# Red Hat Enterprise Linux OpenStack Platform 6
# Administration Guide

Managing a Red Hat Enterprise Linux OpenStack Platform environment

OpenStack Documentation TeamRed Hat

# Red Hat Enterprise Linux OpenStack Platform 6 Administration Guide

## Managing a Red Hat Enterprise Linux OpenStack Platform environment

OpenStack Documentation Team
Red Hat Customer Content Services
rhos-docs@redhat.com

## Legal Notice

## Abstract

This Administration Guide provides procedures for the management of a Red Hat Enterprise Linux OpenStack Platform environment. Procedures to manage both user projects and the cloud configuration are provided.

# CHAPTER 1. INTRODUCTION

Red Hat Enterprise Linux OpenStack Platform (RHEL OpenStack Platform) provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud on top of Red Hat Enterprise Linux. It offers a massively scalable, fault-tolerant platform for the development of cloud-enabled workloads.

This guide provides cloud-management procedures for the following OpenStack services: Block Storage, Compute, Dashboard, Identity, Image, Object Storage, OpenStack Networking, Orchestration, and Telemetry.

Procedures for both administrators and project users (end users) are provided; administrator-only procedures are marked as such.

You can manage the cloud using either the OpenStack dashboard or the command-line clients. Most procedures can be carried out using either method; some of the more advanced procedures can only be executed on the command line. This guide provides procedures for the dashboard where possible.

**Note**

For the complete suite of documentation for RHEL OpenStack Platform, see https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/

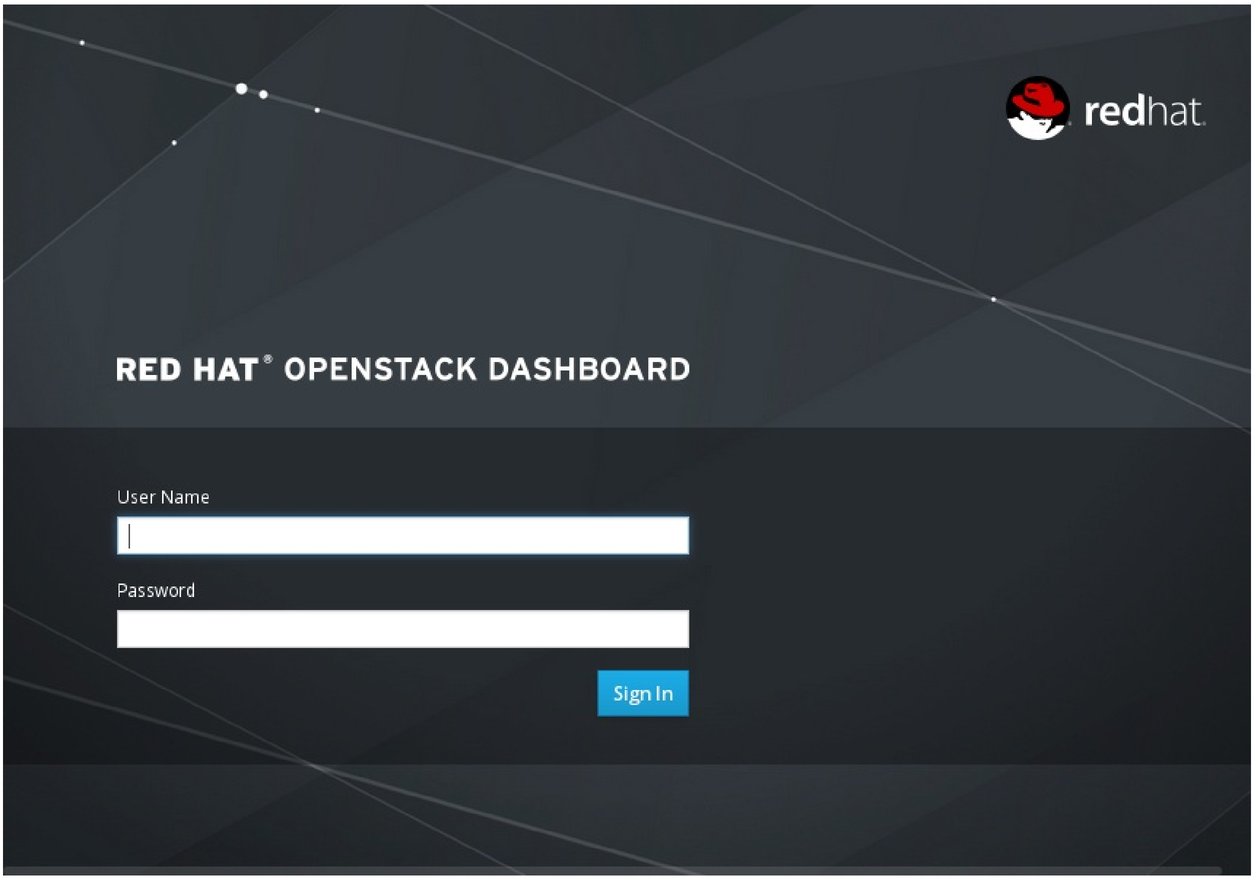
## 1.1. OPENSTACK DASHBOARD

The OpenStack dashboard is a web-based graphical user interface for managing OpenStack services.

To access the browser dashboard, the dashboard service must be installed, and you must know the dashboard host name (or IP) and login password. The dashboard URL will be:

```
http://HOSTNAME/dashboard/
```

**Figure 1.1. Log In Screen**



RED HAT® OPENSTACK DASHBOARD

User Name

Password

## 1.2. COMMAND-LINE CLIENTS

Each RHEL OpenStack Platform component typically has its own management client. For example, the Compute service has the **nova** client. For a complete listing of client commands and parameters, see the "Command-Line Interface Reference" in https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/

To use a command-line client, the client must be installed and you must first load the environment variables used for authenticating with the Identity service. You can do this by creating an RC (run control) environment file, and placing it in a secure location to run as needed.

Run the file using:

```
$ source RC_FileName
```

**Example 1.1.**

```
$ source ~/keystonerc_admin
```

**Note**

By default, the Packstack utility creates the **admin** and **demo** users, and their **keystone_admin** and **keystone_demo** RC files.

### 1.2.1. Automatically Create an RC File

Using the dashboard, you can automatically generate and download an RC file for the current project user, which enables the use of the OpenStack command-line clients (see Section 1.2, "Command-line Clients"). The file's environment variables map to the project and the current project's user.

1. In the dashboard, select the **Project** tab, and click **Compute > Access & Security**.

2. Select the **API Access** tab, which lists all services that are visible to the project's logged-in user.

3. Click **Download OpenStack RC file** to generate the file. The file name maps to the current user. For example, if you are an 'admin' user, an **admin-openrc.sh** file is generated and downloaded through the browser.

### 1.2.2. Manually Create an RC File

If you create an RC file manually, you must set the following environment variables:

- OS_USERNAME=*userName*
- OS_TENANT_NAME=*tenantName*
- OS_PASSWORD=*userPassword*
- OS_AUTH_URL=http://*IP*:35357/v2.0/
- PS1='[\u@\h \W(keystone_*userName*)]\$ '

**Example 1.2.**

The following example file sets the necessary variables for the **admin** user:

```
export OS_USERNAME=admin
export OS_TENANT_NAME=admin
export OS_PASSWORD=secretPass
export OS_AUTH_URL=http://192.0.2.24:35357/v2.0/
export PS1='[\u@\h \W(keystone_admin)]\$ '
```

# CHAPTER 2. PROJECTS AND USERS

As a cloud administrator, you can 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 can be members of one or more projects.

During cloud setup, 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 a cloud administrator, you can create additional projects and users as needed. Additionally, you can add, update, and delete projects and users, assign users to one or more projects, and change or remove these assignments. To enable or temporarily disable a project or user, update that project or user.

After 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. MANAGE PROJECTS

### 2.1.1. Create a Project

1. As an admin user in the dashboard, select **`Identity > Projects`**.
2. Click **`Create Project`**.
3. On the **`Project Information`** tab, enter a name and description for the project (the **`Enabled`** check box is selected by default).
4. On the **`Project Members`** tab, add members to the project from the **`All Users`** list.
5. On the **`Quotas`** tab, specify resource limits for the project.
6. Click **`Create Project`**.

### 2.1.2. Update a Project

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

1. As an admin user in the dashboard, select **`Identity > Projects`**.
2. In the project's **`Actions`** column, click the arrow, and click **`Edit Project`**.
3. In the **`Edit Project`** window, you can update a project to change its name or description, and enable or temporarily disable the project.

4. On the **Project Members** tab, add members to the project, or remove them as needed.

5. Click **Save**.

**Note**

The **Enabled** check box is selected by default. To temporarily disable the project, clear the **Enabled** check box. To enable a disabled project, select the **Enabled** check box.

### 2.1.3. Delete a Project

1. As an admin user in the dashboard, select **Identity > Projects**.

2. Select the project to delete.

3. Click **Delete Projects**.

4. Click **Delete Projects** again.

**Note**

You cannot undo the delete action.

### 2.1.4. Update Project Quotas

Quotas are maximum limits that can be set per project, so that the project's resources are not exhausted.

1. As an admin user in the dashboard, select **Identity > Projects**.

2. In the project's **Actions** column, click the arrow, and click **Modify Quotas**.

3. In the **Quota** tab, modify project quotas as needed.

4. Click **Save**.

### 2.1.5. Manage Project Security

Security groups are sets of IP filter rules that can be assigned to project instances, and which define networking access to the instance. Security groups are project specific; project members can edit the default rules for their security 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 values, this security group denies all incoming traffic and allows only outgoing traffic to your instance.

### 2.1.5.1. Create a Security Group

1. In the dashboard, select **`Project > Compute > Access & Security`**.
2. On the **`Security Groups`** tab, click **`Create Security Group`**.
3. Provide a name and description for the group, and click **`Create Security Group`**.

### 2.1.5.2. Add a Security Group Rule

By default, rules for a new group only provide outgoing access. You must add new rules to provide additional access.

1. In the dashboard, select **`Project > Compute > Access & Security`**.
2. On the **`Security Groups`** tab, click **`Manage Rules`** for the security group.
3. Click **`Add Rule`** to add a new rule.
4. Specify the rule values, and click **`Add`**.

**Table 2.1. Required Rule Fields**

| Field | Description |
|---|---|
| Rule | Rule type. If you specify a rule template (for example, 'SSH'), its fields are automatically filled in:<br>- TCP: Typically used to exchange data between systems, and for end-user communication.<br>- UDP: Typically used to exchange data between systems, particularly at the application level.<br>- ICMP: Typically used by network devices, such as routers, to send error or monitoring messages. |
| Direction | Ingress (inbound), or Egress (outbound) |

| Field | Description |
|---|---|
| Open Port | For TCP or UDP rules, the **Port** or **Port Range** to open for the rule (single port or range of ports):<br>- For a range of ports, enter port values in the **From Port** and **To Port** fields.<br>- For a single port, enter the port value in the **Port** field. |
| Type | The type for ICMP rules; must be in the range '-1:255'. |
| Code | The code for ICMP rules; must be in the range '-1:255'. |
| Remote | The traffic source for this rule:<br>- CIDR (Classless Inter-Domain Routing): IP address block, which limits access to IPs within the block. Enter the CIDR in the Source field.<br>- Security Group: Source group that enables any instance in the group to access any other group instance. |

### 2.1.5.3. Delete a Security Group Rule

1. In the dashboard, select **Project > Compute > Access & Security**.
2. On the **Security Groups** tab, click **Manage Rules** for the security group.
3. Select the security group rule, and click **Delete Rule**.
4. Click **Delete Rule**.

**Note**

You cannot undo the delete action.

### 2.1.5.4. Delete a Security Group

1. In the dashboard, select **Project > Compute > Access & Security**.
2. On the **Security Groups** tab, select the group, and click **Delete Security Groups**.
3. Click **Delete Security Groups**.

**Note**

You cannot undo the delete action.

## 2.2. MANAGE USERS

### 2.2.1. Create a User

1. As an admin user in the dashboard, select **`Identity > Users`**.

2. Click **`Create User`**.

3. Enter a user name, email, and preliminary password for the user.

4. Select a project from the **`Primary Project`** list.

5. Select a role for the user from the **`Role`** list (the default role is **`_member_`**).

6. Click **`Create User`**.

### 2.2.2. Enable or Disable a User

You can disable or enable only one user at a time.

1. As an admin user in the dashboard, select **`Identity > Users`**.

2. In the **`Actions`** column, click the arrow, and select **`Enable User`** or **`Disable User`**. In the **`Enabled`** column, the value then updates to either **`True`** or **`False`**.

### 2.2.3. Delete a User

1. As an admin user in the dashboard, select **`Identity > Users`**.

2. Select the users that to delete.

3. Click **`Delete Users`**.

4. Click **`Delete Users`**.

**Note**

You cannot undo the delete action.

### 2.2.4. Manage Roles

#### 2.2.4.1. View Roles

To list the available roles:

```
$ keystone role-list
+----------------------------------+---------------+
|                id                |      name     |
+----------------------------------+---------------+
| 71ccc37d41c8491c975ae72676db687f |     Member    |
| 149f50a1fe684bfa88dae76a48d26ef7 | ResellerAdmin |
| 9fe2ff9ee4384b1894a90878d3e92bab |    _member_   |
| 6ecf391421604da985db2f141e46a7c8 |     admin     |
+----------------------------------+---------------+
```

To get details for a specified role:

```
$ keystone role-get ROLE
```

**Example 2.1.**

```
$ keystone role-get admin
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | 6ecf391421604da985db2f141e46a7c8 |
|   name   |              admin               |
+----------+----------------------------------+
```

#### 2.2.4.2. Create and Assign a Role

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

**Note**

Either the name or ID can be used to specify users, roles, or projects.

1. Create the **`new-role`** role:

   ```
   $ keystone role-create --name ROLE_NAME
   ```

   **Example 2.2.**

```
$ keystone role-create --name new-role
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | 61013e7aa4ba4e00a0a1ab4b14bc6b2a |
|   name   |             new-role             |
+----------+----------------------------------+
```

2. 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 names or IDs.

   a. List users:

   ```
   $ keystone user-list
   ```

   b. List roles:

   ```
   $ keystone role-list
   ```

   c. List projects:

   ```
   $ keystone tenant-list
   ```

3. Assign a role to a user-project pair.

```
$ keystone user-role-add --user USER_NAME --role ROLE_NAME --
tenant TENANT_NAME
```

**Example 2.3.**

In this example, you assign the **new-role** role to the **demo**-**demo** pair:

```
$ keystone user-role-add --user demo --role new-role --
tenant demo
```

4. Verify the role assignment for the user **demo**:

```
$ keystone user-role-list --user USER_NAME --tenant
TENANT_NAME
```

**Example 2.4.**

```
$ keystone user-role-list --user demo --tenant demo
```

#### 2.2.4.3. Delete a Role

1. Remove a role from a user-project pair:

   ```
   $ keystone user-role-remove --user USER_NAME --role ROLE --
   tenant TENANT_NAME
   ```

2. Verify the role removal:

   ```
   $ keystone user-role-list --user USER_NAME --tenant
   TENANT_NAME
   ```

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

### 2.2.5. View Compute Quotas for a Project User

To list the currently set quota values for a project user (tenant user), run:

```
$ nova quota-show --user USER --tenant TENANT
```

**Example 2.5.**

```
$ nova quota-show --user demoUser --tenant demo
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 5     |
| 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    |
| server_groups               | 10    |
| server_group_members        | 10    |
+-----------------------------+-------+
```

### 2.2.6. Update Compute Quotas for a Project User

**Procedure 2.1. Update Compute Quotas for User**

To update a particular quota value, run:

```
$ nova quota-update --user USER --QUOTA_NAME QUOTA_VALUE TENANT
```

**Example 2.6.**

```
$ nova quota-update --user demoUser --floating-ips 10 demo
$ nova quota-show --user demoUser --tenant demo
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 10    |
| ...                         |       |
+-----------------------------+-------+
```

**Note**

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

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

### 2.2.7. Configure Role Access Control

A user can have different roles in different tenants. A user can also have multiple roles in the same tenant.

The **/etc/*[SERVICE_CODENAME]*/policy.json** file controls the tasks that users can perform for a given service. For example:

- **/etc/nova/policy.json** specifies the access policy for the Compute service.
- **/etc/glance/policy.json** specifies the access policy for the Image Service
- **/etc/keystone/policy.json** specifies the access policy for the Identity Service.

The default **policy.json** files for the Compute, Identity, and Image services recognize only the admin role; all operations that do not require the admin role are accessible by any user that has any role in a tenant.

For example, if you wish to restrict users from performing operations in the Compute service, you must create a role in the Identity service, give users that role, and then modify **/etc/nova/policy.json** so that the role is required for Compute operations.

**Example 2.7.**

The following line in **`/etc/nova/policy.json`** specifies that there are no restrictions on which users can create volumes; if the user has any role in a tenant, they can create volumes in that tenant.

```
"volume:create": [],
```

**Example 2.8.**

To restrict creation of volumes to users who had the compute-user role in a particular tenant, you would add "role:compute-user" to the Compute policy:

```
"volume:create": ["role:compute-user"],
```

**Example 2.9.**

To restrict all Compute service requests to require this role, values in the file might look like the following (not a complete example):

```
{"admin_or_owner": [["role:admin"], ["project_id:%
(project_id)s"]],
"default": [["rule:admin_or_owner"]],
"compute:create": ["role:compute-user"],
"compute:create:attach_network": ["role:compute-user"],
"compute:create:attach_volume": ["role:compute-user"],
"compute:get_all": ["role:compute-user"],
"compute:unlock_override": ["rule:admin_api"],
"admin_api": [["role:admin"]],
"compute_extension:accounts": [["rule:admin_api"]],
"compute_extension:admin_actions": [["rule:admin_api"]],
"compute_extension:admin_actions:pause":
[["rule:admin_or_owner"]],
"compute_extension:admin_actions:unpause":
[["rule:admin_or_owner"]],
"compute_extension:admin_actions:suspend":
[["rule:admin_or_owner"]],
"compute_extension:admin_actions:resume":
[["rule:admin_or_owner"]],
"compute_extension:admin_actions:lock": [["rule:admin_or_owner"]],
"compute_extension:admin_actions:unlock":
[["rule:admin_or_owner"]],
"compute_extension:admin_actions:resetNetwork":
[["rule:admin_api"]],
"compute_extension:admin_actions:injectNetworkInfo":
[["rule:admin_api"]],
"compute_extension:admin_actions:createBackup":
[["rule:admin_or_owner"]],
"compute_extension:admin_actions:migrateLive":
```

```
[["rule:admin_api"]],
"compute_extension:admin_actions:migrate": [["rule:admin_api"]],
"compute_extension:aggregates": [["rule:admin_api"]],
"compute_extension:certificates": ["role:compute-user"],
"compute_extension:cloudpipe": [["rule:admin_api"]],
"compute_extension:console_output": ["role:compute-user"],
"compute_extension:consoles": ["role:compute-user"],
"compute_extension:createserverext": ["role:compute-user"],
"compute_extension:deferred_delete": ["role:compute-user"],
"compute_extension:disk_config": ["role:compute-user"],
"compute_extension:evacuate": [["rule:admin_api"]],
"compute_extension:extended_server_attributes":
[["rule:admin_api"]],
 ...
```

# CHAPTER 3. VIRTUAL MACHINE INSTANCES

The RHEL OpenStack Platform allows you to easily manage virtual machine instances in the cloud. OpenStack Compute is the central component that creates, schedules, and manages instances, and exposes this functionality to other OpenStack components.

**Note**

The term 'instance' is used by OpenStack to mean a virtual machine instance.

## 3.1. MANAGE INSTANCES

### 3.1.1. Create an Instance

**Prerequisites:** Ensure that a network, key pair, and a boot source are available:

1. In the dashboard, select **`Project`**.

2. Select **`Network > Networks`**, and ensure there is a private network to which you can attach the new instance (to create a network, see Section 5.1.1, “Add a Network”).

3. Select **`Compute > Access & Security > Key Pairs`**, and ensure there is a key pair (to create a key pair, see Section 3.2.1, “Manage Key Pairs”).

4. Ensure that you have either an image or a volume that can be used as a boot source:

   - To view boot-source images, select the **`Images`** tab (to create an image, see Section 4.1.1, “Create an Image”).

   - To view boot-source volumes, select the **`Volumes`** tab (to create a volume, see Section 4.2.1.1, “Create a Volume”).

