------+
```

5.1.2. Create service users

1. Create a project for the service users. Typically, this project is named **service**, but choose any name you like:

   ```
   $ keystone tenant-create --name service
   ```

   The output shows the ID for the project.

   Make a note of this ID. You need it to create service users and assign roles.

   ```
   +-------------+----------------------------------+
   |   Property  |              Value               |
   +-------------+----------------------------------+
   | description |                                  |
   |   enabled   |               True               |
   |      id     | 3e9f3f5399624b2db548d7f871bd5322 |
   |     name    |              service             |
   +-------------+----------------------------------+
   ```

2. Create service users for the relevant services for your deployment.

3. To assign the admin role to the service user-project pairs, run this command to get the ID of the admin role:

   ```
   $ keystone role-list
   ```

   ```
   +----------------------------------+---------------+
   |                id                |      name     |
   +----------------------------------+---------------+
   | 71ccc37d41c8491c975ae72676db687f |     Member    |
   | 149f50a1fe684bfa88dae76a48d26ef7 | ResellerAdmin |
   | 9fe2ff9ee4384b1894a90878d3e92bab |    _member_   |
   | 6ecf391421604da985db2f141e46a7c8 |     admin     |
   | deb4fffd123c4d02a907c2c74559dccf |  anotherrole  |
   | bef1f95537914b1295da6aa038ef4de6 |    new-role   |
   +----------------------------------+---------------+
   ```

4. Assign the admin role to the user-project pair:

   ```
   $ keystone user-role-add --user SERVICE_USER_ID --role
   ADMIN_ROLE_ID --tenant SERVICE_PROJECT_ID
   ```

5.1.3. Delete a service

To delete a specified service, specify its ID:

```
$ keystone service-delete SERVICE_ID
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       object store service       |
```

```
|      id      | 84c23f4b942c44c38b9c42c5e517cd9a |
|     name     |              swift               |
|     type     |          object-store            |
+--------------+----------------------------------+
```

5.2. Manage Compute services

You can enable and disable Compute services. The following examples disable and enable the **nova-compute** service.

1. List the Compute services:

```
$ nova service-list
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
| Binary           | Host     | Zone     | Status  | State |
Updated_at                 | Disabled Reason |
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
| nova-conductor   | devstack | internal | enabled | up    |
2013-10-16T00:56:08.000000 | None            |
| nova-cert        | devstack | internal | enabled | up    |
2013-10-16T00:56:09.000000 | None            |
| nova-compute     | devstack | nova     | enabled | up    |
2013-10-16T00:56:07.000000 | None            |
| nova-network     | devstack | internal | enabled | up    |
2013-10-16T00:56:06.000000 | None            |
| nova-scheduler   | devstack | internal | enabled | up    |
2013-10-16T00:56:04.000000 | None            |
| nova-consoleauth | devstack | internal | enabled | up    |
2013-10-16T00:56:07.000000 | None            |
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
```

2. Disable a nova service:

```
$ nova service-disable localhost.localdomain nova-compute --
reason 'trial log'
+----------+--------------+----------+------------------+
| Host     | Binary       | Status   | Disabled Reason  |
+----------+--------------+----------+------------------+
| devstack | nova-compute | disabled | Trial log        |
+----------+--------------+----------+------------------+
```

3. Check the service list:

```
$ nova service-list
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
| Binary           | Host     | Zone     | Status  | State |
Updated_at                 | Disabled Reason |
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
| nova-conductor   | devstack | internal | enabled | up    |
```

```
2013-10-16T00:56:48.000000 | None            |
| nova-cert        | devstack | internal | enabled  | up    |
2013-10-16T00:56:49.000000 | None            |
| nova-compute     | devstack | nova     | disabled | up    |
2013-10-16T00:56:47.000000 | Trial log       |
| nova-network     | devstack | internal | enabled  | up    |
2013-10-16T00:56:51.000000 | None            |
| nova-scheduler   | devstack | internal | enabled  | up    |
2013-10-16T00:56:44.000000 | None            |
| nova-consoleauth | devstack | internal | enabled  | up    |
2013-10-16T00:56:47.000000 | None            |
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
```

4. Enable the service:

```
$ nova service-enable localhost.localdomain nova-compute
+----------+--------------+---------+
| Host     | Binary       | Status  |
+----------+--------------+---------+
| devstack | nova-compute | enabled |
+----------+--------------+---------+
```

5. Check the service list:

```
$ nova service-list
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
| Binary           | Host     | Zone     | Status  | State |
Updated_at                 | Disabled Reason |
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
| nova-conductor   | devstack | internal | enabled | up    |
2013-10-16T00:57:08.000000 | None            |
| nova-cert        | devstack | internal | enabled | up    |
2013-10-16T00:57:09.000000 | None            |
| nova-compute     | devstack | nova     | enabled | up    |
2013-10-16T00:57:07.000000 | None            |
| nova-network     | devstack | internal | enabled | up    |
2013-10-16T00:57:11.000000 | None            |
| nova-scheduler   | devstack | internal | enabled | up    |
2013-10-16T00:57:14.000000 | None            |
| nova-consoleauth | devstack | internal | enabled | up    |
2013-10-16T00:57:07.000000 | None            |
+------------------+----------+----------+---------+-------+-
---------------------------+-----------------+
```

6. Manage images

The cloud operator assigns roles to users. Roles determine who can upload and manage images. The operator might restrict image upload and management to only cloud administrators or operators.

You can upload images through the **`glance`** client or the Image Service API. You can also use the **`nova`** client to list images, set and delete image metadata, delete images, and take a

snapshot of a running instance to create an image. After you upload an image, you cannot change it.

6.1. List or get details for images (glance)

To get a list of images and to then get further details about a single image, use **`glance image-list`** and **`glance image-show`**.

