

OpenStack

End User Guide

master (September 3, 2013)



OpenStack End User Guide

master (2013-09-03)

Copyright © 2013 OpenStack Foundation Some rights reserved.

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



Except where otherwise noted, this document is licensed under
Creative Commons Attribution 3.0 License.
<http://creativecommons.org/licenses/by/3.0/legalcode>

Table of Contents

How can I use an OpenStack cloud?	4
Document change history	4
1. OpenStack dashboard	1
Log in to the dashboard	2
Upload images	5
Configure access and security for instances	7
Launch and manage instances	9
Manage volumes	13
2. OpenStack command-line clients	15
Overview	15
Install the Openstack command-line clients	16
Get the version for a client	18
Download and source the OpenStack RC file	19
Manage images	20
Configure access and security for instances	24
Launch instances	28
Manage instances and hosts	36
Create and manage networks	48
Create and manage stacks	52
Manage volumes	55
A. Command reference	60
keystone commands	60
glance commands	62
neutron commands	64
nova commands	67
cinder commands	71
swift commands	73
heat commands	74
Resources	77

How can I use an OpenStack cloud?

Document change history 4

As an OpenStack cloud end user, you can provision your own resources within the limits set by administrators.

The examples in this guide show you how to complete these tasks with either:

- The OpenStack dashboard. Use this Web-based graphical interface, code named [horizon](#), to view, create, and manage resources.
- The OpenStack command-line clients. Each core OpenStack project has a command-line client that lets you run simple commands to view, create, and manage resources in a cloud and automate tasks by using scripts.

You can modify these examples for your specific use cases.

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

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

Document change history

This version of the guide replaces and obsoletes all previous versions. The following table describes the most recent changes:

Revision Date	Summary of Changes
August 19, 2013	<ul style="list-style-type: none">• Editorial changes.
July 29, 2013	<ul style="list-style-type: none">• First edition of this document.

1. OpenStack dashboard

Log in to the dashboard	2
Upload images	5
Configure access and security for instances	7
Launch and manage instances	9
Manage volumes	13

As a cloud end user, you can use the OpenStack dashboard to provision your own resources within the limits set by administrators. You can modify these examples to create other types and sizes of server instances.

Log in to the dashboard

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

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



Note

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

3. In the address bar, type the host name or IP address that you got from the cloud operator:

```
https://IP_ADDRESS_OR_HOSTNAME/
```



Certificate Warning

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

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

The top-level row shows your user name. You can also access **Settings** or sign out of the dashboard.

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

If you are logged in as an end user, the main screen shows the [Project](#) tab.

If you are logged in as an administrator, the main screen shows the [Project](#) tab and [Admin](#) tab.

OpenStack dashboard—Project tab

Select a project from the drop-down list on the left side to view and perform tasks for resources created by users of that project. The **Project** tab displays the details of the projects to which you belong.

The screenshot shows the OpenStack dashboard 'Overview' page. The user is logged in as 'demo'. The page displays a 'Quota Summary' with three progress bars: 'Used 1 of 10 Available Instances', 'Used 1 of 20 Available vCPUs', and 'Used 2,048 MB of 51,200 MB Available RAM'. Below this is a 'Select a month to query its usage:' section with a dropdown menu set to 'June' and the year '2013'. A 'Submit' button is next to it. The usage statistics are: 'Active Instances: 1 Active RAM: 2GB This Month's VCPU-Hours: 235.96 This Month's GB-Hours: 4719.11'. A 'Usage Summary' table is shown with one row for 'myserver'.

Instance Name	VCPUs	Disk	RAM	Uptime
myserver	1	20	2GB	2 months, 3 weeks

Then, access the following categories to perform the following tasks:

- | | |
|-------------------------------|--|
| Overview | View reports for the project. |
| Instances | View, launch, create a snapshot from, stop, pause, or reboot instances, or connect to them through VNC. |
| Volumes | View, create, edit, and delete volumes. |
| Images & Snapshots | View images, instance snapshots, and volume snapshots created by users of the project, plus any images that are publicly available. Create, edit, and delete images, and launch instances from images and snapshots. |
| Access & Security | <ul style="list-style-type: none"> • Security Groups tab. View, create, edit, and delete security groups and security group rules. • Keypairs tab. View, create, edit, and import keypairs, and delete keypairs. • Floating IPs tab. Allocate an IP address to or release it from a project. • API Access tab. View API endpoints. |

OpenStack dashboard—Admin tab

Enables administrative users to view usage and manage instances, volumes, flavors, images, projects, users, services, and quotas.