**Procedure 3.1. Create an Instance**

1. In the dashboard, select **`Project > Compute > Instances`**.

2. Click **`Launch Instance`**.

3. Fill out instance fields (those marked with '*' are required), and click **`Launch`** when finished.

| Tab | Field | Notes |
|---|---|---|

| Tab | Field | Notes |
|---|---|---|
| Details | Availability Zone | Zones are logical groupings of cloud resources in which your instance can be placed. If you are unsure, use the default zone (for more information, see section Section 3.4, "Manage Host Aggregates"). |
| | Instance Name | The name must be unique within the project. |
| | Flavor | The flavor determines what resources the instance is given (for example, memory). For default flavor allocations and information on creating new flavors, see Section 3.3, "Manage Flavors". |
| | Instance Boot Source | Depending on the item selected, new fields are displayed allowing you to select the source:<br>- Image sources must be compatible with OpenStack (see Section 4.1, "Manage Images").<br>- If a volume or volume source is selected, the source must be formatted using an image (see Section 4.2, "Manage Volumes"). |
| Access and Security | Key Pair | The specified key pair is injected into the instance and is used to remotely access the instance using SSH (if neither a direct login information or a static key pair is provided). Usually one key pair per project is created. |
| | Security Groups | Security groups contain firewall rules which filter the type and direction of the instance's network traffic (for more information on configuring groups, see Section 2.1.5, "Manage Project Security"). |
| Networking | Selected Networks | You must select at least one network. Instances are typically assigned to a private network, and then later given a floating IP address to enable external access. |

| Tab | Field | Notes |
|---|---|---|
| Post-Creation | Customization Script Source | You can provide either a set of commands or a script file, which will run after the instance is booted (for example, to set the instance hostname or a user password). If 'Direct Input' is selected, write your commands in the **`Script Data`** field; otherwise, specify your script file. **Note:** Any script that starts with '#cloud-config' is interpreted as using the cloud-config syntax (for information on the syntax, see http://cloudinit.readthedocs.org/en/latest/topics/examples.html). |
| Advanced Options | Disk Partition | By default, the instance is built as a single partition and dynamically resized as needed. However, you can choose to manually configure the partitions yourself. |
| | Configuration Drive | If selected, OpenStack writes metadata to a read-only configuration drive that is attached to the instance when it boots (instead of to Compute's metadata service). After the instance has booted, you can mount this drive to view its contents (enables you to provide files to the instance). |

### 3.1.2. Update an Instance (Actions menu)

You can update an instance by selecting **`Project > Compute > Instance`**, and selecting an action for that instance in the Actions column. Actions allow you to manipulate the instance in a number of ways:

| Action | Description |
|---|---|
| Create Snapshot | Snapshots preserve the disk state of a running instance. You can create a snapshot to migrate the instance, as well as to preserve backup copies. |
| Associate/Disassociate Floating IP | You must associate an instance with a floating IP (external) address before it can communicate with external networks, or be reached by external users. Because there are a limited number of external addresses in your external subnets, it is recommended that you disassociate any unused addresses. |
| Edit Instance | Update the instance's name and associated security groups. |
| Edit Security Groups | Add and remove security groups to or from this instance using the list of available security groups (for more information on configuring groups, see Section 2.1.5, “Manage Project Security”). |

| Action | Description |
|---|---|
| Console | View the instance's console in the browser (allows easy access to the instance). |
| View Log | View the most recent section of the instance's console log. Once opened, you can view the full log by clicking **View Full Log**. |
| Pause/Resume Instance | Immediately pause the instance (you are not asked for confirmation); the state of the instance is stored in memory (RAM). |
| Suspend/Resume Instance | Immediately suspend the instance (you are not asked for confirmation); like hybernation, the state of the instance is kept on disk. |
| Resize Instance | Bring up the Resize Instance window (see Section 3.1.3, “Resize an instance”). |
| Soft Reboot | Gracefully stop and restart the instance. A soft reboot attempts to gracefully shut down all processes before restarting the instance. |
| Hard Reboot | Stop and restart the instance. A hard reboot effectively just shuts down the instance's 'power' and then turns it back on. |
| Shut Off Instance | Gracefully stop the instance. |
| Rebuild Instance | Use new image and disk-partition options to rebuild the image (shut down, re-image, and re-boot the instance). If encountering operating system issues, this option is easier to try than terminating the instance and starting over. |
| Terminate Instance | Permanently destroy the instance (you are asked for confirmation). |

For example, you can create and allocate an external address by using the 'Associate Floating IP' action.

### Procedure 3.2. Update Example - Assign a Floating IP

1. In the dashboard, select **Project > Compute > Instances**.

2. Select the **Associate Floating IP** action for the instance.

   **Note**

   A floating IP address can only be selected from an already created floating IP pool (see Section 5.2.1, “Create Floating IP Pools”).

3. Click '+' and select **Allocate IP > Associate**.

**Note**

If you do not know the name of the instance, just its IP address (and you do not want to flip through the details of all your instances), you can run the following on the command line:

```
$ nova list --ip IPAddress
```

Where *IPAddress* is the IP address you are looking up.

```
$ nova list --ip 192.0.2.0
```

### 3.1.3. Resize an instance

To resize an instance (memory or CPU count), you must select a new flavor for the instance that has the right capacity. If you are increasing the size, remember to first ensure that the host has enough space.

1. If you are resizing an instance in a distributed deployment, you must ensure communication between hosts. Set up each host with SSH key authentication so that Compute can use SSH to move disks to other hosts (for example, compute nodes can share the same SSH key). For more information about setting up SSH key authentication, see Section 3.1.4, “Configure SSH Tunneling between Nodes”.

2. Enable resizing on the original host by setting the following parameter in the **/etc/nova/nova.conf** file:

   ```
   [DEFAULT] allow_resize_to_same_host = True
   ```

3. In the dashboard, select **Project > Compute > Instances**.

4. Click the instance's **Actions** arrow, and select **Resize Instance**.

5. Select a new flavor in the **New Flavor** field.

6. If you want to manually partition the instance when it launches (results in a faster build time):

   a. Select **Advanced Options**.

   b. In the **Disk Partition** field, select 'Manual'.

7. Click **Resize**.

### 3.1.4. Configure SSH Tunneling between Nodes

**Warning**

Red Hat does not recommend any particular libvirt security strategy; SSH-tunneling steps are provided for user reference only. Only users with **root** access can set up SSH tunneling.

To migrate instances between nodes using SSH tunneling or to resize instance in a distributed environment, each node must be set up with SSH key authentication so that the Compute service can use SSH to move disks to other nodes. For example, compute nodes could use the same SSH key to ensure communication.

**Note**

If the Compute service cannot migrate the instance to a different node, it will attempt to migrate the instance back to its original host. To avoid migration failure in this case, ensure that 'allow_migrate_to_same_host=True' is set in the **/etc/nova/nova.conf** file.

To share a key pair between compute nodes:

1. As root on both nodes, make **nova** a login user:

   ```
   # usermod -s /bin/bash nova
   ```

2. On the first compute node, generate a key pair for the **nova** user:

   ```
   # su nova
   # ssh-keygen
   # echo 'StrictHostKeyChecking no' >> /var/lib/nova/.ssh/config
   # cat /var/lib/nova/.ssh/id_rsa.pub >>
   /var/lib/nova/.ssh/authorized_keys
   ```

   The key pair, **id_rsa** and **id_rsa.pub**, is generated in **/var/lib/nova/.ssh**.

3. As root, copy the created key pair to the second compute node:

   ```
   # scp /var/lib/nova/.ssh/id_rsa root@computeNodeAddress:~/
   # scp /var/lib/nova/.ssh/id_rsa.pub root@computeNodeAddress:~/
   ```

4. As root on the second compute node, change the copied key pair's permissions back to 'nova', and then add the key pair into SSH:

   ```
   # chown nova:nova id_rsa
   # chown nova:nova id_rsa.pub
   # su nova
   # mkdir -p /var/lib/nova/.ssh
   ```

```
# cp id_rsa /var/lib/nova/.ssh/
# cat id_rsa.pub >> /var/lib/nova/.ssh/authorized_keys
# echo 'StrictHostKeyChecking no' >> /var/lib/nova/.ssh/config
```

5. Ensure that the **nova** user can now log into each node without using a password:

```
# su nova
# ssh nova@computeNodeAddress
```

6. As root on both nodes, restart both libvirt and the Compute services:

```
# systemctl restart libvirtd.service
# systemctl restart openstack-nova-compute.service
```

### 3.1.5. Connect to an Instance

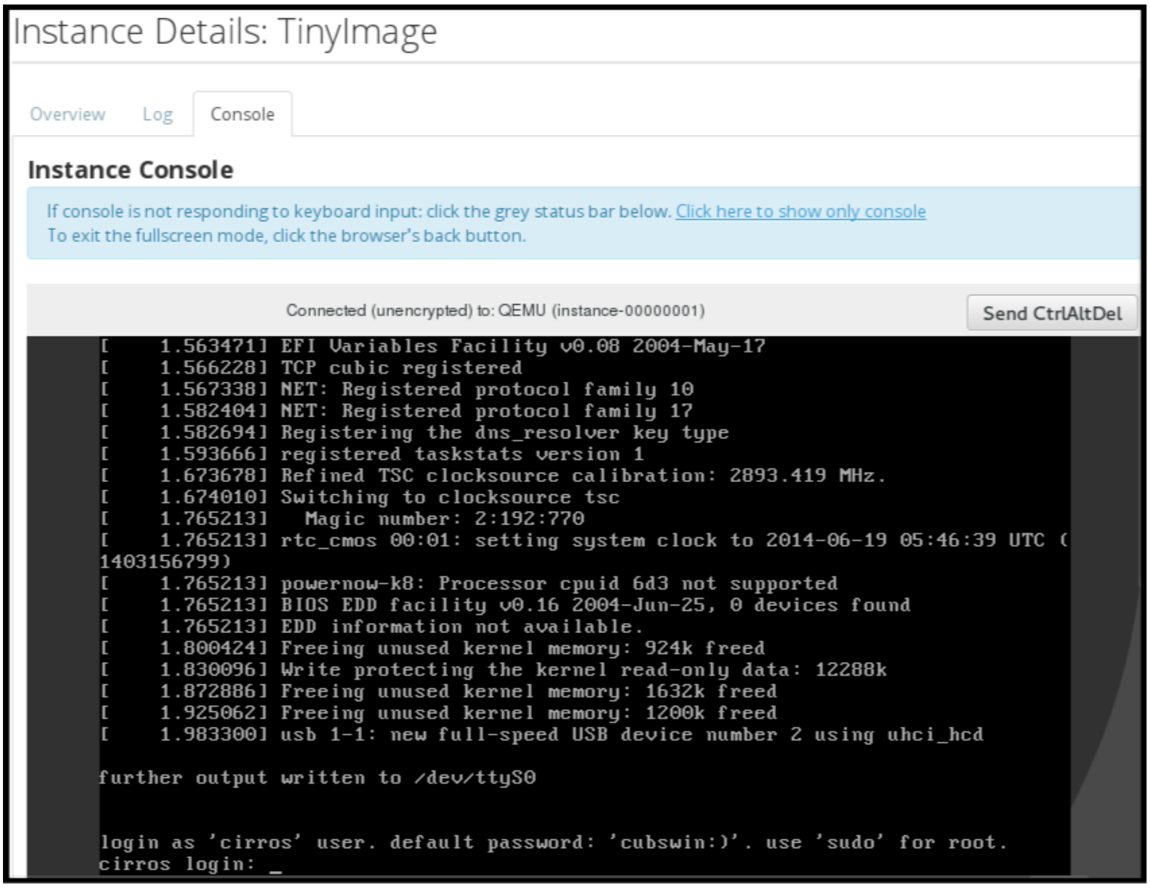
#### 3.1.5.1. Access using the Dashboard Console

The console allows you a way to directly access your instance within the dashboard.

1. In the dashboard, select **Compute > Instances**.

2. Click the instance's **More** button and select **Console**.

**Figure 3.1. Console Access**



Instance Details: TinyImage

Instance Console

If console is not responding to keyboard input: click the grey status bar below. Click here to show only console
To exit the fullscreen mode, click the browser's back button.

Connected (unencrypted) to: QEMU (instance-00000001)

```
[    1.563471] EFI Variables Facility v0.08 2004-May-17
[    1.566228] TCP cubic registered
[    1.567338] NET: Registered protocol family 10
[    1.582404] NET: Registered protocol family 17
[    1.582694] Registering the dns_resolver key type
[    1.593666] registered taskstats version 1
[    1.673678] Refined TSC clocksource calibration: 2893.419 MHz.
[    1.674010] Switching to clocksource tsc
[    1.765213]   Magic number: 2:192:770
[    1.765213] rtc_cmos 00:01: setting system clock to 2014-06-19 05:46:39 UTC (
1403156799)
[    1.765213] powernow-k8: Processor cpuid 6d3 not supported
[    1.765213] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[    1.765213] EDD information not available.
[    1.800424] Freeing unused kernel memory: 924k freed
[    1.830096] Write protecting the kernel read-only data: 12288k
[    1.872886] Freeing unused kernel memory: 1632k freed
[    1.925062] Freeing unused kernel memory: 1200k freed
[    1.983300] usb 1-1: new full-speed USB device number 2 using uhci_hcd

further output written to /dev/ttyS0


login as 'cirros' user. default password: 'cubswin:)'. use 'sudo' for root.
cirros login: _
```

3. Log in using the image's user name and password (for example, a CirrOS image uses 'cirros'/'cubswin:)').

**Note**

Red Hat Enterprise Linux guest images typically do not allow direct console access; you must SSH into the instance (see Section 3.1.5.4, “SSH into an Instance”).

### 3.1.5.2. Directly Connect to a VNC Console

You can directly access an instance's VNC console using a URL returned by **`nova get-vnc-console`** command.

**Browser**

To obtain a browser URL, use:

```
$ nova get-vnc-console INSTANCE_ID novnc
```

**Java Client**

To obtain a Java-client URL, use:

```
$ nova get-vnc-console INSTANCE_ID xvpvnc
```

**Note**

**nova-xvpvncviewer** provides a simple example of a Java client. To download the client, use:

```
# git clone https://github.com/cloudbuilders/nova-
xvpvncviewer
# cd nova-xvpvncviewer/viewer
# make
```

Run the viewer with the instance's Java-client URL:

```
# java -jar VncViewer.jar URL
```

This tool is provided only for customer convenience, and is not officially supported by Red Hat.

#### 3.1.5.3. Directly Connect to a Serial Console

You can directly access an instance's serial port using a websocket client. Serial connections are typically used as a debugging tool (for example, instances can be accessed even if the network configuration fails). To obtain a serial URL for a running instance, use:

```
$ nova get-serial-console INSTANCE_ID
```

**Note**

**novaconsole** provides a simple example of a websocket client. To download the client, use:

```
# git clone https://github.com/larsks/novaconsole/
# cd novaconsole
```

Run the client with the instance's serial URL:

```
# python console-client-poll.py URL
```

This tool is provided only for customer convenience, and is not officially supported by Red Hat.

However, depending on your installation, the administrator may need to first set up the **nova-serialproxy** service. The proxy service is a websocket proxy that allows connections to OpenStack Compute serial ports.

### Procedure 3.3. Install and Configure nova-serialproxy

1. Install the **nova-serialproxy** service:

   ```
   # yum install openstack-nova-serialproxy
   ```

2. Update the **serial_console** section in **/etc/nova/nova.conf**:

   a. Enable the **nova-serialproxy** service:

      ```
      $ openstack-config --set /etc/nova/nova.conf
      serial_console enabled true
      ```

   b. Specify the string used to generate URLS provided by the **nova get-serial-console** command.

      ```
      $ openstack-config --set /etc/nova/nova.conf
      serial_console base_url ws://PUBLIC_IP:6083/
      ```

      Where *PUBLIC_IP* is the public IP address of the host running the **nova-serialproxy** service.

   c. Specify the IP address on which the instance serial console should listen (string).

      ```
      $ openstack-config --set /etc/nova/nova.conf
      serial_console listen 0.0.0.0
      ```

   d. Specify the address to which proxy clients should connect (string).

      ```
      $ openstack-config --set /etc/nova/nova.conf
      serial_console proxyclient_address ws://HOST_IP:6083/
      ```

      Where *HOST_IP* is the IP address of your Compute host.

   **Example 3.1. Enabled nova-serialproxy**

   ```
   [serial_console]
   enabled=true
   base_url=ws://192.0.2.0:6083/
   listen=0.0.0.0
   proxyclient_address=192.0.2.3
   ```

3. Restart Compute services:

```
# openstack-service restart nova
```

4. Start the **nova-serialproxy** service:

   ```
   # systemctl enable openstack-nova-serialproxy
   # systemctl start openstack-nova-serialproxy
   ```

5. Restart any running instances, to ensure that they are now listening on the right sockets.

6. Open the firewall for serial-console port connections. Serial ports are set using **[serial_console] port_range** in **/etc/nova/nova.conf**; by default, the range is 10000:20000. Update iptables with:

   ```
   # iptables -I INPUT 1 -p tcp --dport 10000:20000 -j ACCEPT
   ```

### 3.1.5.4. SSH into an Instance

1. Ensure that the instance's security group has an SSH rule (see Section 2.1.5, “Manage Project Security”).

2. Ensure the instance has a floating IP address (external address) assigned to it (see Section 3.2.2, “Create, Assign, and Release Floating IP Addresses”).

3. Obtain the instance's key-pair certificate. The certificate is downloaded when the key pair is created; if you did not create the key pair yourself, ask your administrator (see Section 3.2.1, “Manage Key Pairs”).

4. On your local machine, load the key-pair certificate into SSH. For example:

   ```
   $ ssh-add ~/.ssh/os-key.pem
   ```

5. You can now SSH into the file with the user supplied by the image.

   The following example command shows how to SSH into the Red Hat Enterprise Linux guest image with the user 'cloud-user':

   ```
   $ ssh cloud-user@192.0.2.24
   ```

   **Note**

   You can also use the certificate directly. For example:

   ```
   $ ssh -i /myDir/os-key.pem cloud-user@192.0.2.24
   ```

### 3.1.6. View Instance Usage

The following usage statistics are available:

- Per Project

  To view instance usage per project, select **Project > Compute > Overview**. A usage summary is immediately displayed for all project instances.

  You can also view statistics for a specific period of time by specifying the date range and clicking **Submit**.

- Per Hypervisor

  If logged in as an administrator, you can also view information for all projects. Click **Admin > System** and select one of the tabs. For example, the **Resource Usage** tab offers a way to view reports for a distinct time period. You might also click **Hypervisors** to view your current vCPU, memory, or disk statistics.

  **Note**

  The 'vCPU Usage' value ('x of y') reflects the number of total vCPUs of all virtual machines (x) and the total number of hypervisor cores (y).

### 3.1.7. Delete an Instance

1. In the dashboard, select **Project > Compute > Instances**, and select your instance.

2. Click **Terminate Instance**.

**Note**

Deleting an instance does not delete its attached volumes; you must do this separately (see Section 4.2.1.4, “Delete a Volume”).

## 3.2. MANAGE INSTANCE SECURITY

You can manage access to an instance by assigning it the correct security group (set of firewall rules) and key pair (enables SSH user access). Further, you can assign a floating IP address to an instance to enable external network access. The sections below outline how to create and manage key pairs and floating IP addresses. For information on managing security groups, see Section 2.1.5, “Manage Project Security”.

### 3.2.1. Manage Key Pairs

Key pairs provide SSH access to the instances. Each time a key pair is generated, its certificate is downloaded to the local machine and can be distributed to users. Typically, one key pair is created for each project (and used for multiple instances).

You can also import an existing key pair into OpenStack.

#### 3.2.1.1. Create a Key Pair

1. In the dashboard, select **`Project > Compute > Access & Security`**.

2. On the **`Key Pairs`** tab, click **`Create Key Pair`**.

3. Specify a name in the **`Key Pair Name`** field, and click **`Create Key Pair`**.

   When the key pair is created, a key pair file is automatically downloaded through the browser. Save this file for later connections from external machines. For command-line SSH connections, you can load this file into SSH by executing:

   ```
   # ssh-add ~/.ssh/OS-Key.pem
   ```

#### 3.2.1.2. Import a Key Pair

1. In the dashboard, select **`Project > Compute > Access & Security`**.

2. On the **`Key Pairs`** tab, click **`Import Key Pair`**.

3. Specify a name in the **`Key Pair Name`** field, and copy and paste the contents of your public key into the **`Public Key`** field.