```
$ glance image-list
+--------------------------------------+---------------------------
------+-------------+------------------+----------+--------+
| ID                                   | Name
| Disk Format | Container Format | Size     | Status |
+--------------------------------------+---------------------------
------+-------------+------------------+----------+--------+
| 397e713c-b95b-4186-ad46-6126863ea0a9 | cirros-0.3.2-x86_64-uec
| ami         | ami              | 25165824 | active |
| df430cc2-3406-4061-b635-a51c16e488ac | cirros-0.3.2-x86_64-uec-
kernel  | aki         | aki              | 4955792  | active |
| 3cf852bd-2332-48f4-9ae4-7d926d50945e | cirros-0.3.2-x86_64-uec-
ramdisk | ari         | ari              | 3714968  | active |
| 7e5142af-1253-4634-bcc6-89482c5f2e8a | myCirrosImage
| ami         | ami              | 14221312 | active |
+--------------------------------------+---------------------------
------+-------------+------------------+----------+--------+
```

```
$ glance image-show myCirrosImage

+--------------------------------------+---------------------------
------------+
| Property                             | Value
|
+--------------------------------------+---------------------------
------------+
| Property 'base_image_ref'            | 397e713c-b95b-4186-ad46-
6126863ea0a9 |
| Property 'image_location'            | snapshot
|
| Property 'image_state'               | available
|
| Property 'image_type'                | snapshot
|
| Property 'instance_type_ephemeral_gb' | 0
|
| Property 'instance_type_flavorid'    | 2
|
| Property 'instance_type_id'          | 5
|
| Property 'instance_type_memory_mb'   | 2048
|
| Property 'instance_type_name'        | m1.small
|
| Property 'instance_type_root_gb'     | 20
|
| Property 'instance_type_rxtx_factor' | 1
```

```
|
| Property 'instance_type_swap'         | 0
|
| Property 'instance_type_vcpu_weight'  | None
|
| Property 'instance_type_vcpus'        | 1
|
| Property 'instance_uuid'              | 84c6e57d-a6b1-44b6-81eb-
fcb36afd31b5 |
| Property 'kernel_id'                  | df430cc2-3406-4061-b635-
a51c16e488ac |
| Property 'owner_id'                   |
66265572db174a7aa66eba661f58eb9e     |
| Property 'ramdisk_id'                 | 3cf852bd-2332-48f4-9ae4-
7d926d50945e |
| Property 'user_id'                    |
376744b5910b4b4da7d8e6cb483b06a8     |
| checksum                              |
8e4838effa1969ad591655d6485c7ba8     |
| container_format                      | ami
|
| created_at                            | 2013-07-22T19:45:58
|
| deleted                               | False
|
| disk_format                           | ami
|
| id                                    | 7e5142af-1253-4634-bcc6-
89482c5f2e8a |
| is_public                             | False
|
| min_disk                              | 0
|
| min_ram                               | 0
|
| name                                  | myCirrosImage
|
| owner                                 |
66265572db174a7aa66eba661f58eb9e     |
| protected                             | False
|
| size                                  | 14221312
|
| status                                | active
|
| updated_at                            | 2013-07-22T19:46:42
|
+--------------------------------------+--------------------------
------------+
```

When viewing a list of images, you can also use **grep** to filter the list, as follows:

```
$ glance image-list | grep 'cirros'
| 397e713c-b95b-4186-ad46-6126863ea0a9 | cirros-0.3.2-x86_64-uec
| ami         | ami               | 25165824 | active |
| df430cc2-3406-4061-b635-a51c16e488ac | cirros-0.3.2-x86_64-uec-
```

```
kernel   | aki          | aki                | 4955792   | active |
| 3cf852bd-2332-48f4-9ae4-7d926d50945e | cirros-0.3.2-x86_64-uec-
ramdisk | ari          | ari                | 3714968   | active |
```

Note

To store location metadata for images, which enables direct file access for a client, update the **/etc/glance/glance.conf** file with the following statements:

- **show_multiple_locations = True**
- **filesystem_store_metadata_file = *filePath***, where *filePath* points to a JSON file that defines the mount point for OpenStack images on your system and a unique ID. For example:

```
[{
    "id": "2d9bb53f-70ea-4066-a68b-67960eaae673",
    "mountpoint": "/var/lib/glance/images/"
}]
```

After you restart the Image Service, you can use the following syntax to view the image's location information:

```
$ glance --os-image-api-version=2 image-show imageID
```

For example, using the image ID shown above, you would issue the command as follows:

```
$ glance --os-image-api-version=2 image-show 2d9bb53f-70ea-4066-
a68b-67960eaae673
```

6.2. Create or update an image (glance)

To create an image, use **glance image-create**:

```
$ glance image-create imageName
```

To update an image by name or ID, use **glance image-update**:

```
$ glance image-update imageName
```

The following table lists the optional arguments that you can use with the **create** and **update** commands to modify image properties. For more information, see the Image Service chapter in the *Command Line Reference Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

--name NAME	The name of the image.

--disk-format DISK_FORMAT	The disk format of the image. Acceptable formats are ami, ari, aki, vhd, vmdk, raw, qcow2, vdi, and iso.
--container-format CONTAINER_FORMAT	The container format of the image. Acceptable formats are ami, ari, aki, bare, and ovf.
--owner TENANT_ID	The tenant who should own the image.
--size SIZE	The size of image data, in bytes.
--min-disk DISK_GB	The minimum size of the disk needed to boot the image, in gigabytes.
--min-ram DISK_RAM	The minimum amount of RAM needed to boot the image, in megabytes.
--location IMAGE_URL	The URL where the data for this image resides. For example, if the image data is stored in swift, you could specify **`swift://account:key@example.com/container/obj`**.
--file FILE	Local file that contains the disk image to be uploaded during the update. Alternatively, you can pass images to the client through stdin.
--checksum CHECKSUM	Hash of image data to use for verification.
--copy-from IMAGE_URL	Similar to ***--location*** in usage, but indicates that the image server should immediately copy the data and store it in its configured image store.
--is-public [True|False]	Makes an image accessible for all the tenants.
--is-protected [True|False]	Prevents an image from being deleted.

--property KEY=VALUE	Arbitrary property to associate with image. This option can be used multiple times.
--purge-props	Deletes all image properties that are not explicitly set in the update request. Otherwise, those properties not referenced are preserved.
--human-readable	Prints the image size in a human-friendly format.

The following example shows the command that you would use to upload a CentOS 6.3 image in qcow2 format and configure it for public access:

```
$ glance image-create --name centos63-image --disk-format=qcow2 \
 --container-format=bare --is-public=True --file=./centos63.qcow2
```

The following example shows how to update an existing image with a properties that describe the disk bus, the CD-ROM bus, and the VIF model:

```
$ glance image-update \
 --property hw_disk_bus=scsi \
 --property hw_cdrom_bus=ide \
 --property hw_vif_model=e1000 \
 f16-x86_64-openstack-sda
```

Currently the libvirt virtualization tool determines the disk, CD-ROM, and VIF device models based on the configured hypervisor type (**`libvirt_type`** in **`/etc/nova/nova.conf`**). For the sake of optimal performance, libvirt defaults to using virtio for both disk and VIF (NIC) models. The disadvantage of this approach is that it is not possible to run operating systems that lack virtio drivers, for example, BSD, Solaris, and older versions of Linux.

If you specify a disk or CD-ROM bus model that is not supported, see Table 3.2, “Disk and CD-ROM bus model values”. If you specify a VIF model that is not supported, the instance fails to launch. See Table 3.3, “VIF model values”.

The valid model values depend on the **`libvirt_type`** setting, as shown in the following tables.

Table 3.2. Disk and CD-ROM bus model values

libvirt_type setting	Supported model values
qemu or kvm	➢ virtio ➢ scsi ➢ ide ➢ virtio

Table 3.3. VIF model values

libvirt_type setting	Supported model values
qemu or kvm	▸ virtio ▸ ne2k_pci ▸ pcnet ▸ rtl8139 ▸ e1000
vmware	▸ VirtualE1000 ▸ VirtualPCNet32 ▸ VirtualVmxnet

6.3. Create an image (nova)

You can use the **nova** client to take a snapshot of a running instance to create an image.

To minimize the potential for data loss and ensure that you create an accurate image, you should shut down the instance before you take a snapshot.

You cannot create a snapshot from an instance that has an attached volume. Detach the volume, create the image, and remount the volume.

1. Write any buffered data to disk.

   For more information on taking snapshots, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

2. List instances to get the server name:

```
$ nova list
+--------------------------------------+---------------------
-+--------+------------+-------------+------------------+
| ID                                   | Name
| Status | Task State | Power State | Networks         |
+--------------------------------------+---------------------
-+--------+------------+-------------+------------------+
| 84c6e57d-a6b1-44b6-81eb-fcb36afd31b5 | myCirrosServer
| ACTIVE | None       | Running     | private=10.0.0.3 |
+--------------------------------------+---------------------
-+--------+------------+-------------+------------------+
```

   In this example, the server is named **myCirrosServer**.

3. Use this server to create a snapshot:

```
$ nova image-create myCirrosServer myCirrosImg
```

The command creates a qemu snapshot and automatically uploads the image to your repository. Only the tenant that creates the image has access to it.

4. Get details for your image to check its status:

```
$ nova image-show myCirrosImg

+----------------------+--------------------------------------
-+
| Property             | Value
|
+----------------------+--------------------------------------
-+
| OS-EXT-IMG-SIZE:size | 13147648
|
| created              | 2014-10-22T01:47:45Z
|
| id                   | 901d3823-2be1-4ce3-b9b4-56330d56fd45
|
| minDisk              | 0
|
| minRam               | 0
|
| name                 | myCirrosImg
|
| progress             | 100
|
| status               | ACTIVE
|
| updated              | 2014-10-22T01:49:25Z
|
+----------------------+--------------------------------------
-+
```

The image status changes from **SAVING** to **ACTIVE**. Only the tenant who creates the image has access to it.

To launch an instance from your image, include the image ID, flavor ID, and network ID (if more than one network has been set up). For example:

```
$ nova boot newServer --image 7e5142af-1253-4634-bcc6-89482c5f2e8a \
--flavor 3 --nic net-id=3d706957-7696-4aa8-973f-b80892ff9a95
+--------------------------------------+--------------------------
------------------------+
| Property                             | Value
|
+--------------------------------------+--------------------------
------------------------+
| OS-DCF:diskConfig                    | MANUAL
|
| OS-EXT-AZ:availability_zone          | nova
|
| OS-EXT-SRV-ATTR:host                 | -
|
```

```
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -
|
| OS-EXT-SRV-ATTR:instance_name        | instance-00000002
|
| OS-EXT-STS:power_state               | 0
|
| OS-EXT-STS:task_state                | scheduling
|
| OS-EXT-STS:vm_state                  | building
|
| OS-SRV-USG:launched_at               | -
|
| OS-SRV-USG:terminated_at             | -
|
| accessIPv4                           |
|
| accessIPv6                           |
|
| adminPass                            | VptbvZmS6pbX
|
| config_drive                         |
|
| created                              | 2014-10-22T02:09:13Z
|
| flavor                               | m1.medium (3)
|
| hostId                               |
|
| id                                   | 4e02a9dd-8833-4fe8-ae5d-
dcf0cd83ebcc               |
| image                                | myCirrosImg (901d3823-
2be1-4ce3-b9b4-56330d56fd45) |
| key_name                             | -
|
| metadata                             | {}
|
| name                                 | newServer
|
| os-extended-volumes:volumes_attached | []
|
| progress                             | 0
|
| security_groups                      | default
|
| status                               | BUILD
|
| tenant_id                            |
a419abe86e224a518472a8ab9a3f3d4c                    |
| updated                              | 2014-10-22T02:09:14Z
|
| user_id                              |
35c32e9fa2f14bb89556cea01e72c413                    |
+--------------------------------------+---------------------------
-------------------------+
```

6.4. Troubleshoot image creation

6.11. Troubleshoot image creation

If you encounter problems in creating an image in Image Service or Compute, the following information may help you troubleshoot the creation process.

- Ensure that the version of qemu you are using is version 0.14 or later. Earlier versions of qemu result in an **`unknown option -s`** error message in the **`nova-compute.log`** file.
- Examine the **`/var/log/nova-api.log`** and **`/var/log/nova-compute.log`** log files for error messages.

7. Manage volumes

A volume is a detachable block storage device, similar to a USB hard drive. You can attach a volume to only one instance. To create and manage volumes, you use a combination of **`nova`** and **`cinder`** client commands.

7.1. Migrate a volume

As an administrator, you can migrate a volume with its data from one location to another in a manner that is transparent to users and workloads. You can migrate only detached volumes with no snapshots.

Possible use cases for data migration include:

- Bring down a physical storage device for maintenance without disrupting workloads.
- Modify the properties of a volume.
- Free up space in a thinly-provisioned back end.

Migrate a volume with the **`cinder migrate`** command, as shown in the following example:

```
$ cinder migrate volumeID destinationHost --force-host-
copy=True|False
```

In this example, **`--force-host-copy=True`** forces the generic host-based migration mechanism and bypasses any driver optimizations.

> **Note**
>
> If the volume is in use or has snapshots, the specified host destination cannot accept the volume. If the user is not an administrator, the migration fails.

7.2. Set a volume to read-only access

To give multiple users shared, secure access to the same data, you can set a volume to read-only access.

Run the following command to set a volume to read-only access:

```
$ cinder read-only-mode-update VOLUME BOOLEAN
```

VOLUME is the ID of the target volume and *BOOLEAN* is a flag that enables read-only or read/write access to the volume.

The following values for *BOOLEAN* are valid:

- **`true`**. Sets the read-only flag in the volume. When you attach the volume to an instance, the instance checks for this flag to determine whether to restrict volume access to read-only.
- **`false`**. Sets the volume to read/write access.

8. Manage flavors

In OpenStack, flavors define the compute, memory, and storage capacity of nova computing instances. To put it simply, a flavor is an available hardware configuration for a server. It defines the “size” of a virtual server that can be launched.

> **Note**
>
> Flavors can also determine on which compute host a flavor can be used to launch an instance. For information about customizing flavors, see the *Cloud Administration Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

A flavor consists of the following parameters:

Flavor ID

Automatically generated by OpenStack. For private flavors, a value from 1 to 255.

Name

Name for the new flavor.

VCPUs

Number of virtual CPUs to use.

Memory MB

Amount of RAM to use (in megabytes).

Root Disk GB

Amount of disk space (in gigabytes) to use for the root (/) partition.

Ephemeral Disk GB

Amount of disk space (in gigabytes) to use for the ephemeral partition. If unspecified, the value is 0 by default.

Ephemeral disks offer machine local disk storage linked to the life cycle of a VM instance. When a VM is terminated, all data on the ephemeral disk is lost. Ephemeral disks are not included in any snapshots.

Swap

Amount of swap space (in megabytes) to use. If unspecified, the value is 0 by default.

RXTX_Factor

Bandwidth cap (versus the network's default value to which the flavor's instance is attached). This factor is multiplied by the **rxtx_base** network property.

The default flavors are:

- m1.tiny (1 VCPU/1 GB Disk/512 MB RAM)
- m1.small (1 VCPU/20 GB Disk/2048 MB RAM)
- m1.medium (2 VCPU/40 GB Disk/4096 MB RAM)
- m1.large (4 VCPU/80 GB Disk/8192 MB RAM)
- m1.xlarge (8 VCPU/160 GB Disk/16384 MB RAM)

You can create and manage flavors with the nova **flavor-*** commands provided by the **python-novaclient** package.

8.1. Create a flavor

1. List flavors to show the ID and name, the amount of memory, the amount of disk space for the root partition and for the ephemeral partition, the swap, bandwidth cap, and the number of virtual CPUs for each flavor.

   ```
   $ nova flavor-list
   ```

2. To create a flavor, specify a name, ID, RAM size, disk size, and the number of VCPUs for the flavor, as follows:

   ```
   $ nova flavor-create FLAVOR_NAME FLAVOR_ID RAM_IN_MB
   ROOT_DISK_IN_GB NUMBER_OF_VCPUS
   ```

   > **Note**
   >
   > The flavor ID is a number from 1 to 255 and cannot contain special characters or spaces.

   For a list of optional parameters, run this command:

   ```
   $ nova help flavor-create
   ```

3. After you create a flavor, assign it to a project by specifying the flavor name or ID and the tenant ID:

   ```
   $ nova flavor-access-add FLAVOR TENANT_ID
   ```

8.2. Delete a flavor

- Delete a specified flavor, as follows:

  ```
  $ nova flavor-delete FLAVOR_ID
  ```

9. Manage the OpenStack environment

This section includes tasks specific to the OpenStack environment.

9.1. Select hosts where instances are launched

With the appropriate permissions, you can select on which host instances are launched and which roles can boot instances on this host.

1. To select the host where instances are launched, use the **--availability_zone *zone*:*host*** parameter on the **nova boot** command. For example:

```
$ nova boot --image <uuid> --flavor m1.tiny --nic net-id=
<networkID> --key_name test --availability-zone nova:server2
```

2. To specify which roles can launch an instance on a specified host, enable the **create:forced_host** option in the **policy.json** file. By default, this option is enabled for only the admin role.

3. To view the list of valid compute hosts, use the **nova hypervisor-list** command, as follows:

```
$ nova hypervisor-list
+----+---------------------+
| ID | Hypervisor hostname |
+----+---------------------+
| 1  | server2             |
| 2  | server3             |
| 3  | server4             |
+----+---------------------+
```

9.2. Evacuate instances

If a cloud compute node fails due to a hardware malfunction or another reason, you can evacuate instances to make them available again. You can choose evacuation parameters for your use case.

To preserve user data on server disk, you must configure shared storage on the target host. Also, you must validate that the current VM host is down; otherwise, the evacuation fails with an error.

1. To list hosts and find a different host for the evacuated instance, run:

```
$ nova host-list
```

2. Evacuate the instance. You can pass the instance password to the command by using the **--password <pwd>** option. If you do not specify a password, one is generated and printed after the command finishes successfully. The following command evacuates a server without shared storage from a host that is down to the specified *host_b*:

```
$ nova evacuate evacuated_server_name host_b
```

The instance is booted from a new disk, but preserves its configuration including its ID, name, uid, IP address, and so on. The command returns a password:

```
+-----------+--------------+
| Property  |    Value     |
+-----------+--------------+
| adminPass | kRAJpErnT4xZ |
+-----------+--------------+
```

3. To preserve the user disk data on the evacuated server, deploy OpenStack Compute with a shared file system. To configure your system, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/. In the following example, the password remains unchanged:

```
$ nova evacuate evacuated_server_name host_b --on-shared-
storage
```

9.3. Manage IP addresses

Each instance has a private, fixed IP address (assigned when launched) and can also have a public, or floating, address. Private IP addresses are used for communication between instances, and public addresses are used for communication with networks outside the cloud, including the Internet.

- By default, both administrative and end users can associate floating IP addresses with projects and instances. You can change user permissions for managing IP addresses by updating the **/etc/nova/policy.json** file. For basic floating-IP procedures, see the *Manage IP Addresses* section in the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.
- For details on creating public networks using OpenStack Networking (**neutron**), see the *Cloud Administration Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/. No floating IP addresses are created by default in OpenStack Networking.

As an administrator using legacy networking (**nova-network**), you can use the following bulk commands to list, create, and delete ranges of floating IP addresses. These addresses can then be associated with instances by end users:

List addresses for all projects

To list all floating IP addresses for all projects, run:

```
$ nova floating-ip-bulk-list
+------------+---------------+---------------+--------+------
-----+
| project_id | address       | instance_uuid | pool   |
interface |
+------------+---------------+---------------+--------+------
-----+
| None       | 172.24.4.225  | None          | public | eth0
|
| None       | 172.24.4.226  | None          | public | eth0
```

```
|
| None        | 172.24.4.227  | None           | public | eth0
|
| None        | 172.24.4.228  | None           | public | eth0
|
| None        | 172.24.4.229  | None           | public | eth0
|
| None        | 172.24.4.230  | None           | public | eth0
|
| None        | 172.24.4.231  | None           | public | eth0
|
| None        | 172.24.4.232  | None           | public | eth0
|
| None        | 172.24.4.233  | None           | public | eth0
|
| None        | 172.24.4.234  | None           | public | eth0
|
| None        | 172.24.4.235  | None           | public | eth0
|
| None        | 172.24.4.236  | None           | public | eth0
|
| None        | 172.24.4.237  | None           | public | eth0
|
| None        | 172.24.4.238  | None           | public | eth0
|
| None        | 192.168.253.1 | None           | test   | eth0
|
| None        | 192.168.253.2 | None           | test   | eth0
|
| None        | 192.168.253.3 | None           | test   | eth0
|
| None        | 192.168.253.4 | None           | test   | eth0
|
| None        | 192.168.253.5 | None           | test   | eth0
|
| None        | 192.168.253.6 | None           | test   | eth0
|
+------------+---------------+---------------+--------+------
-----+
```

Bulk create floating IP addresses

To create a range of floating IP addresses, run:

```
$ nova floating-ip-bulk-create [--pool POOL_NAME] [--interface
INTERFACE] RANGE_TO_CREATE
```

For example:

```
$ nova floating-ip-bulk-create --pool test 192.168.1.56/29
```

By default, **`floating-ip-bulk-create`** uses the **`public`** pool and **`eth0`** interface values.

> **Note**
>
> You should use a range of free IP addresses that is correct for your network. If you are not sure, at least try to avoid the DHCP address range:
>
> - Pick a small range (/29 gives an 8 address range, 6 of which will be usable)
> - Use **nmap** to check a range's availability. For example, 192.168.1.56/29 represents a small range of addresses (192.168.1.56-63, with 57-62 usable), and you could run the command **nmap -sn 192.168.1.56/29** to check whether the entire range is currently unused.

Bulk delete floating IP addresses

To delete a range of floating IP addresses, run:

```
$ nova floating-ip-bulk-delete RANGE_TO_DELETE
```

For example:

```
$ nova floating-ip-bulk-delete 192.168.1.56/29
```

9.4. Launch and manage stacks

The Orchestration service provides a template-based orchestration engine for the OpenStack cloud, which can be used to create and manage cloud infrastructure resources such as storage, networking, instances, and applications as a repeatable running environment.

Templates are used to create stacks, which are collections of resources. For example, a stack might include instances, floating IPs, volumes, security groups, or users. The Orchestration service offers access to all OpenStack core services via a single modular template, with additional orchestration capabilities such as auto-scaling and basic high availability.

For information about the:

- Basic creation and deletion of Orchestration stacks, see the *End User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.
- **heat** CLI commands, see the *Command Line Reference Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

As an administrator, you can also carry out stack functions on behalf of your users. For example, to resume, suspend, or delete a stack, run:

```
$ action-resume stackID
$ action-suspend stackID
$ stack-delete stackID
```

10. Manage quotas

To prevent system capacities from being exhausted without notification, you can set up quotas. Quotas are operational limits. For example, the number of gigabytes allowed for each tenant can be controlled so that cloud resources are optimized. Quotas can be enforced at both the tenant (or project) and the tenant-user level.

Using the command-line interface, you can manage quotas for the OpenStack Compute service, the OpenStack Block Storage service, and the OpenStack Networking service.

The cloud operator typically changes default values because a tenant requires more than ten Block Storage volumes or 1 TB on a compute node.

Note

To view all tenants (projects), run:

```
$ keystone tenant-list
+----------------------------------+----------+---------+
|                id                |   name   | enabled |
+----------------------------------+----------+---------+
| e66d97ac1b704897853412fc8450f7b9 |  admin   |   True  |
| bf4a37b885fe46bd86e999e50adad1d3 | services |   True  |
| 21bd1c7c95234fd28f589b60903606fa | tenant01 |   True  |
| f599c5cd1cba4125ae3d7caed08e288c | tenant02 |   True  |
+----------------------------------+----------+---------+
```

To display all current users for a tenant, run:

```
$ keystone user-list --tenant-id tenantID
+----------------------------------+--------+---------+-------+
|                id                |  name  | enabled | email |
+----------------------------------+--------+---------+-------+
| ea30aa434ab24a139b0e85125ec8a217 | demo00 |   True  |       |
| 4f8113c1d838467cad0c2f337b3dfded | demo01 |   True  |       |
+----------------------------------+--------+---------+-------+
```

10.1. Manage Compute service quotas

As an administrative user, you can use the **`nova quota-*`** commands, which are provided by the **`python-novaclient`** package, to update the Compute service quotas for a specific tenant or tenant user, as well as update the quota defaults for a new tenant.

Table 3.4. Compute quota descriptions

Quota name	Description
`cores`	Number of instance cores (VCPUs) allowed per tenant.

Quota name	Description
fixed-ips	Number of fixed IP addresses allowed per tenant. This number must be equal to or greater than the number of allowed instances.
floating-ips	Number of floating IP addresses allowed per tenant.
injected-file-content-bytes	Number of content bytes allowed per injected file.
injected-file-path-bytes	Length of injected file path.
injected-files	Number of injected files allowed per tenant.
instances	Number of instances allowed per tenant.
key-pairs	Number of key pairs allowed per user.
metadata-items	Number of metadata items allowed per instance.
ram	Megabytes of instance ram allowed per tenant.
security-groups	Number of security groups per tenant.
security-group-rules	Number of rules per security group.

10.1.1. View and update Compute quotas for a tenant (project)

Procedure 3.5. To view and update default quota values

1. List all default quotas for all tenants, as follows:

   ```
   $ nova quota-defaults
   ```

   For example:

   –

```
$ nova quota-defaults
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 10    |
| fixed_ips                   | -1    |
| metadata_items              | 128   |
| injected_files              | 5     |
| injected_file_content_bytes | 10240 |
| injected_file_path_bytes    | 255   |
| key_pairs                   | 100   |
| security_groups             | 10    |
| security_group_rules        | 20    |
+-----------------------------+-------+
```

2. Update a default value for a new tenant, as follows:

```
$ nova quota-class-update --key value default
```

For example:

```
$ nova quota-class-update --instances 15 default
```

Procedure 3.6. To view quota values for an existing tenant (project)

1. Place the tenant ID in a usable variable, as follows:

```
$ tenant=$(keystone tenant-list | awk '/tenantName/ {print
$2}')
```

2. List the currently set quota values for a tenant, as follows:

```
$ nova quota-show --tenant $tenant
```

For example:

```
$ nova quota-show --tenant $tenant
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 10    |
| fixed_ips                   | -1    |
| metadata_items              | 128   |
| injected_files              | 5     |
| injected_file_content_bytes | 10240 |
| injected_file_path_bytes    | 255   |
```

```
| key_pairs                    | 100   |
| security_groups              | 10    |
| security_group_rules         | 20    |
+------------------------------+-------+
```

Procedure 3.7. To update quota values for an existing tenant (project)

1. Obtain the tenant ID, as follows:

```
$ tenant=$(keystone tenant-list | awk '/tenantName/ {print
$2}')
```

2. Update a particular quota value, as follows:

```
$ nova quota-update --quotaName quotaValue tenantID
```

For example:

```
$ nova quota-update --floating-ips 20 $tenant
$ nova quota-show --tenant $tenant
+------------------------------+-------+
| Quota                        | Limit |
+------------------------------+-------+
| instances                    | 10    |
| cores                        | 20    |
| ram                          | 51200 |
| floating_ips                 | 20    |
| fixed_ips                    | -1    |
| metadata_items               | 128   |
| injected_files               | 5     |
| injected_file_content_bytes  | 10240 |
| injected_file_path_bytes     | 255   |
| key_pairs                    | 100   |
| security_groups              | 10    |
| security_group_rules         | 20    |
+------------------------------+-------+
```

Note

To view a list of options for the **quota-update** command, run:

```
$ nova help quota-update
```

10.1.2. View and update Compute quotas for a tenant user

Procedure 3.8. To view quota values for a tenant user

1. Place the user ID in a usable variable, as follows:

```
$ tenantUser=$(keystone user-list | awk '/userName/ {print
$2}')
```

2. Place the user's tenant ID in a usable variable, as follows:

```
$ tenant=$(keystone tenant-list | awk '/tenantName/ {print
$2}')
```

3. List the currently set quota values for a tenant user, as follows:

```
$ nova quota-show --user $tenantUser --tenant $tenant
```

For example:

```
$ nova quota-show --user $tenantUser --tenant $tenant
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 20    |
| fixed_ips                   | -1    |
| metadata_items              | 128   |
| injected_files              | 5     |
| injected_file_content_bytes | 10240 |
| injected_file_path_bytes    | 255   |
| key_pairs                   | 100   |
| security_groups             | 10    |
| security_group_rules        | 20    |
+-----------------------------+-------+
```

Procedure 3.9. To update quota values for a tenant user

1. Place the user ID in a usable variable, as follows:

```
$ tenantUser=$(keystone user-list | awk '/userName/ {print
$2}')
```

2. Place the user's tenant ID in a usable variable, as follows:

```
$ tenant=$(keystone tenant-list | awk '/userName/ {print $2}')
```

3. Update a particular quota value, as follows:

```
$ nova quota-update --user $tenantUser --quotaName quotaValue
$tenant
```

For example:

```
$ nova quota-update --user $tenantUser --floating-ips 12
$tenant
$ nova quota-show --user $tenantUser --tenant $tenant
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
```

```
| cores                        | 20    |
| ram                          | 51200 |
| floating_ips                 | 12    |
| fixed_ips                    | -1    |
| metadata_items               | 128   |
| injected_files               | 5     |
| injected_file_content_bytes | 10240 |
| injected_file_path_bytes     | 255   |
| key_pairs                    | 100   |
| security_groups              | 10    |
| security_group_rules         | 20    |
+-----------------------------+-------+
```

Note

To view a list of options for the **quota-update** command, run:

```
$ nova help quota-update
```

10.2. Manage Block Storage service quotas

As an administrative user, you can update the OpenStack Block Storage service quotas for a project. You can also update the quota defaults for a new project.

Table 3.5. Block Storage quotas

Property name	Defines the number of
gigabytes	Volume gigabytes allowed for each tenant.
snapshots	Volume snapshots allowed for each tenant.
volumes	Volumes allowed for each tenant.

10.2.1. View and update Block Storage service quotas

As an administrative user, you can view and update Block Storage service quotas.

1. List the default quotas for all projects, as follows:

```
$ cinder quota-defaults
```

```
+-----------+-------+
|  Property | Value |
+-----------+-------+
| gigabytes |  1000 |
```

```
| snapshots |   10  |
|  volumes  |   10  |
+-----------+-------+
```

2. To update a default value for a new project, update the property in the **/etc/cinder/cinder.conf** file.

3. View Block Storage service quotas for a project, as follows:

```
$ cinder quota-show TENANT_NAME
```

For example:

```
$ cinder quota-show tenant01
```

```
+-----------+-------+
|  Property | Value |
+-----------+-------+
| gigabytes |  1000 |
| snapshots |   10  |
|  volumes  |   10  |
+-----------+-------+
```

4. To update Block Storage service quotas, place the tenant ID in a usable variable, as follows:

```
$ tenant=$(keystone tenant-list | awk '/tenantName/ {print
$2}')
```

5. Update a particular quota value, as follows:

```
$ cinder quota-update --quotaName NewValue tenantID
```

For example:

```
$ cinder quota-update --volumes 15 $tenant
```

```
$ cinder quota-show tenant01
```

```
+-----------+-------+
|  Property | Value |
+-----------+-------+
| gigabytes |  1000 |
| snapshots |   10  |
|  volumes  |   15  |
+-----------+-------+
```

6. To clear per-tenant quota limits, use the **quota-delete** command:

```
$ cinder quota-delete tenantID
```

10.3. Manage Networking service quotas

A quota limits the number of available resources. A default quota might be enforced for all tenants. When you try to create more resources than the quota allows, an error occurs:

```
$ neutron net-create test_net
```

```
Quota exceeded for resources: ['network']
```

Per-tenant quota configuration is also supported by the quota extension API. See Per-tenant quota configuration for details.

10.3.1. Basic quota configuration

In the Networking default quota mechanism, all tenants have the same quota values, such as the number of resources that a tenant can create.

The quota value is defined in the OpenStack Networking **neutron.conf** configuration file. To disable quotas for a specific resource, such as network, subnet, or port, remove a corresponding item from **quota_items**. This example shows the default quota values:

```
[quotas]
# resource name(s) that are supported in quota features
quota_items = network,subnet,port

# number of networks allowed per tenant, and minus means unlimited
quota_network = 10

# number of subnets allowed per tenant, and minus means unlimited
quota_subnet = 10

# number of ports allowed per tenant, and minus means unlimited
quota_port = 50

# default driver to use for quota checks
quota_driver = neutron.quota.ConfDriver
```

OpenStack Networking also supports quotas for L3 resources: router and floating IP. Add these lines to the **quotas** section in the **neutron.conf** file.

```
[quotas]
# number of routers allowed per tenant, and minus means unlimited
quota_router = 10

# number of floating IPs allowed per tenant, and minus means unlimited
quota_floatingip = 50
```

Note

The **quota_items** option does not affect these quotas.

OpenStack Networking also supports quotas for security group resources: number of security groups and the number of rules for each security group. Add these lines to the **quotas** section in the **neutron.conf** file:

```
[quotas]
# number of security groups per tenant, and minus means unlimited
quota_security_group = 10

# number of security rules allowed per tenant, and minus means
unlimited
quota_security_group_rule = 100
```

Note

The **quota_items** option does not affect these quotas.

10.3.2. Configure per-tenant quotas

OpenStack Networking also supports per-tenant quota limit by quota extension API.

Use these commands to manage per-tenant quotas:

- **neutron quota-delete**. Deletes defined quotas for a specified tenant.
- **neutron quota-list**. Lists defined quotas for all tenants.
- **neutron quota-show**. Shows quotas for a specified tenant.
- **neutron quota-update**. Updates quotas for a specified tenant.

Only users with the **admin** role can change a quota value. By default, the default set of quotas are enforced for all tenants, so no **quota-create** command exists.

1. **Configure Networking to show per-tenant quotas**

   Set the **quota_driver** option in the **neutron.conf** file:

   ```
   quota_driver = neutron.db.quota_db.DbQuotaDriver
   ```

   When you set this option, the output for Networking commands shows **quotas**.

2. **List Networking extensions**

   To list the Networking extensions, run this command:

   ```
   $ neutron ext-list -c alias -c name
   ```

   The command shows the **quotas** extension, which provides per-tenant quota management support:

   ```
   +-----------------+--------------------------+
   | alias           | name                     |
   +-----------------+--------------------------+
   | agent_scheduler | Agent Schedulers         |
   | security-group  | security-group           |
   | binding         | Port Binding             |
   | quotas          | Quota management support |
   | agent           | agent                    |
   ```

```
| provider         | Provider Network         |
| router           | Neutron L3 Router        |
| lbaas            | LoadBalancing service    |
| extraroute       | Neutron Extra Route      |
+------------------+--------------------------+
```

3. **Show information for the quotas extension**

   To show information for the **quotas** extension, run this command:

```
$ neutron ext-show quotas
```

```
+--------------+----------------------------------------------
--------------+
| Field        | Value
|
+--------------+----------------------------------------------
--------------+
| alias        | quotas
|
| description  | Expose functions for quotas management per
tenant           |
| links        |
|
| name         | Quota management support
|
| namespace    | http://docs.openstack.org/network/ext/quotas-
sets/api/v2.0 |
| updated      | 2012-07-29T10:00:00-00:00
|
+--------------+----------------------------------------------
--------------+
```

> **Note**
>
> Only some plug-ins support per-tenant quotas. Specifically, Open vSwitch, Linux Bridge, and VMware NSX support them, but new versions of other plug-ins might bring additional functionality. See the documentation for each plug-in.

4. **List tenants who have per-tenant quota support**

   The **quota-list** command lists tenants for which the per-tenant quota is enabled. The command does not list tenants with default quota support. You must be an administrative user to run this command:

```
$ neutron quota-list
```

```
+------------+---------+------+--------+--------+------------
----------------------+
| floatingip | network | port | router | subnet | tenant_id
|
+------------+---------+------+--------+--------+------------
----------------------+
|         20 |       5 |   20 |     10 |      5 |
```

```
6f88036c45344d9999a1f971e4882723 |
|          25 |       10 |    30 |      10 |      10 |
bff5c9455ee24231b5bc713c1b96d422 |
+------------+---------+------+--------+--------+------------
----------------------+
```

5. **Show per-tenant quota values**

   The **quota-show** reports the current set of quota limits for the specified tenant. Non-administrative users can run this command without the ***--tenant_id*** parameter. If per-tenant quota limits are not enabled for the tenant, the command shows the default set of quotas:

```
$ neutron quota-show --tenant_id
6f88036c45344d9999a1f971e4882723
```

```
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 20    |
| network    | 5     |
| port       | 20    |
| router     | 10    |
| subnet     | 5     |
+------------+-------+
```

   The following command shows the command output for a non-administrative user:

```
$ neutron quota-show
```

```
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 20    |
| network    | 5     |
| port       | 20    |
| router     | 10    |
| subnet     | 5     |
+------------+-------+
```

6. **Update quota values for a specified tenant**

   Use the **quota-update** command to update a quota for a specified tenant:

```
$ neutron quota-update --tenant_id
6f88036c45344d9999a1f971e4882723 --network 5
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 50    |
| network    | 5     |
| port       | 50    |
| router     | 10    |
| subnet     | 10    |
+------------+-------+
```

–

You can update quotas for multiple resources through one command:

```
$ neutron quota-update --tenant_id
6f88036c45344d9999a1f971e4882723 --subnet 5 --port 20
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 50    |
| network    | 5     |
| port       | 20    |
| router     | 10    |
| subnet     | 5     |
+------------+-------+
```

To update the limits for an L3 resource such as, router or floating IP, you must define new values for the quotas after the **--** directive.

This example updates the limit of the number of floating IPs for the specified tenant:

```
$ neutron quota-update --tenant_id
6f88036c45344d9999a1f971e4882723 -- --floatingip 20
```

```
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 20    |
| network    | 5     |
| port       | 20    |
| router     | 10    |
| subnet     | 5     |
+------------+-------+
```

You can update the limits of multiple resources by including L2 resources and L3 resource through one command.

```
$ neutron quota-update --tenant_id
6f88036c45344d9999a1f971e4882723 --network 3 --subnet 3 --port
3 -- --floatingip 3 --router 3
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 3     |
| network    | 3     |
| port       | 3     |
| router     | 3     |
| subnet     | 3     |
+------------+-------+
```

7. **Delete per-tenant quota values**

   To clear per-tenant quota limits, use the **quota-delete** command:

```
$ neutron quota-delete --tenant_id
6f88036c45344d9999a1f971e4882723
```

```
Deleted quota: 6f88036c45344d9999a1f971e4882723
```

After you run this command, you can see that quota values for the tenant are reset to the default values:

```
$ neutron quota-show --tenant_id
6f88036c45344d9999a1f971e4882723
```

```
+------------+-------+
| Field      | Value |
+------------+-------+
| floatingip | 50    |
| network    | 10    |
| port       | 50    |
| router     | 10    |
| subnet     | 10    |
+------------+-------+
```

11. Analyze log files

Use the swift command-line client to analyze log files.

The swift client is simple to use, scalable, and flexible.

Use the swift client **-o** or **-output** option to get short answers to questions about logs.

You can use the **-o –output** option with a single object download to redirect the command output to a specific file or to STDOUT (**-**). The ability to redirect the output to STDOUT enables you to pipe (**|**) data without saving it to disk first.

11.1. Upload and analyze log files

1. This example assumes that **logtest** directory contains the following log files:

Example 3.1. Example files

```
2010-11-16-21_access.log
2010-11-16-22_access.log
2010-11-15-21_access.log
2010-11-15-22_access.log
```

Each file uses the following line format:

Example 3.2. Log line format

```
Nov 15 21:53:52 lucid64 proxy-server - 127.0.0.1
15/Nov/2010/22/53/52 DELETE
/v1/AUTH_cd4f57824deb4248a533f2c28bf156d3/2eefc05599d44df38a
```