4. Click **`Import Key Pair`**.

#### 3.2.1.3. Delete a Key Pair

1. In the dashboard, select **`Project > Compute > Access & Security`**.

2. On the **`Key Pairs`** tab, click the key's **`Delete Key Pair`** button.

### 3.2.2. Create, Assign, and Release Floating IP Addresses

By default, an instance is given an internal IP address when it is first created. However, you can enable access through the public network by creating and assigning a floating IP address (external address). You can change an instance's associated IP address regardless of the instance's state.

Projects have a limited range of floating IP address that can be used (by default, the limit is 50), so you should release these addresses for reuse when they are no longer needed. Floating IP addresses can only be allocated from an existing floating IP pool (see Section 5.2.1, “Create Floating IP Pools”).

### Procedure 3.4. Allocate a Floating IP to the Project

1. In the dashboard, select **`Project > Compute > Access & Security`**.
2. On the **`Floating IPs`** tab, click **`Allocate IP to Project`**.
3. Select a network from which to allocate the IP address in the **`Pool`** field.
4. Click **`Allocate IP`**.

### Procedure 3.5. Assign a Floating IP

1. In the dashboard, select **`Project > Compute > Access & Security`**.
2. On the **`Floating IPs`** tab, click the address' **`Associate`** button.
3. Select the address to be assigned in the **`IP address`** field.

   **Note**

   If no addresses are available, you can click the **+** button to create a new address.

4. Select the instance to be associated in the **`Port to be Associated`** field. An instance can only be associated with one floating IP address.
5. Click **`Associate`**.

### Procedure 3.6. Release a Floating IP

1. In the dashboard, select **`Project > Compute > Access & Security`**.
2. On the **`Floating IPs`** tab, click the address' menu arrow (next to the **`Associate/Disassociate`** button.
3. Select **`Release Floating IP`**.

## 3.3. MANAGE FLAVORS

Each created instance is given a flavor (resource template), which determines the instance's size and capacity. Flavors can also specify secondary ephemeral storage, swap disk, metadata to restrict usage, or special project access (none of the default flavors have these additional attributes defined).

Table 3.1. Default Flavors

| Name | vCPUs | RAM | Root Disk Size |
|---|---|---|---|
| m1.tiny | 1 | 512 MB | 1 GB |
| m1.small | 1 | 2048 MB | 20 GB |
| m1.medium | 2 | 4096 MB | 40 GB |
| m1.large | 4 | 8192 MB | 80 GB |
| m1.xlarge | 8 | 16384 MB | 160 GB |

The majority of end users will be able to use the default flavors. However, you might need to create and manage specialized flavors. For example, you might:

- Change default memory and capacity to suit the underlying hardware needs.
- Add metadata to force a specific I/O rate for the instance or to match a host aggregate.

**Note**

Behavior set using image properties overrides behavior set using flavors (for more information, see Section 4.1, “Manage Images”).

### 3.3.1. Update Configuration Permissions

By default, only administrators can create flavors or view the complete flavor list (select **Admin > System > Flavors**). To allow all users to configure flavors, specify the following in the **/etc/nova/policy.json** file (**nova-api** server):

```
"compute_extension:flavormanage": "",
```

### 3.3.2. Create a Flavor

1. As an admin user in the dashboard, select **Admin > System > Flavors**.
2. Click **Create Flavor**, and specify the following fields:

| Tab | Field | Description |
|---|---|---|
| Flavor Information | Name | Unique name. |
| | ID | Unique ID. The default value, 'auto', generates a UUID4 value, but you can also manually specify an integer or UUID4 value. |
| | VCPUs | Number of virtual CPUs. |
| | RAM (MB) | Memory (in megabytes). |
| | Root Disk (GB) | Ephemeral disk size (in gigabytes); to use the native image size, specify '0'. This disk is not used if 'Instance Boot Source=Boot from Volume'. |
| | Epehemeral Disk (GB) | Secondary ephemeral disk size (in gigabytes). |
| | Swap Disk (MB) | Swap disk size (in megabytes). |
| Flavor Access | Selected Projects | Projects which can use the flavor. If no projects are selected, all projects have access ('Public=Yes'). |

3. Click **`Create Flavor`**.

### 3.3.3. Update General Attributes

1. As an admin user in the dashboard, select **`Admin > System > Flavors`**.

2. Click the flavor's **`Edit Flavor`** button.

3. Update the values, and click **Save**.

### 3.3.4. Update Flavor Metadata

In addition to editing general attributes, you can add metadata to a flavor ('extra_specs'), which can help fine-tune instance usage. For example, you might want to set the maximum-allowed bandwidth or disk writes.

- Pre-defined keys determine hardware support or quotas. Pre-defined keys are limited by the hypervisor you are using (for libvirt, see Table 3.2, “Libvirt Metadata”).

- Both pre-defined and user-defined keys can determine instance scheduling. For example, you might specify 'SpecialComp=True'; any instance with this flavor can then only run in a host aggregate with the same key-value combination in its metadata (see Section 3.4, “Manage Host Aggregates”).

### 3.3.4.1. View Metadata

1. As an admin user in the dashboard, select **`Admin > System > Flavors`**.

2. Click the flavor's **`Metadata`** link ('Yes' or 'No'). All current values are listed on the right-hand side under **`Existing Metadata`**.

### 3.3.4.2. Add Metadata

You specify a flavor's metadata using a ***`key/value`*** pair.

1. As an admin user in the dashboard, select **`Admin > System > Flavors`**.

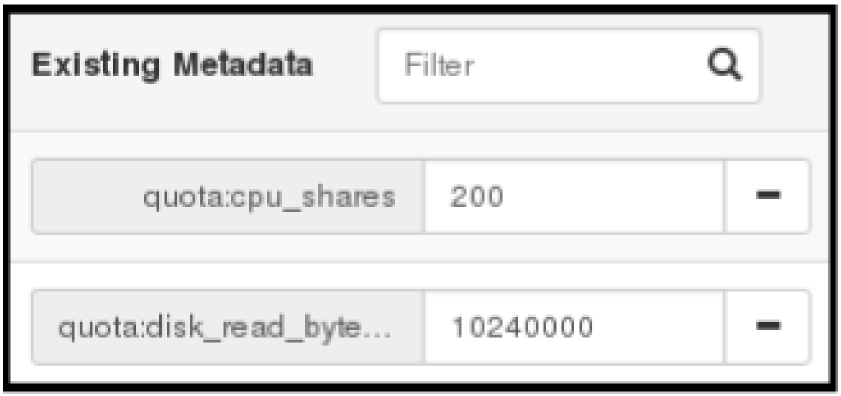
2. Click the flavor's **`Metadata`** link ('Yes' or 'No'). All current values are listed on the right-hand side under **`Existing Metadata`**.

3. Under **`Available Metadata`**, click on the **`Other`** field, and specify the key you want to add (see Table 3.2, “Libvirt Metadata” ).

4. Click the **+** button; you can now view the new key under **`Existing Metadata`**.

5. Fill in the key's value in its right-hand field.

**Figure 3.2. Flavor Metadata**



Existing Metadata

| | | |
|---|---|---|
| quota:cpu_shares | 200 | − |
| quota:disk_read_byte... | 10240000 | − |

6. When finished with adding key-value pairs, click **`Save`**.

**Table 3.2. Libvirt Metadata**

| Key | Description |
|---|---|
| **hw:** *action* | Action that configures support limits per instance. Valid actions are:<br>- **cpu_max_sockets** - Maximum supported CPU sockets.<br>- **cpu_max_cores** - Maximum supported CPU cores.<br>- **cpu_max_threads** - Maximum supported CPU threads.<br>- **cpu_sockets** - Preferred number of CPU sockets.<br>- **cpu_cores** - Preferred number of CPU cores.<br>- **cpu_threads** - Preferred number of CPU threads.<br>- **serial_port_count** - Maximum serial ports per instance.<br>Example: 'hw:cpu_max_sockets=2' |
| **hw:** ***NUMA_def*** | Definition of NUMA topology for the instance. For flavors whose RAM and vCPU allocations are larger than the size of NUMA nodes in the compute hosts, defining NUMA topology enables hosts to better utilize NUMA and improve performance of the guest OS.<br>NUMA definitions defined through the flavor override image definitions. Valid definitions are:<br>- **numa_nodes** - Number of NUMA nodes to expose to the instance. Specify '1' to ensure image NUMA settings are overridden.<br>- **numa_mempolicy** - Memory allocation policy. Valid policies are:<br>  - strict - Mandatory for the instance's RAM allocations to come from the NUMA nodes to which it is bound (default if **numa_nodes** is specified).<br>  - preferred - The kernel can fall back to using an alternative node. Useful when the **numa_nodes** is set to '1'.<br>- **numa_cpus.0** - Mapping of vCPUs N-M to NUMA node 0 (comma-separated list). |

| Key | Description |
|---|---|
| | - **numa_cpus.1** - Mapping of vCPUs N-M to NUMA node 1 (comma-separated list).<br>- **numa_mem.0** - Mapping N GB of RAM to NUMA node 0.<br>- **numa_mem.1** - Mapping N GB of RAM to NUMA node 1.<br>**numa_cpu.N** and **numa_mem.N** are only valid if **numa_nodes** is set. Additionally, they are only required if the instance's NUMA nodes have an asymetrical allocation of CPUs and RAM (important for some NFV workloads). Note: If the values of **numa_cpu** or **numa_mem.N** specify more than that available, an exception is raised.<br>Example when the instance has 8 vCPUs and 4GB RAM:<br>- hw:numa_nodes=2<br>- hw:numa_cpus.0=0,1,2,3,4,5<br>- hw:numa_cpus.1=6,7<br>- hw:numa_mem.0=3<br>- hw:numa_mem.1=1<br>The scheduler looks for a host with 2 NUMA nodes with the ability to run 6 CPUs + 3 GB of RAM on one node, and 2 CPUS + 1 GB of RAM on another node. If a host has a single NUMA node with capability to run 8 CPUs and 4 GB of RAM, it will not be considered a valid match. The same logic is applied in the scheduler regardless of the **numa_mempolicy** setting. |
| **hw:watchdog_action** | An instance watchdog device can be used to trigger an action if the instance somehow fails (or hangs). Valid actions are:<br>- **disabled** - The device is not attached (default value).<br>- **pause** - Pause the instance.<br>- **poweroff** - Forcefully shut down the instance.<br>- **reset** - Forcefully reset the instance.<br>- **none** - Enable the watchdog, but do nothing if the instance fails.<br>Example: 'hw:watchdog_action=poweroff' |

| Key | Description |
|---|---|
| **hw_rng:** *action* | A random-number generator device can be added to an instance using its image properties (see **hw_rng_model** in the "Command-Line Interface Reference" in RHEL OpenStack Platform documentation).<br>If the device has been added, valid actions are:<br>- **allowed** - If 'True', the device is enabled; if 'False', disabled. By default, the device is disabled.<br>- **rate_bytes** - Maximum number of bytes the instance's kernel can read from the host to fill its entropy pool every **rate_period** (integer).<br>- **rate_period** - Duration of the read period in seconds (integer).<br>Example: 'hw_rng:allowed=True'. |
| **hw_video:ram_max_mb** | Maximum permitted RAM to be allowed for video devices (in MB).<br>Example: 'hw:ram_max_mb=64' |
| **quota:** *option* | Enforcing limit for the instance. Valid options are:<br>- **cpu_period** - Time period for enforcing cpu_quota (in microseconds). Within the specified cpu_period, each vCPU cannot consume more than cpu_quota of runtime. The value must be in range [1000, 1000000]; '0' means 'no value'.<br>- **cpu_quota** - Maximum allowed bandwidth (in microseconds) for the vCPU in each cpu_period. The value must be in range [1000, 18446744073709551]. '0' means 'no value'; a negative value means that the vCPU is not controlled. **cpu_quota** and **cpu_period** can be used to ensure that all vCPUs run at the same speed.<br>- **cpu_shares** - Share of CPU time for the domain. The value only has meaning when weighted against other machine values in the same domain. That is, an instance with a flavor with '200' will get twice as much machine time as an instance with '100'. |

| Key | Description |
|---|---|
| | **disk_read_bytes_sec** - Maximum disk reads in bytes per second.<br>**disk_read_iops_sec** - Maximum read I/O operations per second.<br>**disk_write_bytes_sec** - Maximum disk writes in bytes per second.<br>**disk_write_iops_sec** - Maximum write I/O operations per second.<br>**disk_total_bytes_sec** - Maximum total throughput limit in bytes per second.<br>**disk_total_iops_sec** - Maximum total I/O operations per second.<br>**vif_inbound_average** - Desired average of incoming traffic.<br>**vif_inbound_burst** - Maximum amount of traffic that can be received at **vif_inbound_peak** speed.<br>**vif_inbound_peak** - Maximum rate at which incoming traffic can be received.<br>**vif_outbound_average** - Desired average of outgoing traffic.<br>**vif_outbound_burst** - Maximum amount of traffic that can be sent at **vif_outbound_peak** speed.<br>**vif_outbound_peak** - Maximum rate at which outgoing traffic can be sent.<br>Example: 'quota:vif_inbound_average=10240' |

## 3.4. MANAGE HOST AGGREGATES

A single Compute deployment can be partitioned into logical groups for performance or administrative purposes. OpenStack uses the following terms:

- *Host aggregates* - A host aggregate creates logical units in a OpenStack deployment by grouping together hosts. Aggregates are assigned Compute hosts and associated metadata; a host can be in more than one host aggregate. Only administrators can see or create host aggregates.

  An aggregate's metadata is commonly used to provide information for use with the Compute scheduler (for example, limiting specific flavors or images to a subset of hosts). Metadata specified in a host aggregate will limit the use of that host to any instance that has the same metadata specified in its flavor.

Administrators can use host aggregates to handle load balancing, enforce physical isolation (or redundancy), group servers with common attributes, or separate out classes of hardware. When you create an aggregate, a zone name must be specified, and it is this name which is presented to the end user.

- *Availability zones* - An availability zone is the end-user view of a host aggregate. An end user cannot view which hosts make up the zone, nor see the zone's metadata; the user can only see the zone's name.

  End users can be directed to use specific zones which have been configured with certain capabilities or within certain areas.

### 3.4.1. Enable Host Aggregate Scheduling

By default, host-aggregate metadata is not used to filter instance usage; you must update the Compute scheduler's configuration to enable metadata usage:

1. Edit the **/etc/nova/nova.conf** file (you must have either **root** or **nova** user permissions).

2. Ensure that the ***scheduler_default_filters*** parameter contains:

   - 'AggregateInstanceExtraSpecsFilter' for host aggregate metadata. For example:

     ```
     scheduler_default_filters=AggregateInstanceExtraSpecsFilter,Re
     tryFilter,RamFilter,ComputeFilter,ComputeCapabilitiesFilter,I
     magePropertiesFilter,CoreFilter
     ```

   - 'AvailabilityZoneFilter' for availability host specification when launching an instance. For example:

     ```
     scheduler_default_filters=AvailabilityZoneFilter,RetryFilter,R
     amFilter,ComputeFilter,ComputeCapabilitiesFilter,ImagePropert
     iesFilter,CoreFilter
     ```

3. Save the configuration file.

### 3.4.2. View Availability Zones or Host Aggregates

- As an admin user in the dashboard, select **Admin > System > Host Aggregates**. All currently defined aggregates are listed in the **Host Aggregates** section; all zones are in the **Availability Zones** section.

### 3.4.3. Add a Host Aggregate

1. As an admin user in the dashboard, select **Admin > System > Host Aggregates**. All currently defined aggregates are listed in the **Host Aggregates** section.

2. Click **Create Host Aggregate**.

3. Add a name for the aggregate in the **Name** field, and a name by which the end user should see it in the **Availability Zone** field.

4. Click **Manage Hosts within Aggregate**.

5. Select a host for use by clicking its **+** icon.

6. Click **Create Host Aggregate**.

### 3.4.4. Update a Host Aggregate

1. As an admin user in the dashboard, select **Admin > System > Host Aggregates**. All currently defined aggregates are listed in the **Host Aggregates** section.

2. To update the instance's:

   - Name or availability zone:

     - Click the aggregate's **Edit Host Aggregate** button.

     - Update the **Name** or **Availability Zone** field, and click **Save**.

   - Assigned hosts:

     - Click the aggregate's arrow icon under **Actions**.

     - Click **Manage Hosts**.

     - Change a host's assignment by clicking its **+** or **-** icon.

     - When finished, click **Save**.

   - Metatdata:

     - Click the aggregate's arrow icon under **Actions**.

     - Click the **Update Metadata** button. All current values are listed on the right-hand side under **Existing Metadata**.

- Under **Available Metadata**, click on the **Other** field, and specify the key you want to add. Use predefined keys (see Table 3.3, “Host Aggregate Metadata” ) or add your own (which will only be valid if exactly the same key is set in an instance's flavor).

- Click the **+** button; you can now view the new key under **Existing Metadata**.

  **Note:** Remove a key by clicking its **-** icon.

- Click **Save**.

**Table 3.3. Host Aggregate Metadata**

| Key | Description |
|---|---|
| **cpu_allocation_ratio** | Sets allocation ratio of virtual CPU to physical CPU. Depends on the **AggregateCoreFilter** filter being set for the Compute scheduler. |
| **disk_allocation_ratio** | Sets allocation ratio of Virtual disk to physical disk. Depends on the **AggregateDiskFilter** filter being set for the Compute scheduler. |
| **filter_tenant_id** | If specified, the aggregate only hosts this tenant (project). Depends on the **AggregateMultiTenancyIsolation** filter being set for the Compute scheduler. |
| **ram_allocation_ratio** | Sets allocation ratio of virtual RAM to physical RAM. Depends on the **AggregateRamFilter** filter being set for the Compute scheduler. |

### 3.4.5. Delete a Host Aggregate

1. As an admin user in the dashboard, select **Admin > System > Host Aggregates**. All currently defined aggregates are listed in the **Host Aggregates** section.

2. Remove all assigned hosts from the aggregate:

   1. Click the aggregate's arrow icon under **Actions**.

   2. Click **Manage Hosts**.

   3. Remove all hosts by clicking their **-** icon.

4. When finished, click **Save**.

3. Click the aggregate's arrow icon under **Actions**.

4. Click **Delete Host Aggregate** in this and the next dialog screen.

## 3.5. SCHEDULE HOSTS AND CELLS

The Compute scheduling service determines on which cell or host (or host aggregate), an instance will be placed. As an administrator, you can influence where the scheduler will place an instance. For example, you might want to limit scheduling to hosts in a certain group or with the right RAM.

You can configure the following components:

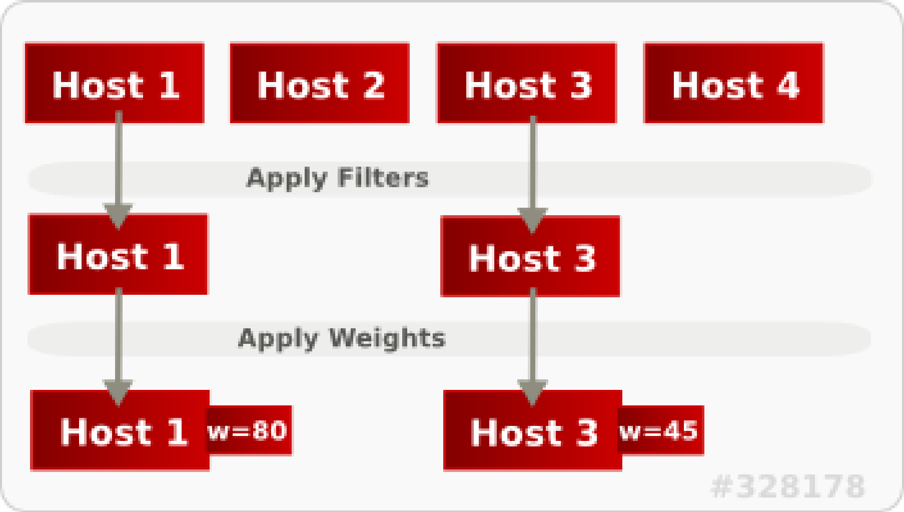
- Filters - Determine the initial set of hosts on which an instance might be placed (see Section 3.5.1, “Configure Scheduling Filters”).

- Weights - When filtering is complete, the resulting set of hosts are prioritized using the weighting system. The highest weight has the highest priority (see Section 3.5.2, “Configure Scheduling Weights”).

- Scheduler service - There are a number of configuration options in the **/etc/nova/nova.conf** file (on the scheduler host), which determine how the scheduler executes its tasks, and handles weights and filters. There is both a host and a cell scheduler. For a list of these options, refer to the "Configuration Reference" (RHEL OpenStack Platform Documentation).