```
7f18b0b42ffedd HTTP/1.0 204 - \
  - test%3Atester%2CAUTH_tkcdab3c6296e249d7b7e2454ee57266ff
- - - txaba5984c-aac7-460e-b04b-afc43f0c6571 - 0.0432
```

2. Change into the **logtest** directory:

```
$ cd logtest
```

3. Upload the log files into the **logtest** container:

```
$ swift -A http://swift-auth.com:11000/v1.0 -U test:tester -K
testing upload logtest *.log
```

```
2010-11-16-21_access.log
2010-11-16-22_access.log
2010-11-15-21_access.log
2010-11-15-22_access.log
```

4. Get statistics for the account:

```
$ swift -A http://swift-auth.com:11000/v1.0 -U test:tester -K
testing -q stat
```

```
Account: AUTH_cd4f57824deb4248a533f2c28bf156d3
Containers: 1
Objects: 4
Bytes: 5888268
```

5. Get statistics for the logtest container:

```
$ swift -A http://swift-auth.com:11000/v1.0 -U test:tester -K
testing stat logtest
```

```
Account: AUTH_cd4f57824deb4248a533f2c28bf156d3
Container: logtest
Objects: 4
Bytes: 5864468
Read ACL:
Write ACL:
```

6. List all objects in the logtest container:

```
$ swift -A http:///swift-auth.com:11000/v1.0 -U test:tester -K
testing list logtest
```

```
2010-11-15-21_access.log
2010-11-15-22_access.log
2010-11-16-21_access.log
2010-11-16-22_access.log
```

11.2. Download and analyze an object

This example uses the **-o –output** option and a hyphen (**-**) to get information about an object.

Use the swift **download** command to download the object. On this command, stream the output to **awk** to break down requests by return code and the date **2200 on November 16th, 2010**.

Using the log line format, find the request type in column 9 and the return code in column 12.

After **awk** processes the output, it pipes it to **sort** and **uniq -c** to sum up the number of occurrences for each request type and return code combination.

1. Download an object:

```
$ swift -A http://swift-auth.com:11000/v1.0 -U test:tester -K
testing \
 download -o - logtest 2010-11-16-22_access.log | awk ‘{ print
$9”-“$12}’ | sort | uniq -c
```

```
805 DELETE-204
12 DELETE-404
2 DELETE-409
723 GET-200
142 GET-204
74 GET-206
80 GET-304
34 GET-401
5 GET-403
18 GET-404
166 GET-412
2 GET-416
50 HEAD-200
17 HEAD-204
20 HEAD-401
8 HEAD-404
30 POST-202
25 POST-204
22 POST-400
6 POST-404
842 PUT-201
2 PUT-202
32 PUT-400
4 PUT-403
4 PUT-404
2 PUT-411
6 PUT-412
6 PUT-413
2 PUT-422
8 PUT-499
```

2. Discover how many PUT requests are in each log file.

   Use a bash for loop with awk and swift with the **-o –output** option and a hyphen (**-**) to discover how many PUT requests are in each log file.

Run the swift **list** command to list objects in the logtest container. Then, for each item in the list, run the swift **download -o -** command. Pipe the output into grep to filter the PUT requests. Finally, pipe into **wc -l** to count the lines.

```
$ for f in `swift -A http://swift-auth.com:11000/v1.0 -U
test:tester -K testing list logtest` ; \
 do echo -ne “PUTS - ” ; swift -A http://swift-
auth.com:11000/v1.0 -U test:tester -K testing download -o -
logtest $f | grep PUT | wc -l ; \
 done
```

```
2010-11-15-21_access.log - PUTS - 402
2010-11-15-22_access.log - PUTS - 1091
2010-11-16-21_access.log - PUTS - 892
2010-11-16-22_access.log - PUTS - 910
```

3. List the object names that begin with a specified string.

   Run the swift **list -p 2010-11-15** command to list objects in the logtest container that begin with the **2010-11-15** string.

   For each item in the list, run the swift **download -o -** command.

   Pipe the output to **grep** and **wc**. Use the **echo** command to display the object name.

```
$ for f in `swift -A http://swift-auth.com:11000/v1.0 -U
test:tester -K testing list -p 2010-11-15 logtest` ; \
 do echo -ne “$f - PUTS - ” ; swift -A
http://127.0.0.1:11000/v1.0 -U test:tester \
 -K testing download -o - logtest $f | grep PUT | wc -l ; \
 done
```

```
2010-11-15-21_access.log - PUTS - 402
2010-11-15-22_access.log - PUTS - 910
```

Revision History

Revision 5.0.0-19 **Fri Oct 31 2014** **Deepti Navale**

BZ#1145399 - Updates to reflect changes to UI.

Revision 5.0.0-18 **Tue Oct 28 2014** **Bruce Reeler**

BZ#1152399 - Additional corrections to align new Dashboard tabs with text.

Revision 5.0.0-17 **Wed Oct 22 2014** **Bruce Reeler**

BZ#1152399 - Corrections of errors found during translation. Removed entries in this Revision History prior to ver 5.

BZ#1152400 - Updated "View cloud resources" and "Manage projects and users " instructions to match GUI.

Revision 5.0.0-16 **Wed Oct 22 2014** **Summer Long**

BZ#1147796 - Added --nic info to 'nova boot' examples.

Revision 5.0.0-15 **Thu Aug 28 2014** **Bruce Reeler**

BZ#1110079 - Removed Additional Documentation table.

Revision 5.0.0-10 **Mon Aug 4 2014** **Bruce Reeler**

Corrected hyperlinks in Legal Notice.

Revision 5.0.0-9 **Mon Jul 7 2014** **Bruce Reeler**

Final version for Red Hat Enterprise Linux OpenStack Platform 5.0

Revision 5.0.0-7 **Thu Jun 26 2014** **Summer Long**

Updated subtitle, brand.

Revision 5.0.0-6 **Wed Jun 18 2014** **Bruce Reeler**

BZ#1110079 - Updated files as required for DocBook5.

Revision 5.0.0-3 **Wed Jun 18 2014** **Deepti Navale**

BZ#1086542 - Included a note in the host aggregates section about managing availability zones.

Revision 5.0.0-2 **Tue May 27 2014** **Bruce Reeler**

BZ#975542 - Added information on creating a new tenant and related objects.

BZ#1081037 - Corrected typo in Legal Notice.

BZ#1056862 - Functionality in Dashboard updated: Create and Manage Images, View and manage quotas, Manage flavors, View cloud resources.

BZ#1063148 - New Dashboard procedure added: Create and manage roles.

BZ#1081699 - Added section: Create and manage host aggregates.

BZ#1086074 - Documented how to view daily usage report.

Revision 5.0.0-1 **Fri May 23 2014** **Bruce Reeler**

Initial version for Red Hat Enterprise Linux OpenStack Platform 5.0