In the following diagram, both host 1 and 3 are eligible after filtering. Host 1 has the highest weight and therefore has the highest priority for scheduling.

**Figure 3.3. Scheduling Hosts**



### 3.5.1. Configure Scheduling Filters

You define which filters you would like the scheduler to use in the **scheduler_default_filters** option (**/etc/nova/nova.conf** file; you must have either **root** or **nova** user permissions). Filters can be added or removed.

By default, the following filters are configured to run in the scheduler:

```
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter
,ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,Server
GroupAntiAffinityFilter,ServerGroupAffinityFilter
```

Some filters use information in parameters passed to the instance in:

- The **nova boot** command, see the "Command-Line Interface Reference" in https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.
- The instance's flavor (see Section 3.3.4, “Update Flavor Metadata”)
- The instance's image (see Appendix A, *Image Configuration Parameters*).

All available filters are listed in the following table.

**Table 3.4. Scheduling Filters**

| Filter | Description |
|---|---|
| AggregateCoreFilter | Uses the host-aggregate metadata key **cpu_allocation_ratio** to filter out hosts exceeding the over-commit ratio (virtual CPU to physical CPU allocation ratio); only valid if a host aggregate is specified for the instance.<br>If this ratio is not set, the filter uses the **cpu_allocation_ratio** value in the **/etc/nova/nova.conf** file. The default value is '16.0' (16 virtual CPU can be allocated per physical CPU). |
| AggregateDiskFilter | Uses the host-aggregate metadata key **disk_allocation_ratio** to filter out hosts exceeding the over-commit ratio (virtual disk to physical disk allocation ratio); only valid if a host aggregate is specified for the instance.<br>If this ratio is not set, the filter uses the **disk_allocation_ratio** value in the **/etc/nova/nova.conf** file. The default value is '1.0' (one virtual disk can be allocated for each physical disk). |

| Filter | Description |
|---|---|
| AggregateImagePropertiesIsolation | Only passes hosts in host aggregates whose metadata matches the instance's image metadata; only valid if a host aggregate is specified for the instance. For more information, see Section 4.1.1, “Create an Image”. |
| AggregateInstanceExtraSpecsFilter | Metadata in the host aggregate must match the host's flavor metadata. For more information, see Section 3.3.4, “Update Flavor Metadata”. |
| AggregateMultiTenancyIsolation | A host with the specified **filter_tenant_id** can only contain instances from that tenant (project). **Note:** The tenant can still place instances on other hosts. |
| AggregateRamFilter | Uses the host-aggregate metadata key **ram_allocation_ratio** to filter out hosts exceeding the over commit ratio (virtual RAM to physical RAM allocation ratio); only valid if a host aggregate is specified for the instance.<br>If this ratio is not set, the filter uses the **ram_allocation_ratio** value in the **/etc/nova/nova.conf** file. The default value is '1.5' (1.5 RAM can be allocated for each physical RAM). |
| AllHostsFilter | Passes all available hosts (however, does not disable other filters). |
| AvailabilityZoneFilter | Filters using the instance's specified availability zone. |
| ComputeCapabilitiesFilter | Ensures Compute metadata is read correctly. Anything before the ':' is read as a namespace. For example, 'quota:cpu_period' uses 'quota' as the namespace and 'cpu_period' as the key. |
| ComputeFilter | Passes only hosts that are operational and enabled. |
| CoreFilter | Uses the **cpu_allocation_ratio** in the **/etc/nova/nova.conf** file to filter out hosts exceeding the over commit ratio(virtual CPU to physical CPU allocation ratio). The default value is '16.0' (16 virtual CPU can be allocated per physical CPU). |
| DifferentHostFilter | Enables an instance to build on a host that is different from one or more specified hosts. Specify 'different' hosts using the **nova boot** option **--different_host** option. |

| Filter | Description |
|---|---|
| DiskFilter | Uses **disk_allocation_ratio** in the **/etc/nova/nova.conf** file to filter out hosts exceeding the over commit ratio(virtual disk to physical disk allocation ratio). The default value is '1.0' (one virtual disk can be allocated for each physical disk). |
| ImagePropertiesFilter | Only passes hosts that match the instance's image properties. For more information, see Section 4.1.1, “Create an Image”. |
| IsolatedHostsFilter | Passes only isolated hosts running isolated images that are specified in the **/etc/nova/nova.conf** file using ***isolated_hosts*** and ***isolated_images*** (comma-separated values). |
| JsonFilter | Recognises and uses an instance's custom JSON filters:<br>- Valid operators are: =, <, >, in, <=, >=, not, or, and<br>- Recognised variables are: **$free_ram_mb**, **$free_disk_mb**, **$total_usable_ram_mb**, **$vcpus_total**, **$vcpus_used**<br>The filter is specfied as a query hint in the **nova boot** command. For example:<br>**--hint query='['>=', '$free_disk_mb', 200 * 1024]'** |
| MetricFilter | Filters out hosts with unavailable metrics. |
| NUMATopologyFilter | Filters out hosts based on its NUMA topology; if the instance has no topology defined, any host can be used. The filter tries to match the exact NUMA topology of the instance to those of the host (it does not attempt to pack the instance onto the host). The filter also looks at the standard over-subscription limits for each NUMA node, and provides limits to the compute host accordingly. |
| RamFilter | Uses **ram_allocation_ratio** in the **/etc/nova/nova.conf** file to filter out hosts exceeding the over commit ratio (virtual RAM to physical RAM allocation ratio). The default value is '1.5' (1.5 RAM can be allocated for each physical RAM). |
| RetryFilter | Filters out hosts that have failed a scheduling attempt; valid if **scheduler_max_attempts** is greater than zero (by default,**scheduler_max_attempts=3**). |

| Filter | Description |
|---|---|
| SameHostFilter | Passes one or more specified hosts; specify hosts for the instance using the ***--hint same_host*** option for **nova boot**. |
| ServerGroupAffinityFilter | Only passes hosts for a specific server group:<br>- Give the server group the affinity policy (**nova server-group-create --policy affinity *groupName***).<br>- Build the instance with that group (**nova boot** option **--hint group=*UUID***). |
| ServerGroupAntiAffinityFilter | Only passes hosts in a server group that do not already host an instance:<br>- Give the server group the anti-affinity policy (**nova server-group-create --policy anti-affinity *groupName***).<br>- Build the instance with that group (**nova boot** option **--hint group=*UUID***). |
| SimpleCIDRAffinityFilter | Only passes hosts on the specified IP subnet range specified by the instance's **cidr** and **build_new_host_ip** hints. Example:<br>**--hint build_near_host_ip=192.0.2.0 --hint cidr=/24** |

### 3.5.2. Configure Scheduling Weights

Both cells and hosts can be weighted for scheduling; the host or cell with the largest weight (after filtering) is selected. All weighers are given a multiplier that is applied after normalising the node's weight. A node's weight is calculated as:

```
w1_multiplier * norm(w1) + w2_multiplier * norm(w2) + ...
```

You can configure weight options in the scheduler host's **/etc/nova/nova.conf** file (must have either **root** or **nova** user permissions).

#### 3.5.2.1. Configure Weight Options for Hosts

You can define the host weighers you would like the scheduler to use in the **[DEFAULT] scheduler_weight_classes** option. Valid weighers are:

- **nova.scheduler.weights.ram** - Weighs the host's available RAM.
- **nova.scheduler.weights.metrics** - Weighs the host's metrics.
- **nova.scheduler.weights.all_weighers** - Uses all host weighers (default).

**Table 3.5. Host Weight Options**

| Weigher | Option | Description |
|---|---|---|
| All | [DEFAULT] scheduler_host_subset_size | Defines the subset size from which a host is selected (integer); must be at least 1. A value of 1 selects the first host returned by the weighing functions. Any value less than 1 is ignored and 1 is used instead (integer value). |
| metrics | [metrics] required | Specifies how to handle metrics in **[metrics] weight_setting** that are unavailable:<br>- **True** - Metrics are required; if unavailable, an exception is raised. To avoid the exception, use the **MetricFilter** filter in the **[DEFAULT]scheduler_default_filters** option.<br>- **False** - The unavailable metric is treated as a negative factor in the weighing process; the returned value is set by **weight_of_unavailable**. |
| metrics | [metrics] weight_of_unavailable | Used as the weight if any metric in **[metrics] weight_setting** is unavailable; valid if **[metrics]required=False**. |
| metrics | [metrics] weight_multiplier | Mulitplier used for weighing metrics. By default, **weight_multiplier=1.0** and spreads instances across possible hosts. If this value is negative, the host with lower metrics is prioritized, and instances are stacked in hosts. |

| Weigher | Option | Description |
|---|---|---|
| metrics | [metrics] weight_setting | Specifies metrics and the ratio with which they are weighed; use a comma-separated list of 'metric=ratio' pairs. Valid metric names are:<br>- **cpu.frequency** - Current CPU frequency<br>- **cpu.user.time** - CPU user mode time<br>- **cpu.kernel.time** - CPU kernel time<br>- **cpu.idle.time** - CPU idle time<br>- **cpu.iowait.time** - CPU I/O wait time<br>- **cpu.user.percent** - CPU user mode percentage<br>- **cpu.kernel.percent** - CPU kernel percentage<br>- **cpu.idle.percent** - CPU idle percentage<br>- **cpu.iowait.percent** - CPU I/O wait percentage<br>- **cpu.percent** - Generic CPU utilization<br>Example: **weight_setting=cpu.user.time=1.0** |
| ram | [DEFAULT] ram_weight_multiplier | Multiplier for RAM (floating point). By default, **ram_weight_multiplier=1.0** and spreads instances across possible hosts. If this value is negative, the host with less RAM is prioritized, and instances are stacked in hosts. |

### 3.5.2.2. Configure Weight Options for Cells

You define which cell weighers you would like the scheduler to use in the **[cells] scheduler_weight_classes** option (**/etc/nova/nova.conf** file; you must have either **root** or **nova** user permissions)

Valid weighers are:

- nova.cells.weights.all_weighers - Uses all cell weighers(default).

- nova.cells.weights.mute_child - Weighs whether a child cell has not sent capacity or capability updates for some time.
- nova.cells.weights.ram_by_instance_type - Weighs the cell's available RAM.
- nova.cells.weights.weight_offset - Evaluates a cell's weight offset. Note: A cell's weight offset is specified using **`--woffset`** in the **`nova-manage cell create`** command.

**Table 3.6. Cell Weight Options**

| Weighers | Option | Description |
|---|---|---|
| mute_child | [cells] mute_weight_multiplier | Multiplier for hosts which have been silent for some time (negative floating point). By default, this value is '-10.0'. |
| mute_child | [cells] mute_weight_value | Weight value given to silent hosts (positive floating point). By default, this value is '1000.0'. |
| ram_by_instance_type | [cells] ram_weight_multiplier | Multiplier for weighing RAM (floating point). By default, this value is '1.0', and spreads instances across possible cells. If this value is negative, the cell with fewer RAM is prioritized, and instances are stacked in cells. |
| weight_offset | [cells] offset_weight_multiplier | Multiplier for weighing cells (floating point). Enables the instance to specify a preferred cell (floating point) by setting its weight offset to 999999999999999 (highest weight is prioritized). By default, this value is '1.0'. |

## 3.6. EVACUATE INSTANCES

If you want to move an instance from a dead or shut-down compute node to a new host server in the same environment (for example, because the server needs to be swapped out), you can evacuate it using **`nova evacuate`**.

- An evacuation is only useful if the instance disks are on shared storage or if the instance disks are Block Storage volumes. Otherwise, the disks will not be accessible and cannot be accessed by the new compute node.
- An instance can only be evacuated from a server if the server is shut down; if the server is not shut down, the **`evacuate`** command will fail.

**Note**

If you have a functioning compute node, and you want to:

- Make a static copy (not running) of an instance for backup purposes or to copy the instance to a different environment, make a snapshot using **`nova image-create`** (see Migrate a Static Instance).
- Move an instance in a static state (not running) to a host in the same environment (shared storage not needed), migrate it using **`nova migrate`** (see Migrate a Static Instance).
- Move an instance in a live state (running) to a host in the same environment, migrate it using **`nova live-migration`** (see Migrate a Live (running) Instance).

### 3.6.1. Evacuate One Instance

Evacuate an instance using:

```
# nova evacuate [--password pass] [--on-shared-storage]
instance_name [target_host]
```

Where:

- **`--password`** ***`pass`*** - Admin password to set for the evacuated instance (cannot be used if **`--on-shared-storage`** is specified). If a password is not specified, a random password is generated and output when evacuation is complete.
- **`--on-shared-storage`** - Indicates that all instance files are on shared storage.
- ***`instance_name`*** - Name of the instance to be evacuated.
- ***`target_host`*** - Host to which the instance is evacuated; if you do not specify the host, the Compute scheduler selects one for you. You can find possible hosts using:

  ```
  # nova host-list | grep compute
  ```

For example:

```
# nova evacuate myDemoInstance Compute2_OnEL7.myDomain
```

### 3.6.2. Evacuate All Instances

Evacuate all instances on a specified host using:

```
# nova host-evacuate instance_name [--target target_host] [--on-
shared-storage] source_host
```

Where:

- **--target *target_host*** - Host to which the instance is evacuated; if you do not specify the host, the Compute scheduler selects one for you. You can find possible hosts using:

  ```
  # nova host-list | grep compute
  ```

- **--on-shared-storage** - Indicates that all instance files are on shared storage.

- ***source_host*** - Name of the host to be evacuated.

For example:

```
# nova host-evacuate --target Compute2_OnEL7.localdomain
myDemoHost.localdomain
```

### 3.6.3. Configure Shared Storage

If you are using shared storage, this procedure exports the instances directory for the Compute service to the two nodes, and ensures the nodes have access. The directory path is set in the ***state_path*** and ***instances_path*** parameters in the **/etc/nova/nova.conf** file. This procedure uses the default value, which is **/var/lib/nova/instances**. Only users with **root** access can set up shared storage.

1. **On the controller host:**

   a. Ensure the **/var/lib/nova/instances** directory has read-write access by the Compute service user (this user must be the same across controller and nodes). For example:

      ```
      drwxr-xr-x.  9 nova nova 4096 Nov  5 20:37 instances
      ```

   b. Add the following lines to the **/etc/exports** file; switch out *node1_IP* and *node2_IP* for the IP addresses of the two compute nodes:

      ```
      /var/lib/nova/instances
      node1_IP(rw,sync,fsid=0,no_root_squash)
      /var/lib/nova/instances
      node2_IP(rw,sync,fsid=0,no_root_squash)
      ```

   c. Export the **/var/lib/nova/instances** directory to the compute nodes.

      ```
      # exportfs -avr
      ```

d. Restart the NFS server:

```
# systemctl restart nfs-server
```

2. **On each compute node:**

a. Ensure the **`/var/lib/nova/instances`** directory exists locally.

b. Add the following line to the **`/etc/fstab`** file:

```
controllerName:/var/lib/nova/instances
/var/lib/nova/instances nfs4 defaults 0 0
```

c. Mount the controller's instance directory (all devices listed in **`/etc/fstab`** ):

```
# mount -a -v
```

d. Ensure qemu can access the directory's images:

```
# chmod o+x /var/lib/nova/instances
```

e. Ensure that the node can see the instances directory with:

```
# ls -ld /var/lib/nova/instances
drwxr-xr-x. 9 nova nova 4096 Nov  5 20:37
/var/lib/nova/instances
```

**Note**

You can also run the following to view all mounted devices:

```
# df -k
```

# CHAPTER 4. IMAGES AND STORAGE

This chapter discusses the steps you can follow to manage images and storage in RHEL OpenStack Platform.

A virtual machine image is a file that contains a virtual disk which has a bootable operating system installed on it. Virtual machine images are supported in different formats. The following are the formats available on RHEL OpenStack Platform:

- **`RAW`** - Unstructured disk image format.
- **`QCOW2`** - Disk format supported by QEMU emulator.
- **`ISO`** - Sector-by-sector copy of the data on a disk, stored in a binary file.
- **`AKI`** - Indicates an Amazon Kernel Image.
- **`AMI`** - Indicates an Amazon Machine Image.
- **`ARI`** - Indicates an Amazon RAMDisk Image.
- **`VDI`** - Disk format supported by VirtualBox virtual machine monitor and the QEMU emulator.
- **`VHD`** - Common disk format used by virtual machine monitors from VMWare, VirtualBox, and others.
- **`VMDK`** - Disk format supported by many common virtual machine monitors.

While we don't normally think of ISO as a virtual machine image format, since ISOs contain bootable filesystems with an installed operating system, you can treat them the same as you treat other virtual machine image files.

To download the official Red Hat Enterprise Linux cloud images, you require a valid Red Hat Enterprise Linux subscription:

- Red Hat Enterprise Linux 7 KVM Guest Image
- Red Hat Enterprise Linux 6 KVM Guest

## 4.1. MANAGE IMAGES

The OpenStack Image service (glance) provides discovery, registration, and delivery services for disk and server images. It provides the ability to copy or snapshot a server image, and immediately store it away. Stored images can be used as a template to get new servers up and running quickly and more consistently than installing a server operating system and individually configuring additional services.

### 4.1.1. Create an Image

This section provides you with the steps to manually create an OpenStack-compatible image in .qcow2 format using a Red Hat Enterprise Linux 7 ISO file.

**Prerequisites:**

- Linux host machine to create an image. This can be any machine on which you can install and run the Linux packages.
- libvirt, virt-manager (run command **yum groupinstall -y @virtualization**). This installs all packages necessary for creating a guest operating system.
- Libguestfs tools (run command **yum install -y libguestfs-tools-c**). This installs a set of tools for accessing and modifying virtual machine images.
- A Red Hat Enterprise Linux 7 ISO file (see Get Installation Media).
- Text editor, if you want to change the **kickstart** files.

**Procedure 4.1. Create an Image**

1. Start the installation using **virt-install** as shown below:

   ```
   # qemu-img create -f qcow2 rhel7.qcow2 8G
   # virt-install --virt-type kvm --name rhel7 --ram 2048 \
   --cdrom /tmp/rhel-server-7.0-x86_64-dvd.iso --disk
   rhel7.qcow2,format=qcow2 \
   --network=bridge:virbr0 --graphics vnc,listen=0.0.0.0 \
   --noautoconsole --os-type=linux --os-variant=rhel7
   ```

   This launches an instance and starts the installation process.

2. After the installation is complete, reboot the instance and log in as the root user.

3. Update the **/etc/sysconfig/network-scripts/ifcfg-eth0** file so it only contains the following values:

   ```
   TYPE=Ethernet
   DEVICE=eth0
   ONBOOT=yes
   BOOTPROTO=dhcp
   NM_CONTROLLED=no
   ```

4. Reboot the machine.

**Note**

Ensure you are subscribed to RHEL 7 using **subscription-manager**. For more information, see Software Repository Configuration

5. Update the system.

   ```
   # yum -y update
   ```

6. Install the **cloud-init** packages and git.

   ```
   # yum install -y cloud-utils-growpart cloud-init
   ```

7. Edit the **/etc/cloud/cloud.cfg** configuration file and under **cloud_init_modules** add:

   ```
   - resolv-conf
   ```

8. Add the following line to **/etc/sysconfig/network** to avoid problems accessing the EC2 metadata service.

   ```
   NOZEROCONF=yes
   ```

9. Power off the instance:

   ```
   # poweroff
   ```

10. Reset and clean the image using the **virt-sysprep** command so it can be to create instances without issues:

    ```
    # virt-sysprep -d rhel7
    ```

11. Reduce image size using the **virt-sparcify** command. This command converts any free space within the disk image back to free space within the host:

    ```
    # virt-sparsify --compress /tmp/rhel7.qcow2 rhel7-cloud.qcow2
    ```

The underlying image file that you created with **qemu-img create** is now ready to be uploaded to the Image service. For more information on uploading this image to your OpenStack deployment using the dashboard, see Section 4.1.2, “Upload an Image”.

### 4.1.2. Upload an Image

1. In the dashboard, select **Project > Compute > Images**.

2. Click **Create Image**.

3. Fill out the values, and click **Create Image** when finished.

| Field | Notes |
|---|---|
| Name | Name for the image. The name must be unique within the project. |
| Description | Brief description to identify the image. |
| Image Source | Image source: **Image Location** or **Image File**. Based on your selection, the next field is displayed. |
| Image Location or Image File | - Select **Image Location** option to specify the image location URL.<br>- Select **Image File** option to upload an image from the local disk. |
| Format | Image format (for example, qcow2). |
| Architecture | Image architecture. For example, use i686 for a 32-bit architecture or x86_64 for a 64-bit architecture. |
| Minimum Disk (GB) | Minimum disk size required to boot the image. If this field is not specified, the default value is 0 (no minimum). |
| Minimum RAM (MB) | Minimum memory size required to boot the image. If this field is not specified, the default value is 0 (no minimum). |
| Public | If selected, makes the image public to all users with access to the project. |
| Protected | If selected, ensures only users with specific permissions can delete this image. |

**Note**

You can also use the **glance image-create** command with the **property** option to create an image. More values are available on the commmand line. For a complete listing, see Appendix A, *Image Configuration Parameters*.

### 4.1.3. Update an Image

1. In the dashboard, select **Project > Compute > Images**.

2. Click **Edit**.

    **Note**

    The **Edit** option is available only when you login as an **admin** user. When you login as a **demo** user, you have the option to **Launch** an instance or **Create Volume**.

3. Update the fields and click **Update Image** when finished. You can update the following values - name, description, kernel ID, ramdisk ID, architecture, format, minimum disk, minimum RAM, public, protected.

4. Click the dropdown menu and select **Update Metadata** option.

5. Specify metadata by adding items from the left column to the right one. In the left column, there are metadata definitions from the Image Service Metadata Catalog. Select **Other** to add metadata with the key of your choice and click **Save** when finished.

**Note**

You can also use the **glance image-update** command with the **property** option to update an image. More values are available on the commmand line; for a complete listing, see Appendix A, *Image Configuration Parameters*.

### 4.1.4. Delete an Image

1. In the dashboard, select **Project > Compute > Images**.

2. Select the image you want to delete and click **Delete Images**.

## 4.2. MANAGE VOLUMES

A volume is a block storage device that provides persistent storage to OpenStack instances.

### 4.2.1. Basic Volume Usage and Configuration

The following procedures describe how to perform basic end-user volume management. These procedures do not require administrative privileges.

#### 4.2.1.1. Create a Volume

1. In the dashboard, select **Project > Compute > Volumes**.

2. Click **Create Volume**, and edit the following fields:

| Field | Description |
|---|---|
| Volume name | Name of the volume. |
| Description | Optional, short description of the volume. |
| Type | Optional volume type (see Section 4.2.4, “Group Volume Settings with Volume Types”).<br>If you have multiple Block Storage back ends, you can use this to select a specific back end. See Section 4.2.1.2, “Specify Back End for Volume Creation” for details. |
| Size (GB) | Volume size (in gigabytes). |
| Availability Zone | Availability zones (logical server groups), along with host aggregates, are a common method for segregating resources within OpenStack. Availability zones are defined during installation. For more information on availability zones and host aggregates, see Section 3.4, “Manage Host Aggregates”. |

3. Specify a **`Volume Source`**:

| Source | Description |
|---|---|
| No source, empty volume | The volume will be empty, and will not contain a file system or partition table. |
| Snapshot | Use an existing snapshot as a volume source. If you select this option, a new **`Use snapshot as a source`** list appears; you can then choose a snapshot from the list. For more information about volume snapshots, refer to Section 4.2.1.8, “Create, Clone, or Delete Volume Snapshots”. |
| Image | Use an existing image as a volume source. If you select this option, a new **`Use image as a source`** lists appears; you can then choose an image from the list. |
| Volume | Use an existing volume as a volume source. If you select this option, a new **`Use volume as a source`** list appears; you can then choose a volume from the list. |

4. Click **`Create Volume`**. After the volume is created, its name appears in the **`Volumes`** table.

#### 4.2.1.2. Specify Back End for Volume Creation

You can configure the Block Storage service to use multiple back ends. For example, Configure OpenStack to Use an NFS Back End provides step-by-step instructions on how to configure the Block Storage service to use an NFS share alongside the default back end.

Whenever multiple Block Storage back ends are configured, you will also need to create a volume type for each back end. You can then use the type to specify which back end should be used for a created volume. For more information about volume types, see Section 4.2.4, “Group Volume Settings with Volume Types”.

To specify a back end when creating a volume, select its corresponding volume type from the **Type** drop-down list (see Section 4.2.1.1, “Create a Volume”).

If you do not specify a back end during volume creation, the Block Storage service will automatically choose one for you. By default, the service will choose the back end with the most available free space. You can also configure the Block Storage service to choose randomly among all available back ends instead; for more information, see Section 4.2.7, “Configure How Volumes are Allocated to Multiple Back Ends”.

#### 4.2.1.3. Edit a Volume's Name or Description

1. In the dashboard, select **Project > Compute > Volumes**.

2. Select the volume's **Edit Volume** button.

3. Edit the volume name or description as required.

4. Click **Edit Volume** to save your changes.

**Note**

To create an encrypted volume, you must first have a volume type configured specifically for volume encryption. In addition, both Compute and Block Storage services must be configured to use the same static key. For information on how to set up the requirements for volume encryption, refer to Section 4.2.6, “Encrypt Volumes with Static Keys”.

#### 4.2.1.4. Delete a Volume

1. In the dashboard, select **Project > Compute > Volumes**.

2. In the **Volumes** table, select the volume to delete.

3. Click **Delete Volumes**.

**Note**

A volume cannot be deleted if it has existing snapshots. For instructions on how to delete snapshots, see Section 4.2.1.8, “Create, Clone, or Delete Volume Snapshots”.

### 4.2.1.5. Attach and Detach a Volume to an Instance

Instances can use a volume for persistent storage. A volume can only be attached to one instance at a time. For more information on instances, see Section 3.1, “Manage Instances”.

**Procedure 4.2. Attach a Volume from an Instance**

1. In the dashboard, select **`Project > Compute > Volumes`**.
2. Select the volume's **`Edit Attachments`** action. If the volume is not attached to an instance, the **`Attach To Instance`** drop-down list is visible.
3. From the **`Attach To Instance`** list, select the instance to which you wish to attach the volume.
4. Click **`Attach Volume`**.

**Procedure 4.3. Detach a Volume From an Instance**

1. In the dashboard, select **`Project > Compute > Volumes`**.
2. Select the volume's **`Edit Attachments`** action. If the volume is attached to an instance, the instance's name is displayed in the **`Attachments`** table.
3. Click **`Detach Volume`** in this and the next dialog screen.

### 4.2.1.6. Set a Volume to Read-Only

You can give multiple users shared access to a single volume without allowing them to edit its contents. To do so, set the volume to **`read-only`** using the following command:

```
# cinder readonly-mode-update VOLUME true
```

Replace *VOLUME* with the ID of the target volume.

To set a read-only volume back to read-write, run:

```
# cinder readonly-mode-update VOLUME false
```

#### 4.2.1.7. Change a Volume's Owner

To change a volume's owner, you will have to perform a *volume transfer*. A volume transfer is initiated by the volume's owner, and the volume's change in ownership is complete after the transfer is accepted by the volume's new owner.

1. From the command line, log in as the volume's current owner.

2. List the available volumes:

   ```
   # cinder list
   ```

3. Initiate the volume transfer:

   ```
   # cinder transfer-create VOLUME
   ```

   Where *VOLUME* is the name or ID of the volume you wish to transfer.

   **Example 4.1.**

   ```
   # cinder transfer-create samplevolume
   +------------+--------------------------------------+
   |  Property  |                Value                 |
   +------------+--------------------------------------+
   |  auth_key  |           f03bf51ce7ead189           |
   | created_at |      2014-12-08T03:46:31.884066      |
   |     id     | 3f5dc551-c675-4205-a13a-d30f88527490 |
   |    name    |                 None                 |
   | volume_id  | bcf7d015-4843-464c-880d-7376851ca728 |
   +------------+--------------------------------------+
   ```

   The **`cinder transfer-create`** command clears the ownership of the volume and creates an **`id`** and **`auth_key`** for the transfer. These values can be given to, and used by, another user to accept the transfer and become the new owner of the volume.

4. The new user can now claim ownership of the volume. To do so, the user should first log in from the command line and run:

   ```
   # cinder transfer-accept TRANSFERID TRANSFERKEY
   ```

   Where *TRANSFERID* and *TRANSFERKEY* are the **`id`** and **`auth_key`** values returned by the **`cinder transfer-create`** command, respectively.

   **Example 4.2.**

```
# cinder transfer-accept 3f5dc551-c675-4205-a13a-
d30f88527490 f03bf51ce7ead189
```

**Note**

You can view all available volume transfers using:

```
# cinder transfer-list
```

### 4.2.1.8. Create, Clone, or Delete Volume Snapshots

You can preserve a volume's state at a specific point in time by creating a volume snapshot. You can then use the snapshot to clone new volumes.

**Warning**

Creating a snapshot of a volume that is attached to an instance may corrupt the snapshot. For instructions on how to detach a volume from an instance, see Procedure 4.3, “Detach a Volume From an Instance”.

**Note**

Volume backups are different from snapshots. Backups preserve the data contained in the volume, whereas snapshots preserve the state of a volume at a specific point in time. In addition, you cannot delete a volume if it has existing snapshots. Volume backups are used to prevent data loss, whereas snapshots are used to facilitate cloning.

For more information about volume backups, refer to Section 4.2.2, “Back Up and Restore a Volume”.

To create a volume snapshot:

1. In the dashboard, select **Project > Compute > Volumes**.

2. Select the target volume's **Create Snapshot** action.

3. Provide a **Snapshot Name** for the snapshot and click **Create a Volume Snapshot**. The **Volume Snapshots** tab displays all snapshots.

You can clone new volumes from a snapshot once it appears in the **Volume Snapshots** table. To do so, select the snapshot's **Create Volume** action. For more information about volume creation, see Section 4.2.1.1, “Create a Volume”.

To delete a snapshot, select its **`Delete Volume Snapshot`** action.

If your OpenStack deployment uses a Red Hat Ceph back end, see Section 4.2.1.8.1, “Protected and Unprotected Snapshots in a Red Hat Ceph Back End” for more information on snapshot security and troubleshooting.

#### 4.2.1.8.1. Protected and Unprotected Snapshots in a Red Hat Ceph Back End

When using Red Hat Ceph as a back end for your OpenStack deployment, you can set a snapshot to *protected* in the back end. Attempting to delete protected snapshots through OpenStack (as in, through the dashboard or the **`cinder snapshot-delete`** command) will fail.

When this occurs, set the snapshot to *unprotected* in the Red Hat Ceph back end first. Afterwards, you should be able to delete the snapshot through OpenStack as normal.

For related instructions, see Protecting a Snapshot and Unprotecting a Snapshot.

### 4.2.1.9. Upload a Volume to the Image Service

You can upload an existing volume as an image to the Image service directly. To do so:

1. In the dashboard, select **`Project > Compute > Volumes`**.

2. Select the target volume's **`Upload to Image`** action.

3. Provide an **`Image Name`** for the volume and select a **`Disk Format`** from the list.

4. Click **`Upload`**. The QEMU disk image utility uploads a new image of the chosen format using the name you provided.

To view the uploaded image, select **`Project > Compute > Images`**. The new image appears in the **`Images`** table. For information on how to use and configure images, see Section 4.1, “Manage Images”.

## 4.2.2. Back Up and Restore a Volume

A volume backup is a full, persistent copy of a volume's contents. Volume backups are typically created as object stores, and therefore are managed through the Object Storage service.

When creating a volume backup, all of the backup's metadata is stored in the Block Storage service's database. The **`cinder`** utility uses this metadata when restoring a volume from the backup. As such, when recovering from a catastrophic database loss, you must restore the Block Storage service's database first before restoring any volumes from backups. This also presumes that the Block Storage service database is being restored with all the original volume backup metadata intact.

If you wish to configure only a subset of volume backups to survive a catastrophic database loss, you can also export the backup's metadata. In doing so, you can then re-import the metadata to the Block Storage database later on, and restore the volume backup as normal.

**Note**

Volume backups are different from snapshots. Backups preserve the data contained in the volume, whereas snapshots preserve the state of a volume at a specific point in time. In addition, you cannot delete a volume if it has existing snapshots. Volume backups are used to prevent data loss, whereas snapshots are used to facilitate cloning.

For more information about volume snapshots, refer to Section 4.2.1.8, “Create, Clone, or Delete Volume Snapshots”.

### 4.2.2.1. Create a Volume Backup

1. As a user with administrative privileges, view the **`ID`** or **`Display Name`** of the volume you wish to back-up:

   ```
   # cinder list
   ```

2. Back up the volume:

   ```
   # cinder backup-create VOLUME
   ```

   Where *VOLUME* is the **`ID`** or **`Display Name`** of the volume you wish to back-up.

   **Example 4.3.**

   ```
   # cinder backup-create volumename
   +-----------+--------------------------------------+
   |  Property |                Value                 |
   +-----------+--------------------------------------+
   |     id    | e9d15fc7-eeae-4ca4-aa72-d52536dc551d |
   |    name   |                 None                 |
   | volume_id | 5f75430a-abff-4cc7-b74e-f808234fa6c5 |
   +-----------+--------------------------------------+
   ```

   Note that the **`volume_id`** of the resulting backup is identical to the **`ID`** of **`volumename`**.

3. Verify that the volume backup creation is complete:

   ```
   # cinder backup-list
   ```

The volume backup creation is complete when the **Status** of the backup entry is **available**.

At this point, you can also export and store the volume backup's metadata. This allows you to restore the volume backup, even if the Block Storage database suffers a catastrophic loss. To do so, run:

```
# cinder --os-volume-api-version 2 backup-export BACKUPID
```

Where *BACKUPID* is the ID or name of the volume backup.

**Example 4.4.**

```
# cinder --os-volume-api-version 2 backup-export e9d15fc7-eeae-
4ca4-aa72-d52536dc551d
+----------------+------------------------------------------------
-------------------+
|    Property    |                                    Value
|
+----------------+------------------------------------------------
-------------------+
| backup_service |
cinder.backup.drivers.swift              |
|   backup_url   | eyJzdGF0dXMi...
|
|                | ...c2l6ZSI6IDF9
|
+----------------+------------------------------------------------
-------------------+
```

The volume backup metadata consists of the **backup_service** and **backup_url** values.

#### 4.2.2.2. Restore a Volume After a Block Storage Database Loss

Typically, a Block Storage database loss prevents you from restoring a volume backup. This is because the Block Storage database contains metadata required by the volume backup. This metadata consists of **backup_service** and **backup_url** values, which you can export after creating the volume backup (as shown in Section 4.2.2.1, “Create a Volume Backup”).

If you exported and stored this metadata, then you can import it to a new Block Storage database (thereby allowing you to restore the volume backup).

1. As a user with administrative privileges, run:

   ```
   # cinder --os-volume-api-version 2 backup-import
   backup_service backup_url
   ```

Where *backup_service* and *backup_url* are from the metadata you exported.

> **Example 4.5.**
>
> Using the exported sample metadata from Section 4.2.2.1, “Create a Volume Backup”:
>
> ```
> # cinder --os-volume-api-version 2 backup-import
> cinder.backup.drivers.swift eyJzdGF0dXMi...c2l6ZSI6IDF9
> +----------+--------------------------------------+
> | Property |                Value                 |
> +----------+--------------------------------------+
> |    id    | 77951e2f-4aff-4365-8c64-f833802eaa43 |
> |   name   |                 None                 |
> +----------+--------------------------------------+
> ```

2. After the metadata is imported into the Block Storage service database, you can restore the volume as normal (see Section 4.2.2.3, “Restore a Volume from a Backup”).

#### 4.2.2.3. Restore a Volume from a Backup

1. As a user with administrative privileges, find the **ID** of the volume backup you wish to use:

   ```
   # cinder backup-list
   ```

   The **Volume ID** should match the ID of the volume you wish to restore.

2. Restore the volume backup:

   ```
   # cinder backup-restore BACKUP_ID
   ```

   Where *BACKUP_ID* is the ID of the volume backup you wish to use.

3. If you no longer need the backup, delete it:

   ```
   # cinder backup-delete BACKUP_ID
   ```

### 4.2.3. Migrate a Volume

Only an administrator can migrate volumes; volumes to be migrated cannot be in use nor can they have any snapshots.

1. As an administrative user, list all available volumes:

```
# cinder list
```

2. List the available back ends (hosts) and their respective availability zones:

```
# cinder-manage host list
```

3. Initiate the migration:

```
# cinder migrate VOLUME BACKEND
```

Where:

- *VOLUME* is the ID of the volume to be migrated.
- *BACKEND* is the back end to where the volume should be migrated.

4. View the current status of the volume to be migrated:

```
# cinder show VOLUME
```

**Example 4.6.**

```
# cinder show 45a85c3c-3715-484d-ab5d-745da0e0bd5a
+----------------------------------------+------------------
--------------------+
|                Property                |
Value                 |
+----------------------------------------+------------------
--------------------+
|                  ...                   |
...                  |
|         os-vol-host-attr:host          |
server1               |
|     os-vol-mig-status-attr:migstat     |
None                 |
|                  ...                   |
...                  |
+----------------------------------------+------------------
--------------------+
```

During migration, note the following attributes:

**os-vol-host-attr:host**

The volume's current back end. Once the migration completes, this displays the target back end (namely, *BACKEND*).

**os-vol-mig-status-attr:migstat**

The status of the migration. A status of ***None*** means a migration is no longer in progress.

### 4.2.4. Group Volume Settings with Volume Types

OpenStack allows you to create *volume types*, which allows you apply the type's associated settings when creating a volume (Section 4.2.1.1, “Create a Volume”). For example, you can associate:

- Whether or not a volume is encrypted (Section 4.2.6.2, “Configure Volume Type Encryption”)
- Which back end a volume should use (Section 4.2.1.2, “Specify Back End for Volume Creation”)
- Quality-of-Service (QoS) Specs

Settings are associated with volume types using key-value pairs called *Extra Specs*. When you specify a volume type during volume creation, the Block Storage scheduler applies these key/value pairs as settings. You can associate multiple key/value pairs to the same volume type.

> **Example 4.7.**
>
> Volume types provide the capability to provide different users with storage tiers. By associating specific performance, resilience, and other settings as key/value pairs to a volume type, you can map tier-specific settings to different volume types. You can then apply tier settings when creating a volume by specifying the corresponding volume type.

**Note**

Available and supported Extra Specs vary per volume driver. Consult your volume driver's documentation for a list of valid Extra Specs.

#### 4.2.4.1. Create and Configure a Volume Type

1. As an admin user in the dashboard, select **Admin > Volumes > Volume Types**.
2. Click **Create Volume Type**.
3. Enter the volume type name in the **Name** field.
4. Click **Create Volume Type**. The new type appears in the **Volume Types** table.

5. Select the volume type's **View Extra Specs** action.

6. Click **Create**, and specify the **Key** and **Value**. The key/value pair must be valid; otherwise, specifying the volume type during volume creation will result in an error.

7. Click **Create**. The associated setting (key/value pair) now appears in the **Extra Specs** table.

> **Note**
>
> You can also associate a QOS Spec to the volume type. For details, refer to Section 4.2.5.2, “Associate a QOS Spec with a Volume Type”.

### 4.2.4.2. Edit a Volume Type

1. As an admin user in the dashboard, select **Admin > Volumes > Volume Types**.

2. In the **Volume Types** table, select the volume type's **View Extra Specs** action.

3. On the **Extra Specs** table of this page, you can:

   - Add a new setting to the volume type. To do this, click **Create**, and specify the key/value pair of the new setting you wish to associate to the volume type.

   - Edit an existing setting associated with the volume type. To do this, select the settings **Edit** action.

   - Delete existing settings associated with the volume type. To do this, select the extra specs' check box and click **Delete Extra Specs** in this and the next dialog screen.

### 4.2.4.3. Delete a Volume Type

To delete a volume type, select its corresponding checkboxes from the **Volume Types** table and click **Delete Volume Types**.

## 4.2.5. Use Quality-of-Service Specifications

You can map multiple performance settings to a single Quality-of-Service specification (*QOS Specs*). Doing so allows you to provide performance tiers for different user types.

Performance settings are mapped as key/value pairs to QOS Specs, similar to the way volume settings are associated to a volume type. However, QOS Specs are different from volume types in the following respects:

- QOS Specs are used to apply performance settings, which include limiting read/write operations to disks. Available and supported performance settings vary per storage driver.

  To determine which QOS Specs are supported by your back end, consult the documentation of your back end device's volume driver.

- Volume types are directly applied to volumes, whereas QOS Specs are not. Rather, QOS Specs are associated to volume types. During volume creation, specifying a volume type also applies the performance settings mapped to the volume type's associated QOS Specs.

### 4.2.5.1. Create and Configure a QOS Spec

As an administrator, you can create and configure a QOS Spec through the **`QOS Specs`** table. You can associate more than one key/value pair to the same QOS Spec.

1. As an admin user in the dashboard, select **`Admin > Volumes > Volume Types`**.

2. On the **`QOS Specs`** table, click **`Create QOS Spec`**.

3. Enter a name for the QOS Spec.

4. In the **`Consumer`** field, specify where the QOS policy should be enforced:

   **Table 4.1. Consumer Types**

   | Type | Description |
   |---|---|
   | back-end | QOS policy will be applied to the Block Storage back end. |
   | front-end | QOS policy will be applied to Compute. |
   | both | QOS policy will be applied to both Block Storage and Compute. |

5. Click **`Create`**. The new QOS Spec should now appear in the **`QOS Specs`** table.

6. In the QOS Specs table, select the new spec's **`Manage Specs`** action.

7. Click **`Create`**, and specify the **`Key`** and **`Value`**. The key/value pair must be valid; otherwise, specifying a volume type associated with this QOS Spec during volume creation will fail.

8. Click **`Create`**. The associated setting (key/value pair) now appears in the **`Key-Value Pairs`** table.

#### 4.2.5.2. Associate a QOS Spec with a Volume Type

As an administrator, you can associate a QOS Spec to an existing volume type using the **`Volume Types`** table.

1. As an administrator in the dashboard, select **`Admin > Volumes > Volume Types`**.

2. In the **`Volume Types`** table, select the type's **`Manage QOS Spec Association`** action.

3. Select a QOS Spec from the **`QOS Spec to be associated`** list.

4. Click **`Associate`**. The selected QOS Spec now appears in the **`Associated QOS Spec`** column of the edited volume type.

#### 4.2.5.3. Disassociate a QOS Spec from a Volume Type

1. As an administrator in the dashboard, select **`Admin > Volumes > Volume Types`**.

2. In the **`Volume Types`** table, select the type's **`Manage QOS Spec Association`** action.

3. Select 'None' from the **`QOS Spec to be associated`** list.

4. Click **`Associate`**. The selected QOS Spec is no longer in the **`Associated QOS Spec`** column of the edited volume type.

### 4.2.6. Encrypt Volumes with Static Keys

Volume encryption helps provide basic data protection in case the volume back-end is either compromised or outright stolen. The contents of an encrypted volume can only be read with the use of a specific key; both Compute and Block Storage services must be configured to use the same key in order for instances to use encrypted volumes. This section describes how to configure an OpenStack deployment to use a single key for encrypting volumes.

#### 4.2.6.1. Configure a Static Key

The first step in implementing basic volume encryption is to set a *static key*. This key must be a hex string, which will be used by the Block Storage volume service (namely, **`openstack-cinder-volume`**) and all Compute services (**`openstack-nova-compute`**). To configure both services to use this key, set the key as the **`fixed_key`** value in the **`[keymgr]`** section of both service's respective configuration files.

1. From the command line, log in as **`root`** to the node hosting **`openstack-cinder-volume`**.

2. Set the static key:

   ```
   # openstack-config --set /etc/cinder/cinder.conf keymgr
   fixed_key HEX_KEY
   ```

   Replace *HEX_KEY* with a 16-digit alphanumeric hex key (for example, **0000000000000000000000000000000000000000000000000000000000000000**).

3. Restart the Block Storage volume service:

   ```
   # openstack-service restart cinder-volume
   ```

4. Log in to the node hosting **openstack-nova-compute**, and set the same static key:

   ```
   # openstack-config --set /etc/nova/nova.conf keymgr fixed_key
   HEX_KEY
   ```

   **Note**

   If you have multiple Compute nodes (multiple nodes hosting **openstack-nova-compute**), then you need to set the same static key in **/etc/nova/nova.conf** of each node.

5. Restart the Compute service:

   ```
   # openstack-service restart nova-compute
   ```

   **Note**

   Likewise, if you set the static key on multiple Compute nodes, you need to restart the **openstack-nova-compute** service on each node as well.

At this point, both Compute and Block Storage volume services can now use the same static key to encrypt/decrypt volumes. That is, new instances will be able to use volumes encrypted with the static key (*HEX_KEY*).

#### 4.2.6.2. Configure Volume Type Encryption

To create volumes encrypted with the static key from Section 4.2.6.1, “Configure a Static Key”, you need an *encrypted volume type*. Configuring a volume type as encrypted involves setting what provider class, cipher, and key size it should use. To do so, run:

```
# cinder encryption-type-create --cipher aes-xts-plain64 --key_size
BITSIZE --control_location front-end VOLTYPE
nova.volume.encryptors.luks.LuksEncryptor
```

Where:

- *BITSIZE* is the key size (for example, **`512`** for a 512-bit key).
- *VOLTYPE* is the name of the volume type you want to encrypt.

This command sets the **`nova.volume.encryptors.luks.LuksEncryptor`** provider class and **`aes-xts-plain64`** cipher. As of this release, this is the only supported class/cipher configuration for volume encryption.

Once you have an encrypted volume type, you can invoke it to automatically create encrypted volumes. Specifically, select the encrypted volume type from the **`Type`** drop-down list in the **Create Volume** window (see to Section 4.2.1, “Basic Volume Usage and Configuration”).

### 4.2.7. Configure How Volumes are Allocated to Multiple Back Ends

If the Block Storage service is configured to use multiple back ends, you can use configured volume types to specify where a volume should be created. For details, see Section 4.2.1.2, “Specify Back End for Volume Creation”.

Conversely, you can also configure how the Block Storage service should choose a back end if none is specified during volume creation. To do so:

1. Enable the **`FilterScheduler`**.

   ```
   # openstack-config --set /etc/cinder/cinder.conf DEFAULT
   scheduler_driver
   cinder.scheduler.filter_scheduler.FilterScheduler
   # openstack-config --set /etc/cinder/cinder.conf DEFAULT
   scheduler_default_filters
   AvailabilityZoneFilter,CapacityFilter,CapabilitiesFilter
   ```

   The **`FilterScheduler`** uses the following filters to list all suitable back ends:

   **AvailabilityZoneFilter**

   Filters out all back ends that do not meet the availability zone requirements of the requested volume

   **CapacityFilter**

   Selects only back ends with enough space to accommodate the volume

   **CapabilitiesFilter**

   Selects only back ends that can support any specified settings in the volume

2. Configure how the scheduler should select a suitable back end. If you want the scheduler:

   - To always choose the back end with the most available free space, run:

     ```
     # openstack-config --set /etc/cinder/cinder.conf DEFAULT
     scheduler_default_weighers AllocatedCapacityWeigher
     # openstack-config --set /etc/cinder/cinder.conf DEFAULT
     sallocated_capacity_weight_multiplier -1.0
     ```

   - To choose randomly among all suitable back ends, run:

     ```
     # openstack-config --set /etc/cinder/cinder.conf DEFAULT
     scheduler_default_weighers ChanceWeigher
     ```

3. Restart the Block Storage scheduler to apply your settings:

   ```
   # openstack-service restart openstack-cinder-scheduler
   ```

## 4.3. MANAGE CONTAINERS

OpenStack Object Storage (swift) stores its objects (data) in containers, which are similar to directories in a file system although they cannot be nested. Containers provide an easy way for users to store any kind of unstructured data; for example, objects might include photos, text files, or images. Stored objects are not encrypted nor are they compressed.

To help with organization, pseudo-folders are logical devices that can contain objects (and can be nested). For example, you might create an 'Images' folder in which to store pictures and a 'Media' folder in which to store videos.

You can create one or more containers in each project, and one or more objects or pseudo-folders in each container.

### 4.3.1. Create a Container

1. In the dashboard, select **Project > Object Store > Containers**.

2. Click **Create Container**.

3. Specify the **Container Name**, and select one of the following in the **Container Access** field.

| Type | Description |
|---|---|
| Private | Limits access to a user in the current project. |

| Type | Description |
|---|---|
| Public | Permits API access to anyone with the public URL. However, in the dashboard, project users cannot see public containers and data from other projects. |

4. Click **`Create Container`**.

### 4.3.2. Create Pseudo Folder for Container

1. In the dashboard, select **`Project > Object Store > Containers`**.
2. Click the name of the container to which you want to add the pseudo-folder.
3. Click **`Create Pseudo-folder`**.
4. Specify the name in the **`Pseudo-folder Name`** field, and click **`Create`**.

### 4.3.3. Upload an Object

If you do not upload an actual file, the object is still created (as placeholder) and can later be used to upload the file.

1. In the dashboard, select **`Project > Object Store > Containers`**.
2. Click the name of the container in which the uploaded object will be placed; if a pseudo-folder already exists in the container, you can click its name.
3. Browse for your file, and click **`Upload Object`**.
4. Specify a name in the **`Object Name`** field:
   - Pseudo-folders can be specified in the name using a '/' character (for example, 'Images/myImage.jpg'). If the specified folder does not already exist, it is created when the object is uploaded.
   - A name that is not unique to the location (that is, the object already exists) overwrites the object's contents.
5. Click **`Upload Object`**.

### 4.3.4. Copy an Object

1. In the dashboard, select **`Project > Object Store > Containers`**.

2. Click the name of the object's container or folder (to display the object).

3. Click **`Upload Object`**.

4. Browse for the file to be copied, and select **`Copy`** in its arrow menu.

5. Specify the following:

| Field | Description |
|---|---|
| Destination container | Target container for the new object. |
| Path | Pseudo-folder in the destination container; if the folder does not already exist, it is created. |
| Destination object name | New object's name. If you use a name that is not unique to the location (that is, the object already exists), it overwrites the object's previous contents. |

6. Click **`Copy Object`**.

### 4.3.5. Delete an Object

1. In the dashboard, select **`Project > Object Store > Containers`**.

2. Browse for the object, and select **`Delete Object`** in its arrow menu.

3. Click **`Delete Object`** to confirm the object's removal.

### 4.3.6. Delete a Container

1. In the dashboard, select **`Project > Object Store > Containers`**.

2. Browse for the container in the **`Containers`** section, and ensure all objects have been deleted (see Section 4.3.5, “Delete an Object”).

3. Select **`Delete Container`** in the container's arrow menu.

4. Click **`Delete Container`** to confirm the container's removal.

# CHAPTER 5. NETWORKING

OpenStack Networking (neutron) is the software-defined networking component of RHEL OpenStack Platform. The virtual network infrastructure enables connectivity between instances and the physical external network.

## 5.1. MANAGE NETWORK RESOURCES

Add and remove OpenStack Networking resources such as subnets and routers to suit your RHEL OpenStack Platform deployment.

### 5.1.1. Add a Network

Create a network to give your instances a place to communicate with each other and receive IP addresses using DHCP.

A network can also be integrated with external networks in your RHEL OpenStack Platform deployment or elsewhere, such as the physical network. This integration allows your instances to communicate with, and be reachable by, outside systems. To integrate your network with your physical external network, see Section 5.3, “Bridge the physical network”.

When creating networks, it is important to know that networks can host multiple subnets. This is useful if you intend to host distinctly different systems in the same network, and would prefer a measure of isolation between them. For example, you can designate that only webserver traffic is present on one subnet, while database traffic traverse another. Subnets are isolated from each other, and any instance that wishes to communicate with another subnet must have their traffic directed by a router. Consider placing systems that will require a high volume of traffic amongst themselves in the same subnet, so that they don't require routing, and avoid the subsequent latency and load.

#### 5.1.1.1. Create a Network

1. In the dashboard, select **`Project > Network > Networks`**.

2. Click **`+Create Network`**.

3. Specify the following:

| Field | Description |
|---|---|

| Field | Description |
|---|---|
| Network Name | Descriptive name, based on the role that the network will perform. If you are integrating the network with an external VLAN, consider appending the VLAN ID number to the name. Examples:<br>**webservers_122**, if you are hosting HTTP web servers in this subnet, and your VLAN tag is 122.<br>**internal-only**, if you intend to keep the network traffic private, and not integrate it with an external network. |
| Admin State | Controls whether the network is immediately available. This field allows you to create the network but still keep it in a **Down** state, where it is logically present but still inactive. This is useful if you do not intend to enter the network into production right away. |

4. Click the **Next** button, and specify the following in the **Subnet** tab:

| Field | Description |
|---|---|
| Create Subnet | Determines whether a subnet is created. For example, you might not want to create a subnet if you intend to keep this network as a placeholder without network connectivity. |
| Subnet Name | Descriptive name. |
| Network Address | Address in CIDR format, which contains the IP address range and subnet mask in one value. To determine the address, calculate the number of bits masked in the subnet mask and append that value to the IP address range. For example, the subnet mask 255.255.255.0 has 24 masked bits. To use this mask with the IPv4 address range 192.168.122.0, specify the address **192.168.122.0/24**. |
| IP Version | Internet protocol, where valid types are **IPv4** or **IPv6**. The IP address range in the **Network Address** field must match whichever version you select. |
| Gateway IP | IP address of the router interface for your default gateway. This address is the next hop for routing any traffic destined for an external location, and must be within the range specified in the **Network Address** field. For example, if your CIDR network address is 192.168.122.0/24, then your default gateway is likely to be 192.168.122.1. |
| Disable Gateway | Disables forwarding and keeps the network isolated. |

5. Click **Next** to specify DHCP options:

| Field | Description |
|---|---|
| Enable DHCP | Enables DHCP services for this subnet. DHCP allows you to automate the distribution of IP settings to your instances. |
| IPv6 Address Configuration Mode | If creating an IPv6 network, specifies how IPv6 addresses and additional information are allocated:<br>- **No Options Specified** - Select this option if IP addresses are set manually, or a non OpenStack-aware method is used for address allocation.<br>- **SLAAC (Stateless Address Autoconfiguration)** - Instances generate IPv6 addresses based on Router Advertisement (RA) messages sent from the OpenStack Networking router. This configuration results in an OpenStack Networking subnet created with **ra_mode** set to **slaac** and **address_mode** set to **slaac**.<br>- **DHCPv6 stateful** - Instances receive IPv6 addresses as well as additional options (for example, DNS) from OpenStack Networking DHCPv6 service. This configuration results in a subnet created with **ra_mode** set to **dhcpv6-stateful** and **address_mode** set to **dhcpv6-stateful**.<br>- **DHCPv6 stateless** - Instances generate IPv6 addresses based on Router Advertisement (RA) messages sent from the OpenStack Networking router. Additional options (for example, DNS) are allocated from the OpenStack Networking DHCPv6 service. This configuration results in a subnet created with **ra_mode** set to **dhcpv6-stateless** and **address_mode** set to **dhcpv6-stateless**. |
| Allocation Pools | Range of IP addresses you would like DHCP to assign. For example, the value **192.168.22.100,192.168.22.100** considers all 'up' addresses in that range as available for allocation. |
| DNS Name Servers | IP addresses of the DNS servers available on the network. DHCP distributes these addresses to the instances for name resolution. |
| Host Routes | Static host routes. First specify the destination network in CIDR format, followed by the next hop that should be used for routing. For example: 192.168.23.0/24, 10.1.31.1<br>Provide this value if you need to distribute static routes to instances. |

6. Click **Create**.

The completed network is available for viewing in the **Networks** tab. You can also click **Edit** to change any options as needed. Now when you create instances, you can configure them now to use its subnet, and they will subsequently receive any specified DHCP options.

#### 5.1.1.2. Create an Advanced Network

Advanced network options are available for administrators, when creating a network from the **Admin** view. These options define the network type to use, and allow tenants to be specified:

1. In the dashboard, select **Admin > Networks > Create Network > Project**.
2. Select a destination project to host the new network using **Project**.
3. Review the options in **Provider Network Type**:
   - **Local** - Traffic remains on the local Compute host and is effectively isolated from any external networks.
   - **Flat** - Traffic remains on a single network and can also be shared with the host. No VLAN tagging or other network segregation takes place.
   - **VLAN** - Create a network using a **VLAN ID** that corresponds to a VLAN present in the physical network. Allows instances to communicate with systems on the same layer 2 VLAN.
   - **GRE** - Use a network overlay that spans multiple nodes for private communication between instances. Traffic egressing the overlay must be routed.
   - **VXLAN** - Use a network overlay that spans multiple nodes for private communication between instances. Traffic egressing the overlay must be routed.
4. Click **Create Network**, and review the Project's **Network Topology** to validate that the network has been successfully created.

#### 5.1.1.3. Add Network Routing

To allow traffic to be routed to and from your new network, you must add its subnet as an interface to an existing virtual router:

1. In the dashboard, select **Project > Network > Routers**.
2. Click on your virtual router's name in the **Routers** list, and click **+Add Interface**.
3. In the **Subnet** list, select the name of your new subnet.
4. You can optionally specify an **IP address** for the interface in this field.

5. Click **Add Interface**.

Instances on your network are now able to communicate with systems outside the subnet.

### 5.1.2. Delete a Network

There are occasions where it becomes necessary to delete a network that was previously created, perhaps as housekeeping or as part of a decommisioning process. In order to successfully delete a network, you must first remove or detach any interfaces where it is still in use. The following procedure provides the steps for deleting a network in your project, together with any dependent interfaces.

1. In the dashboard, select **Project > Network > Networks**.
2. Remove all router interfaces associated with the target network's subnets. To remove an interface:
    a. Find the ID number of the network you would like to delete by clicking on your target network in the **Networks** list, and looking at the its ID field. All the network's associated subnets will share this value in their **Network ID** field.
    b. Select **Project > Network > Routers**, click on your virtual router's name in the **Routers** list, and locate the interface attached to the subnet you would like to delete. You can distinguish it from the others by the IP address that would have served as the gateway IP. In addition, you can further validate the distinction by ensuring that the interface's network ID matches the ID you noted in the previous step.
    c. Click the interface's **Delete Interface** button.
3. Select **Project > Network > Networks**, and click the name of your network. Click the target subnet's **Delete Subnet** button.

    **Note**

    If you are still unable to remove the subnet at this point, ensure it is not already being used by any instances.

4. Select **Project > Network > Networks**, and select the network you would like to delete.
5. Click **Delete Networks** in this and the next dialog screen.

### 5.1.3. Create a Subnet

Subnets are the means by which instances are granted network connectivity. Each instance is assigned to a subnet as part of the instance creation process, therefore it's important to consider proper placement of instances to best accomodate their connectivity requirements. Subnets are created in pre-existing networks.

Remember that tenant networks in OpenStack Networking can host multiple subnets. This is useful if you intend to host distinctly different systems in the same network, and would prefer a measure of isolation between them. For example, you can designate that only webserver traffic is present on one subnet, while database traffic traverse another. Subnets are isolated from each other, and any instance that wishes to communicate with another subnet must have their traffic directed by a router. Consider placing systems that will require a high volume of traffic amongst themselves in the same subnet, so that they don't require routing, and avoid the subsequent latency and load.

**Procedure 5.1. Create a new subnet**

1. In the dashboard, select **Project > Network > Networks**, and click your network's name in the **Networks** view.

2. Click **Create Subnet**, and specify the following.

| Field | Description |
|---|---|
| Subnet Name | Descriptive subnet name. |
| Network Address | Address in CIDR format, which contains the IP address range and subnet mask in one value. To determine the address, calculate the number of bits masked in the subnet mask and append that value to the IP address range. For example, the subnet mask 255.255.255.0 has 24 masked bits. To use this mask with the IPv4 address range 192.168.122.0, specify the address **192.168.122.0/24**. |
| IP Version | Internet protocol, where valid types are **IPv4** or **IPv6**. The IP address range in the **Network Address** field must match whichever version you select. |
| Gateway IP | IP address of the router interface for your default gateway. This address is the next hop for routing any traffic destined for an external location, and must be within the range specified in the **Network Address** field. For example, if your CIDR network address is 192.168.122.0/24, then your default gateway is likely to be 192.168.122.1. |
| Disable Gateway | Disables forwarding and keeps the network isolated. |

3. Click **Next** to specify DHCP options:

| Field | Description |
|---|---|
| Enable DHCP | Enables DHCP services for this subnet. DHCP allows you to automate the distribution of IP settings to your instances. |
| IPv6 Address Configuration Mode | If creating an IPv6 network, specifies how IPv6 addresses and additional information are allocated:<br>- **No Options Specified** - Select this option if IP addresses are set manually, or a non OpenStack-aware method is used for address allocation.<br>- **SLAAC (Stateless Address Autoconfiguration)** - Instances generate IPv6 addresses based on Router Advertisement (RA) messages sent from the OpenStack Networking router. This configuration results in a OpenStack Networking subnet created with **ra_mode** set to **slaac** and **address_mode** set to **slaac**.<br>- **DHCPv6 stateful** - Instances receive IPv6 addresses as well as additional options (for example, DNS) from OpenStack Networking DHCPv6 service. This configuration results in a subnet created with **ra_mode** set to **dhcpv6-stateful** and **address_mode** set to **dhcpv6-stateful**.<br>- **DHCPv6 stateless** - Instances generate IPv6 addresses based on Router Advertisement (RA) messages sent from the OpenStack Networking router. Additional options (for example, DNS) are allocated from the OpenStack Networking DHCPv6 service. This configuration results in a subnet created with **ra_mode** set to **dhcpv6-stateless** and **address_mode** set to **dhcpv6-stateless**. |
| Allocation Pools | Range of IP addresses you would like DHCP to assign. For example, the value **192.168.22.100,192.168.22.100** considers all 'up' addresses in that range as available for allocation. |
| DNS Name Servers | IP addresses of the DNS servers available on the network. DHCP distributes these addresses to the instances for name resolution. |
| Host Routes | Static host routes. First specify the destination network in CIDR format, followed by the next hop that should be used for routing. For example: 192.168.23.0/24, 10.1.31.1<br>Provide this value if you need to distribute static routes to instances. |

4. Click **Create**.

The new subnet is available for viewing in your network's **Subnets** list. You can also click **Edit** to change any options as needed. When you create instances, you can configure them now to use this subnet, and they will subsequently receive any specified DHCP options.

### 5.1.4. Delete a Subnet

You can delete a subnet if it is no longer in use. However, if any instances are still configured to use the subnet, the deletion attempt fails and the dashboard displays an error message.

This procedure demonstrates how to delete a specific subnet in a network:

1. In the dashboard, select **`Project > Network > Networks`**, and click the name of your network.

2. Select the target subnet and click **`Delete Subnets`**.

### 5.1.5. Add a router

OpenStack Networking provides routing services using an SDN-based virtual router. Routers are a requirement for your instances to communicate with external subnets, including those out in the physical network. Routers and subnets connect using **`interfaces`**, with each subnet requiring its own interface to the router.

A router's default gateway defines the next hop for any traffic received by the router. Its network is typically configured to route traffic to the external physical network using a virtual bridge.

1. In the dashboard, select **`Project > Network > Routers`**, and click **`+Create Router`**.

2. Enter a descriptive name for the new router, and click **`Create router`**.

3. Click **`Set Gateway`** next to the new router's entry in the **`Routers`** list.

4. In the **`External Network`** list, specify the network that will receive traffic destined for an external location.

5. Click **`Set Gateway`**.

After adding a router, the next step is to configure any subnets you have created to send traffic using this router. You do this by creating interfaces between the subnet and the router (see Procedure 5.0, “”).

### 5.1.6. Delete a router

You can delete a router if it has no connected interfaces. This procedure describes the steps needed to first remove a router's interfaces, and then the router itself.

1. In the dashboard, select **`Project > Network > Routers`**, and click on the name of the router you would like to delete.

2. Select the interfaces of type **`Internal Interface`**.

3. Click **`Delete Interfaces`**.

4. From the **`Routers`** list, select the target router and click **`Delete Routers`**.

### 5.1.7. Add an interface

Interfaces allow you to interconnect routers with subnets. As a result, the router can direct any traffic that instances send to destinations outside of their intermediate subnet. This procedure adds a router interface and connects it to a subnet.

The procedure uses the Network Topology feature, which displays a graphical representation of all your virtual router and networks and enables you to perform network management tasks.

1. In the dashboard, select **`Project > Network > Network Topology`**.

2. Locate the router you wish to manage, hover your mouse over it, and click **`Add Interface`**.

3. Specify the **`Subnet`** to which you would like to connect the router.

4. You have the option of specifying an **`IP Address`**. The address is useful for testing and troubleshooting purposes, since a successful ping to this interface indicates that the traffic is routing as expected.

5. Click **`Add interface`**.

The **`Network Topology`** diagram automatically updates to reflect the new interface connection between the router and subnet.

### 5.1.8. Delete an interface

You can remove an interface to a subnet if you no longer require the router to direct its traffic. This procedure demonstrates the steps required for deleting an interface:

1. In the dashboard, select **`Project > Network > Routers`**.

2. Click on the name of the router that hosts the interface you would like to delete.

3. Select the interface (will be of type **`Internal Interface`**), and click **`Delete Interfaces`**.

## 5.2. CONFIGURE IP ADDRESSING

You can use procedures in this section to manage your IP address allocation in OpenStack Networking.

### 5.2.1. Create Floating IP Pools

Floating IP addresses allow you to direct ingress network traffic to your OpenStack instances. You begin by defining a pool of validly routable external IP addresses, which can then be dynamically assigned to an instance. OpenStack Networking then knows to route all incoming traffic destined for that floating IP to the instance to which it has been assigned.

**Note**

OpenStack Networking allocates floating IP addresses to all projects (tenants) from the same IP ranges/CIDRs. Meaning that every subnet of floating IPs is consumable by any and all projects. You can manage this behavior using quotas for specific projects. For example, you can set the default to **10** for **ProjectA** and **ProjectB**, while setting **ProjectC's** quota to **0**.

The Floating IP allocation pool is defined when you create an external subnet. If the subnet only hosts floating IP addresses, consider disabling DHCP allocation with the **enable_dhcp=False**option:

```
# neutron subnet-create --name SUBNET_NAME --enable_dhcp=False --
allocation_pool start=IP_ADDRESS,end=IP_ADDRESS --
gateway=IP_ADDRESS NETWORK_NAME CIDR
```

**Example 5.1.**

```
# neutron subnet-create --name public_subnet --enable_dhcp=False
--allocation_pool start=192.168.100.20,end=192.168.100.100 --
gateway=192.168.100.1 public 192.168.100.0/24
```

### 5.2.2. Assign a Specific Floating IP

You can assign a specific floating IP address to an instance using the **nova** command (or through the dashboard; see Section 3.1.2, “Update an Instance (Actions menu)”).

```
# nova add-floating-ip INSTANCE_NAME IP_ADDRESS
```

**Example 5.2.**

In this example, a floating IP address is allocated to an instance named **corp-vm-01**:

```
# nova add-floating-ip corp-vm-01 192.168.100.20
```

### 5.2.3. Assign a Random Floating IP

Floating IP addresses can be dynamically allocated to instances. You do not select a particular IP address, but instead request that OpenStack Networking allocates one from the pool.

1. Allocate a floating IP from the previously created pool:

```
# neutron floatingip-create public
+---------------------+-------------------------------------
+
| Field               | Value
|
+---------------------+-------------------------------------
+
| fixed_ip_address    |                                      |
| floating_ip_address | 192.168.100.20                       |
| floating_network_id | 7a03e6bc-234d-402b-9fb2-0af06c85a8a3 |
| id                  | 9d7e2603482d
|
| port_id             |
|
| router_id           |
|
| status              | ACTIVE
|
| tenant_id           | 9e67d44eab334f07bf82fa1b17d824b6     |
+---------------------+-------------------------------------
+
```

2. With the IP address allocated, you can assign it to a particular instance. Locate the ID of the port associated with your instance (this will match the fixed IP address allocated to the instance). This port ID is used in the following step to associate the instance's port ID with the floating IP address ID. You can further distinguish the correct port ID by ensuring the MAC address in the third column matches the one on the instance.

```
# neutron port-list
+--------+------+-------------+-----------------------------
--------------------------+
| id     | name | mac_address | fixed_ips
|
+--------+------+-------------+-----------------------------
--------------------------+
| ce8320 |      | 3e:37:09:4b | {"subnet_id": "361f27",
"ip_address": "192.168.100.2"} |
| d88926 |      | 3e:1d:ea:31 | {"subnet_id": "361f27",
"ip_address": "192.168.100.5"} |
| 8190ab |      | 3e:a3:3d:2f | {"subnet_id": "b74dbb",
"ip_address": "10.10.1.25"}|
+--------+------+-------------+-------------------------------
--------------------------+
```

3. Use the **neutron** command to associate the floating IP address with the desired port ID of an instance:

```
# neutron floatingip-associate 9d7e2603482d 8190ab
```

### 5.2.4. Create Multiple Floating IP Pools

OpenStack Networking supports one floating IP pool per L3 agent. Therefore, scaling out your L3 agents allows you to create additional floating IP pools.

**Note**

Ensure that **handle_internal_only_routers** in **/etc/neutron/neutron.conf** is configured to *True* for only one L3 agent in your environment. This option configures the L3 agent to manage only non-external routers.

## 5.3. BRIDGE THE PHYSICAL NETWORK

The procedure below enables you to bridge your virtual network to the physical network to enable connectivity to and from virtual instances. In this procedure, the example physical **eth0** interface is mapped to the **br-ex** bridge; the virtual bridge acts as the intermediary between the physical network and any virtual networks. As a result, all traffic traversing **eth0** uses the configured Open vSwitch to reach instances.

1. Map a physical NIC to the virtual Open vSwitch bridge:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=OVSPort
DEVICETYPE=ovs
OVS_BRIDGE=br-ex
ONBOOT=yes
```

2. Configure the virtual bridge with the IP address details that were previously allocated to eth0:

```
# vi /etc/sysconfig/network-scripts/ifcfg-br-ex
DEVICE=br-ex
DEVICETYPE=ovs
TYPE=OVSBridge
BOOTPROTO=static
IPADDR=192.168.120.10
```

```
NETMASK=255.255.255.0
GATEWAY=192.168.120.1
DNS1=192.168.120.1
ONBOOT=yes
```

Where **IPADDR**, **NETMASK** **GATEWAY**, and **DNS1** (name server) must be updated to match your network.

You can now assign floating IP addresses to instances and make them available to the physical network.

# CHAPTER 6. CLOUD RESOURCES

This chapter discusses how to configure stacks and monitor cloud resources in RHEL OpenStack Platform.

## 6.1. MANAGE STACKS

The Orchestration service provides a framework through which you can define an instance's resource parameters (for example, floating IPs, volumes, or security groups) and properties (for example, key pairs, image to be used, or flavor) using *Heat templates*. These templates use a human-readable syntax and can be defined in text files (thereby allowing users to check them into version control). Templates allow you to easily deploy and re-configure infrastructure for applications within the OpenStack cloud.

Instances deployed using Heat templates through the Orchestration service are known as *stacks*. The dashboard allows you to launch, delete, and update stacks from Heat templates. You can input a Heat template directly into the dashboard, or use text files from your local file system or HTTP URL.

### 6.1.1. Download Sample Heat Templates

Red Hat Enterprise Linux OpenStack Platform includes sample templates you can use to test and study Heat's core functionality. To use these templates, install the openstack-heat-templates package:

```
# yum install openstack-heat-templates
```

This package installs the sample Heat templates in **`/usr/share/openstack-heat-templates/software-config/example-templates`**.

### 6.1.2. Launch a Stack

1. In the dashboard, select **`Project > Orchestration > Stacks`**, and click **`Launch Stack`**.

2. Select an option from the **`Template Source`** list:

| Option | Description |
|---|---|
| File | Use a local template file on your system. Select your file by clicking **`Template File > Browse`**. |
| Direct Input | Enter your template directly into the dashboard using the **`Template Data`** field. |

| Option | Description |
|---|---|
| URL | Use an external HTTP URL for the template. Specify the template's URL in the **Template URL** field. |

**Note**

Red Hat Enterprise Linux OpenStack Platform includes sample templates. For more details, see Section 6.1.1, “Download Sample Heat Templates”.

3. Select an option from the **Environment Source** list:

| Option | Description |
|---|---|
| File | Use a **.yaml** file for the environment. Select your environment by clicking **Environment File > Browse**. |
| Direct Input | Enter your environment data directly into the dashboard using the **Environment Data** field. |

4. Click **Next**.

5. Specify values for the following fields:

| Field | Description |
|---|---|
| Stack Name | Name to identify the stack. |
| Creation Timeout (minutes) | Number of minutes before declaring a timeout on the stack launch. |
| Rollback On Failure | If selected, rolls back any changes or updates to the template if the stack launch fails. |
| Password for user *USERNAME* | Temporary password for the user launching the stack. |

The **Launch Stack** window may also contain other fields, depending on the parameters defined in the template. Update these fields as required.

6. Click **Launch**.

6. Click **Launch**.

### 6.1.3. Update a Stack

1. If stack components need to be updated, edit your original template.

2. In the dashboard, select **Project > Orchestration > Stacks**.

3. Select the stack's **Change Stack Template** action.

4. Select an option from the **Template Source** list:

| Option | Description |
|---|---|
| File | Use a local template file on your system. Select your file by clicking **Template File > Browse**. |
| Direct Input | Enter your template directly into the dashboard using the **Template Data** field. |
| URL | Use an external HTTP URL for the template. Specify the template's URL in the **Template URL** field. |

5. Select an option from the **Environment Source** list:

| Option | Description |
|---|---|
| File | Use a **.yaml** file for the environment. Select your environment by clicking **Environment File > Browse**. |
| Direct Input | Enter your environment data directly into the dashboard using the **Environment Data** field. |

6. Click **Next**.

7. Specify values for the following fields:

| Field | Description |
|---|---|
| Creation Timeout (minutes) | Number of minutes before declaring a timeout on the stack launch. |
| Rollback On Failure | If selected, rolls back any changes or updates to the template if the stack launch fails. |
| Password for user *USERNAME* | Temporary password for the user launching the stack. |

The **Launch Stack** window may also contain other fields, depending on the parameters defined in the template. Update these fields as required.

8. Click **Update**. The Orchestration service re-launches the stack with the updated parameters. The **Updated** column on the **Stacks** table now reflects how long it has been since the stack was last updated.

### 6.1.4. Delete a Stack

You can delete a stack through the **Stacks** table:

1. In the dashboard, select **Project > Orchestration > Stacks**.

2. Select **Delete Stack** from the **Actions** column of a stack.

**Note**

Alternatively, you can delete multiple stacks simultaneously by selecting their respective checkboxes and clicking **Delete Stacks**.

## 6.2. USING THE TELEMETRY SERVICE

For help with the ceilometer command, use:

```
# ceilometer help
```

For help with the subcommands, use:

```
# ceilometer help subcommand
```

### 6.2.1. View Existing Alarms

To list configured Telemetry alarms, use:

```
# ceilometer alarm-list
```

To list configured meters for a resource, use:

```
# ceilometer meter-list --query resource=UUID
+---------------------------+------------+------------+------------+--
--------+----------+
| Name                      | Type       | Unit       | Resource   |
User ID  | Project  |
+---------------------------+------------+------------+------------+--
--------+----------+
| cpu                       | cumulative | ns         | 5056eda...|
b0e500...| f23524...|
| cpu_util                  | gauge      | %          | 5056eda...|
b0e500...| f23524...|
| disk.ephemeral.size       | gauge      | GB         | 5056eda...|
b0e500...| f23524...|
| disk.read.bytes           | cumulative | B          | 5056eda...|
b0e500...| f23524...|
                      output omitted
| instance                  | gauge      | instance   | 5056eda...|
b0e500...| f23524...|
| instance:m1.tiny          | gauge      | instance   | 5056eda...|
b0e500...| f23524...|
| memory                    | gauge      | MB         | 5056eda...|
b0e500...| f23524...|
| vcpus                     | gauge      | vcpu       | 5056eda...|
b0e500...| f23524...|
+---------------------------+------------+------------+--------------
--------------------+
```

Where *UUID* is the resource ID for an existing resource (for example, an instance, image, or volume).

### 6.2.2. Configure an Alarm

To configure an alarm to activate when a threshold value is crossed, use the **ceilometer alarm-threshold-create** command with the following syntax:

```
# ceilometer alarm-threshold-create --name alarm-name [--
description alarm-text] --meter-name meter-name --threshold value
```

**Example 6.1.**

To configure an alarm that activates when the average CPU utilization for an individual instance exceeds 50% for three consecutive 600s (10 minute) periods, use:

```
# ceilometer alarm-threshold-create --name cpu_high --description
'CPU usage high' --meter-name cpu_usage_high --threshold 50 --
comparison-operator gt --statistic avg --period 600 --evaluation-
```

```
periods 3 --alarm-action 'log://' --query resource_id=5056eda6-
8a24-4f52-9cc4-c3ddb6fb4a69
```

In this example, the notification action is a log message.

To edit an existing threshold alarm, use the **`ceilometer alarm-threshold-update`** command together with the alarm ID, followed by one or more options to be updated.

**Example 6.2.**

To increase the alarm threshold to 75%, use:

```
# ceilometer alarm-threshold-update 35addb25-d488-4a74-a038-
076aad3a3dc3 --threshold=75
```

### 6.2.3. Disable or Delete an Alarm

To disable an alarm, use:

```
# ceilometer alarm-threshold-update --enabled False ALARM_ID
```

To delete an alarm, use:

```
# ceilometer alarm-delete ALARM_ID
```

### 6.2.4. View Samples

To list all the samples for a particular meter name, use:

```
# ceilometer sample-list --meter METER_NAME
```

To list samples only for a particular resource within a range of time stamps, use:

```
# ceilometer sample-list --meter METER_NAME --query
'resource_id=INSTANCE_ID;timestamp>START_TIME;timestamp>=END_TIME'
```

Where *START_TIME* and *END_TIME* are in the form *iso-dateThh:mm:ss*.

**Example 6.3.**

To query an instance for samples taken between **`13:10:00`** and **`14:25:00`**, use:

```
#ceilometer sample-list --meter cpu --query
'resource_id=5056eda6-8a24-4f52-9cc4-c3ddb6fb4a69;timestamp>2015-
```

```
01-12T13:10:00;timestamp>=2015-01-12T14:25:00'
+--------------------+------+------------+---------------+------+-
--------------------+
| Resource ID        | Name | Type       | Volume        | Unit |
Timestamp           |
+--------------------+------+------------+---------------+------+-
--------------------+
| 5056eda6-8a24-... | cpu  | cumulative | 3.5569e+11    | ns   |
2015-01-12T14:21:44 |
| 5056eda6-8a24-... | cpu  | cumulative | 3.0041e+11    | ns   |
2015-01-12T14:11:45 |
| 5056eda6-8a24-... | cpu  | cumulative | 2.4811e+11    | ns   |
2015-01-12T14:01:54 |
| 5056eda6-8a24-... | cpu  | cumulative | 1.3743e+11    | ns   |
2015-01-12T13:30:54 |
| 5056eda6-8a24-... | cpu  | cumulative | 84710000000.0 | ns   |
2015-01-12T13:20:54 |
| 5056eda6-8a24-... | cpu  | cumulative | 31170000000.0 | ns   |
2015-01-12T13:10:54 |
+--------------------+------+------------+---------------+------+-
--------------------+
```

### 6.2.5. Create a Sample

Samples can be created for sending to the Telemetry service and they need not correspond to a previously defined meter. Use the following syntax:

```
# ceilometer sample-create --resource_id RESOURCE_ID --meter-name
METER_NAME --meter-type METER_TYPE --meter-unit METER_UNIT --
sample-volume SAMPLE_VOLUME
```

Where *METER_TYPE* can be one of:

- Cumulative - a running total
- Delta - a change or difference over time
- Gauge- a discrete value

**Example 6.4.**

```
# ceilometer sample-create -r 5056eda6-8a24-4f52-9cc4-
c3ddb6fb4a69 -m On_Time_Mins --meter-type cumulative --meter-unit
mins --sample-volume 0
+-------------------+-------------------------------------------
+
| Property          | Value
|
+-------------------+-------------------------------------------
+
| message_id        | 521f138a-9a84-11e4-8058-525400ee874f        |
| name              | On_Time_Mins
|
| project_id        | f2352499957d4760a00cebd26c910c0f
|
| resource_id       | 5056eda6-8a24-4f52-9cc4-c3ddb6fb4a69
|
| resource_metadata | {}
|
| source            | f2352499957d4760a00cebd26c910c0f:openstack |
| timestamp         | 2015-01-12T17:56:23.179729                  |
| type              | cumulative
|
| unit              | mins
|
| user_id           | b0e5000684a142bd89c4af54381d3722
|
| volume            | 0.0
|
+-------------------+---------------------------------------------
+
```

Where **volume**, normally the value obtained as a result of the sampling action, is in this case the value being created by the command.

**Note**

Samples are not updated because the moment a sample is created, it is sent to the Telemetry service. Samples are essentially messages, which is why they have a message ID. To create new samples, repeat the **sample-create** command and update the **--sample-volume** value.

### 6.2.6. View Cloud Usage Statistics

OpenStack administrators can use the dashboard to view cloud statistics.

1. As an admin user in the dashboard, select **Admin > System > Resource Usage**.

2. Click one of the following:

   - **Daily Report** — View a report of daily usage per project. Select the date range and a limit for the number of projects, and click **Generate Report**; the daily usage report is displayed.

   - **Stats** — View a graph of metrics grouped by project. Select the values and time period using the drop-down menus; the displayed graph is automatically updated.

The **ceilometer** command line client can also be used for viewing cloud usage statics.

**Example 6.5.**

To view all the statistics for the **`cpu_util`** meter, use:

```
# ceilometer statistics --meter cpu_util
+--------+----------------+--------------+-----+-----+------+---
----+------+--------
| Period | Period Start   |Period End     | Max | Min | Avg  |
Sum   | Count| Dura...
+--------+----------------+--------------+-----+-----+------+---
----+------+--------
| 0      | 2015-01-09T14: |2015-01-09T14:2| 9.44| 0.0 | 6.75 |
337.94| 50   | 2792...
+--------+----------------+--------------+-----+-----+------+---
----+------+--------
```

**Example 6.6.**

Statistics can be restricted to a specific resource by means of the **`--query`** option, and restricted to a specific range by means of the **`timestamp`** option.

```
# ceilometer statistics --meter cpu_util --query
'resource_id=5056eda6-8a24-4f52-9cc4-c3ddb6fb4a69;timestamp>2015-
01-12T13:00:00;timestamp<=2015-01-13T14:00:00'
+--------+----------------+--------------+-----+-----+------+---
----+------+--------
| Period | Period Start   |Period End     | Max | Min | Avg  |
Sum   | Count| Dura...
+--------+----------------+--------------+-----+-----+------+---
----+------+--------
| 0      | 2015-01-12T20:1|2015-01-12T20:1| 9.44| 5.95| 8.90 |
347.10| 39   | 2465...
+--------+----------------+--------------+-----+-----+------+---
----+------+--------
```

# CHAPTER 7. TROUBLESHOOTING

This chapter contains logging and support information to assist with troubleshooting your RHEL OpenStack Platform deployment.

## 7.1. LOGGING

RHEL OpenStack Platform writes informational messages to specific log files; you can use these messages for troubleshooting and monitoring system events.

### 7.1.1. Log Files for OpenStack Services

Each OpenStack component has a separate logging directory containing files specific to a running service.

**Table 7.1. Block Storage (cinder) log files**

| Service | Service Name | Log Path |
|---|---|---|
| Block Storage API | openstack-cinder-api.service | /var/log/cinder/api.log |
| Block Storage Backup | openstack-cinder-backup.service | /var/log/cinder/backup.log |
| Informational messages | The cinder-manage command | /var/log/cinder/cinder-manage.log |
| Block Storage Scheduler | openstack-cinder-scheduler.service | /var/log/cinder/scheduler.log |
| Block Storage Volume | openstack-cinder-volume.service | /var/log/cinder/volume.log |

**Table 7.2. Compute (nova) log files**

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack Compute API service | openstack-nova-api.service | /var/log/nova/nova-api.log |

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack Compute certificate server | openstack-nova-cert.service | /var/log/nova/nova-cert.log |
| OpenStack Compute service | openstack-nova-compute.service | /var/log/nova/nova-compute.log |
| OpenStack Compute Conductor service | openstack-nova-conductor.service | /var/log/nova/nova-conductor.log |
| OpenStack Compute VNC console authentication server | openstack-nova-consoleauth.service | /var/log/nova/nova-consoleauth.log |
| Informational messages | **nova-manage** command | /var/log/nova/nova-manage.log |
| OpenStack Compute NoVNC Proxy service | openstack-nova-novncproxy.service | /var/log/nova/nova-novncproxy.log |
| OpenStack Compute Scheduler service | openstack-nova-scheduler.service | /var/log/nova/nova-scheduler.log |

**Table 7.3. Dashboard (horizon) log files**

| Service | Service Name | Log Path |
|---|---|---|
| Log of certain user interactions | Dashboard interface | /var/log/horizon/horizon.log |

**Table 7.4. Identity Service (keystone) log files**

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack Identity Service | openstack-keystone.service | /var/log/keystone/keystone.log |

**Table 7.5. Image Service (glance) log files**

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack Image Service API server | openstack-glance-api.service | /var/log/glance/api.log |
| OpenStack Image Service Registry server | openstack-glance-registry.service | /var/log/glance/registry.log |

### Table 7.6. OpenStack Networking (neutron) log files

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack Networking Layer 3 Agent | neutron-l3-agent.service | /var/log/neutron/l3-agent.log |
| Open vSwitch agent | neutron-openvswitch-agent.service | /var/log/neutron/openvswitch-agent.log |
| Metadata agent service | neutron-metadata-agent.service | /var/log/neutron/metadata-agent.log |
| OpenStack Networking service | neutron-server.service | /var/log/neutron/server.log |

### Table 7.7. Telemetry (ceilometer) log files

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack ceilometer notification agent | openstack-ceilometer-notification.service | /var/log/ceilometer/agent-notification.log |
| OpenStack ceilometer alarm evaluation | openstack-ceilometer-alarm-evaluator.service | /var/log/ceilometer/alarm-evaluator.log |
| OpenStack ceilometer alarm notification | openstack-ceilometer-alarm-notifier.service | /var/log/ceilometer/alarm-notifier.log |

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack ceilometer API | openstack-ceilometer-api.service | /var/log/ceilometer/api.log |
| Informational messages | MongoDB integration | /var/log/ceilometer/ceilometer-dbsync.log |
| OpenStack ceilometer central agent | openstack-ceilometer-central.service | /var/log/ceilometer/central.log |
| OpenStack ceilometer collection | openstack-ceilometer-collector.service | /var/log/ceilometer/collector.log |
| OpenStack ceilometer compute agent | openstack-ceilometer-compute.service | /var/log/ceilometer/compute.log |

**Table 7.8. Orchestration (heat) log files**

| Service | Service Name | Log Path |
|---|---|---|
| OpenStack Heat API Service | openstack-heat-api.service | /var/log/heat/heat-api.log |
| Openstack Heat Engine Service | openstack-heat-engine.service | /var/log/heat/heat-engine.log |
| Orchestration service events | n/a | /var/log/heat/heat-manage.log |

### 7.1.2. Configure logging options

Each component maintains its own separate logging configuration in its respective configuration file. For example, in Compute, these options are set in **`/etc/nova/nova.conf`**:

- Increase the level of informational logging by enabling debugging. This option greatly increases the amount of information captured, so you may want to consider using it only temporarily, or first reviewing your log rotation settings.

```
debug=True
```

- Enable verbose logging:

```
verbose=True
```

- Change the log file path:

```
log_dir=/var/log/nova
```

- Send your logs to a central syslog server:

```
use_syslog=True
syslog_log_facility=LOG_USER
```

**Note**

Options are also available for timestamp configuration and log formatting, among others. Review the component's configuration file for additional logging options.

## 7.2. SUPPORT

If client commands fail or you run into other issues, please contact Red Hat Technical Support with a description of what happened, an **sosreport**, the full console output, and all log files referenced in the console output.

For information about the **sosreport** command (**sos** package), refer to What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later.

# APPENDIX A. IMAGE CONFIGURATION PARAMETERS

The following keys can be used with the **`property`** option for both the **`glance image-update`** and **`glance image-create`** commands.

**Example A.1.**

```
$ glance image-update IMG-UUID --property architecture=x86_64
```

**Note**

Behavior set using image properties overrides behavior set using flavors. For more information, see Section 3.3, “Manage Flavors”.

**Table A.1. Property keys**

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| All | architecture | The CPU architecture that must be supported by the hypervisor. For example, **`x86_64`**, **`arm`**, or **`ppc64`**. Run **`uname -m`** to get the architecture of a machine. We strongly recommend using the architecture data vocabulary defined by the libosinfo project for this purpose. | **`alpha`**-DEC 64-bit RISC<br>**`armv7l`**-ARM Cortex-A7 MPCore<br>**`cris`**-Ethernet, Token Ring, AXis-Code Reduced Instruction Set<br>**`i686`**-Intel sixth-generation x86 (P6 micro architecture)<br>**`ia64`**-Itanium<br>**`lm32`**-Lattice Micro32<br>**`m68k`**-Motorola 68000<br>**`microblaze`**-Xilinx 32-bit FPGA (Big Endian)<br>**`microblazeel`**-Xilinx 32-bit FPGA (Little Endian)<br>**`mips`**-MIPS 32-bit RISC (Big Endian) |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| | | | **mipsel**-MIPS 32-bit RISC (Little Endian)<br>**mips64**-MIPS 64-bit RISC (Big Endian)<br>**mips64el**-MIPS 64-bit RISC (Little Endian)<br>**openrisc**-OpenCores RISC<br>**parisc**-HP Precision Architecture RISC<br>**parisc64**-HP Precision Architecture 64-bit RISC<br>**ppc**-PowerPC 32-bit<br>**ppc64**-PowerPC 64-bit<br>**ppcemb**-PowerPC (Embedded 32-bit)<br>**s390**-IBM Enterprise Systems Architecture/390<br>**s390x**-S/390 64-bit<br>**sh4**-SuperH SH-4 (Little Endian)<br>**sh4eb**-SuperH SH-4 (Big Endian)<br>**sparc**-Scalable Processor Architecture, 32-bit<br>**sparc64**-Scalable Processor Architecture, 64-bit<br>**unicore32**-Microprocessor Research and Development Center RISC Unicore32<br>**x86_64**-64-bit extension of IA-32<br>**xtensa**-Tensilica Xtensa configurable microprocessor core |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| | | | **xtensaeb**-Tensilica Xtensa configurable microprocessor core (Big Endian)<br>**Note**<br>The **architecture** options fully supported by Red Hat are **i686** and **x86_64**. |
| All | hypervisor_type | The hypervisor type. | **kvm**, **vmware** |
| All | instance_uuid | For snapshot images, this is the UUID of the server used to create this image. | Valid server UUID |
| All | kernel_id | The ID of an image stored in the Image Service that should be used as the kernel when booting an AMI-style image. | Valid image ID |
| All | os_distro | The common name of the operating system distribution in lowercase (uses the same data vocabulary as the libosinfo project). Specify only a recognized value for this field. Deprecated values are listed to assist you in searching for the recognized value. | **arch**-Arch Linux. Do not use **archlinux** or **org.archlinux**<br>**centos**-Community Enterprise Operating System. Do not use **org.centos** or **CentOS**<br>**debian**-Debian. Do not use **Debian** or **org.debian**<br>**fedora**-Fedora. Do not use **Fedora**, **org.fedora**, or **org.fedoraproject**<br>**freebsd**-FreeBSD. Do not use **org.freebsd**, **freeBSD**, or **FreeBSD**<br>**gentoo**-Gentoo Linux. Do not use **Gentoo** or **org.gentoo**<br>**mandrake**-Mandrakelinux (MandrakeSoft) distribution. Do not use **mandrakelinux** or **MandrakeLinux** |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| | | | - **mandriva**-Mandriva Linux. Do not use **mandrivalinux**<br>- **mes**-Mandriva Enterprise Server. Do not use **mandrivaent** or **mandrivaES**<br>- **msdos**-Microsoft Disc Operating System. Do not use **ms-dos**<br>- **netbsd**-NetBSD. Do not use **NetBSD** or **org.netbsd**<br>- **netware**-Novell NetWare. Do not use **novell** or **NetWare**<br>- **openbsd**-OpenBSD. Do not use **OpenBSD** or **org.openbsd**<br>- **opensolaris**-OpenSolaris. Do not use **OpenSolaris** or **org.opensolaris**<br>- **opensuse**-openSUSE. Do not use **suse**, **SuSE**, or **org.opensuse**<br>- **rhel**-Red Hat Enterprise Linux. Do not use **redhat**, **RedHat**, or **com.redhat**<br>- **sled**-SUSE Linux Enterprise Desktop. Do not use **com.suse**<br>- **ubuntu**-Ubuntu. Do not use **Ubuntu**, **com.ubuntu**, **org.ubuntu**, or **canonical**<br>- **windows**-Microsoft Windows. Do not use **com.microsoft.server** |
| All | os_version | The operating system version as specified by the distributor. | Version number (for example, "11.10") |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| All | ramdisk_id | The ID of image stored in the Image Service that should be used as the ramdisk when booting an AMI-style image. | Valid image ID |
| All | vm_mode | The virtual machine mode. This represents the host/guest ABI (application binary interface) used for the virtual machine. | **hvm**-Fully virtualized. This is the mode used by QEMU and KVM. |
| libvirt API driver | hw_disk_bus | Specifies the type of disk controller to attach disk devices to. | **scsi**, **virtio**, **ide**, or **usb**. |
| libvirt API driver | hw_numa_nodes | Number of NUMA nodes to expose to the instance (does not override flavor definition). | Integer. For a detailed example of NUMA-topology definition, refer to the **hw:NUMA_def** key in Section 3.3.4.2, “Add Metadata”. |
| libvirt API driver | hw_numa_mempolicy | NUMA memory allocation policy (does not override flavor definition). | - strict - Mandatory for the instance's RAM allocations to come from the NUMA nodes to which it is bound (default if numa_nodes is specified).<br>- preferred - The kernel can fall back to using an alternative node. Useful when the 'hw:numa_nodes' parameter is set to '1'. |
| libvirt API driver | hw_numa_cpus.0 | Mapping of vCPUs N-M to NUMA node 0 (does not override flavor definition). | Comma-separated list of integers. |
| libvirt API driver | hw_numa_cpus.1 | Mapping of vCPUs N-M to NUMA node 1 (does not override flavor definition). | Comma-separated list of integers. |
| libvirt API driver | hw_numa_mem.0 | Mapping N GB of RAM to NUMA node 0 (does not override flavor definition). | Integer |
| libvirt API driver | hw_numa_mem.1 | Mapping N GB of RAM to NUMA node 1 (does not override flavor definition). | Integer |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| libvirt API driver | hw_rng_model | Adds a random-number generator device to the image's instances. The cloud administrator can enable and control device behavior by configuring the instance's flavor. By default:<br>- The generator device is disabled.<br>- **/dev/random** is used as the default entropy source. To specify a physical HW RNG device, use the following option in the **nova.conf** file:<br>`rng_dev_path=/dev/hwrng` | **virtio**, or other supported device. |
| libvirt API driver | hw_scsi_model | Enables the use of VirtIO SCSI (virtio-scsi) to provide block device access for compute instances; by default, instances use VirtIO Block (virtio-blk). VirtIO SCSI is a para-virtualized SCSI controller device that provides improved scalability and performance, and supports advanced SCSI hardware. | **virtio-scsi** |
| libvirt API driver | hw_video_model | The video image driver used. | **vga**, **cirrus**, **vmvga**, **xen**, or **qxl** |
| libvirt API driver | hw_video_ram | Maximum RAM for the video image. Used only if a **hw_video:ram_max_mb** value has been set in the flavor's **extra_specs** and that value is higher than the value set in **hw_video_ram**. | Integer in MB (for example, '64') |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| libvirt API driver | hw_watchdog_action | Enables a virtual hardware watchdog device that carries out the specified action if the server hangs. The watchdog uses the i6300esb device (emulating a PCI Intel 6300ESB). If **hw_watchdog_action** is not specified, the watchdog is disabled. | - **disabled**-The device is not attached. Allows the user to disable the watchdog for the image, even if it has been enabled using the image's flavor. The default value for this parameter is **disabled**.<br>- **reset**-Forcefully reset the guest.<br>- **poweroff**-Forcefully power off the guest.<br>- **pause**-Pause the guest.<br>- **none**-Only enable the watchdog; do nothing if the server hangs. |
| libvirt API driver | os_command_line | The kernel command line to be used by the libvirt driver, instead of the default. For Linux Containers (LXC), the value is used as arguments for initialization. This key is valid only for Amazon kernel, ramdisk, or machine images (aki, ari, or ami). | |
| libvirt API driver and VMware API driver | hw_vif_model | Specifies the model of virtual network interface device to use. | The valid options depend on the configured hypervisor.<br>- KVM and QEMU: **e1000**, **ne2k_pci**, **pcnet**, **rtl8139**, and **virtio**.<br>- VMware: **e1000**, **e1000e**, **VirtualE1000**, **VirtualE1000e**, **VirtualPCNet32**, **VirtualSriovEthernetCard**, and **VirtualVmxnet**.<br>- Xen: **e1000**, **netfront**, **ne2k_pci**, **pcnet**, and **rtl8139**. |

| Specific to | Key | Description | Supported values |
|---|---|---|---|
| VMware API driver | vmware_adaptertype | The virtual SCSI or IDE controller used by the hypervisor. | **lsiLogic**, **busLogic**, or **ide** |
| VMware API driver | vmware_ostype | A VMware GuestID which describes the operating system installed in the image. This value is passed to the hypervisor when creating a virtual machine. If not specified, the key defaults to **otherGuest**. | See thinkvirt.com. |
| VMware API driver | vmware_image_version | Currently unused. | **1** |
| XenAPI driver | auto_disk_config | If true, the root partition on the disk is automatically resized before the instance boots. This value is only taken into account by the Compute service when using a Xen-based hypervisor with the XenAPI driver. The Compute service will only attempt to resize if there is a single partition on the image, and only if the partition is in **ext3** or **ext4** format. | **true \| false** |
| XenAPI driver | os_type | The operating system installed on the image. The XenAPI driver contains logic that takes different actions depending on the value of the **os_type** parameter of the image. For example, for **os_type=windows** images, it creates a FAT32-based swap partition instead of a Linux swap partition, and it limits the injected host name to less than 16 characters. | **linux** or **windows** |

# APPENDIX B. REVISION HISTORY

| Revision | Date | Author |
|---|---|---|
| **Revision 6.0.2-4** | **Tue Jun 01 2015** | **Don Domingo** |
| BZ#1227123 - Added instructions on how to delete snapshots, along with cross-references to related Ceph instructions to protect/unprotect a snapshot in the back end. | | |
| **Revision 6.0.2-3** | **Fri May 15 2015** | **Summer Long** |
| BZ#1206395 - Added section 3.1.5.2. Directly Connect to the VNC Console.<br>BZ#1194113 - Clarified supported volume encryption settings.<br>BZ#1182406 - Added section 3.1.5.3. Directly Connect to a Serial Console. | | |
| **Revision 6.0.2-2** | **Thu Apr 23 2015** | **Summer Long** |
| BZ#1182817 - Updated scheduling filters, flavor metadata, and image metadata for scheduling instances using NUMA topology definitions. | | |
| **Revision 6.0.2-1** | **Thu Apr 16 2015** | **Summer Long** |
| BZ#1190560 - Added Actions table to 3.1.2. Update an Instance. | | |
| **Revision 6.0.2-0** | **Thu Apr 9 2015** | **Don Domingo** |
| BZ#1194116 - Clarified that readers need to consult driver documentation for valid *Extra Specs* key/value pairs. Also added links to sample procedures where volume types and extra specs are used. | | |
| **Revision 6.0.1-3** | **Tue Apr 7 2015** | **Summer Long** |
| BZ#1209330 - disk_allocation_ratio and AggregateDiskFilter descriptions updated, plus minor edits for clarification. | | |
| **Revision 6.0.1-2** | **Thu Mar 19 2015** | **Martin Lopes** |
| BZ#1163726 - Added note describing Floating IP allocation behavior when using multiple projects (mlopes). | | |
| **Revision 6.0.1-1** | **Tue Mar 17 2015** | **Summer Long** |
| BZ#1147794 - Updated SSH Tunneling section with explicit copying instructions (slong).<br>BZ#1194539 - Added information on sample templates (ddomingo).<br>BZ#1193749 - Updated Image and Storage chapter introduction (dnavale). | | |
| **Revision 6.0.1-0** | **Thu Mar 5 2015** | **Summer Long** |
| Finalized for maintenance release 6.0.1.<br>BZ#1191794 - Structural edits for entire guide. | | |
| **Revision 6.0.0-6** | **Wed Feb 18 2015** | **Don Domingo** |
| BZ#1194112 - Added mini-section on selecting a back end<br>BZ#1041696 - Added "Configure How Volumes are Allocated to Multiple Back Ends".<br>BZ#1190661 - Added "Upload a Volume to the Image Service". | | |
| **Revision 6.0.0-5** | **Thu Feb 12 2015** | **Summer Long** |
| BZ#1191776 - Removed bad table titles in Volume section. | | |
| **Revision 6.0.0-4** | **Thu Feb 5 2015** | **Summer Long** |
| Release for Red Hat Enterprise Linux OpenStack Platform 6.0. | | |