The screenshot shows the OpenStack dashboard 'Overview' page for an administrative user 'admin'. The user is logged in as 'admin'. The page displays a 'Select a month to query its usage:' section with a dropdown menu set to 'August' and the year '2013'. A 'Submit' button is next to it. The usage statistics are: 'Active Instances: 1 Active RAM: 512MB This Month's VCPU-Hours: 48.89 This Month's GB-Hours: 0.00'. A 'Usage Summary' table is shown with one row for 'demo'.

Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours
demo	1	0	512MB	48.89	0.00

On the **Admin** tab, access the following categories to perform the following tasks:

Overview	View basic reports.
Instances	View, pause, resume, suspend, migrate, soft or hard reboot, and delete running instances that belong to users of some, but not all, projects. Also, view the log for an instance or access an instance through VNC.
Volumes	View, create, edit, and delete volumes and volume types.
Flavors	View, create, edit, view extra specs for, and delete flavors. A flavor is size for an instance.
Images	View, create, edit properties for, and delete custom images.
Projects	View, create, assign users to, remove users from, and delete projects.
Users	View, create, enable, disable, and delete users.
System Info	<ul style="list-style-type: none">• Services tab. View services.• Default Quotas tab. View default quota values. Quotas are hard-coded in OpenStack Compute and define the maximum allowable size and number of resources.

Upload images

The cloud operator assigns roles to users, which determines who can upload and manage images. Image upload and management might be restricted to only cloud administrators or cloud operators. If the administrator or cloud operator has granted you permission, you can upload and manage images. You can also use the glance and nova clients or the Image Service and Compute APIs to manage images. See [the section called “Manage images” \[20\]](#). For details about image creation, see the [Virtual Machine Image Guide](#).

1. Log in to the OpenStack dashboard, choose a project, and click the **Images & Snapshots** category.
2. Click **Create Image**. The **Create An Image** window appears:

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

Name	Enter a name for the image.
Description	Enter a brief description about the image.
Image Location	Include the URL of the image.
Image File	Alternatively, browse to find the file on your machine.
Format	Select the image format.
Minimum Disk (GB) and Maximum Disk (GB)	Leave these fields empty.
Public	Select this option to make the image public to all users.
Protected	Select this option to ensure only users with permissions can delete it.

4. Click **Create Image**.

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

Configure access and security for instances

Before you launch a virtual machine, you can add security group rules to enable users to ping and SSH to the instances. To do so, you either add rules to the default security group or add a security group with rules.

Keypairs are SSH credentials that are injected into images when they are launched. For this to work, the image must contain the `cloud-init` package. Create at least one keypair for each project. For information, see [the section called "Add a keypair" \[8\]](#).

If you have generated a keypair with an external tool, you can import it into OpenStack. The keypair can be used for multiple instances that belong to a project. For information, see [the section called "Import a keypair" \[8\]](#).

Add rules to the default security group

1. Log in to the OpenStack dashboard, choose a project, and click the **Security** category. The dashboard shows the security groups that are available for this project.
2. Select the default security group and click **Edit Rules**.
3. To add a TCP rule, click **Add Rule**.
4. In the **Add Rule** window, enter the following values:

IP Protocol	TCP
Open	Port
Port	22
Source	CIDR
CIDR	0.0.0.0/0



Note

To accept requests from a particular range of IP addresses, specify the IP address block in the **CIDR** box.

5. Click **Add**.
Port 22 is now open for requests from any IP address.
6. To add an ICMP rule, click **Add Rule**.
7. In the **Add Rule** window, enter the following values:

IP Protocol	ICMP
Type	-1
Code	-1
Source	CIDR
CIDR	0.0.0.0/0

8. Click **Add**.

Add a keypair

Create at least one keypair for each project.

1. Log in to the OpenStack dashboard, choose a project, and click the **Access & Security** category.
2. The **Keypairs** tab shows the keypairs that are available for this project.
3. Click **Create Keypair**.
4. In the **Create Keypair** window, enter a name for your keypair, and click **Create Keypair**.
5. Respond to the prompt to download the keypair.

Import a keypair

1. Log in to the OpenStack dashboard, choose a project, and click the **Access & Security** category.
2. The **Keypairs** tab shows the keypairs that are available for this project.
3. Click **Import Keypair**.
4. In the **Import Keypair** window, enter the name of your keypair. In the **Public Key** box, copy the public key. Then, click **Import Keypair**.
5. Save the *.pem file locally. To change its permissions so that only you can read and write to the file, run the following command:

```
$ chmod 0600 MY_PRIV_KEY.pem
```

6. To make the keypair known to SSH, run the **ssh-add** command:

```
$ ssh-add MY_PRIV_KEY.pem
```

The Compute database registers the public key of the keypair.

The dashboard lists the keypair in the **Access & Security** category.

Launch and manage instances

Instances are virtual machines that run inside the cloud.

You can [launch an instance from an OpenStack image](#). The OpenStack Image Service provides a pool of images that are accessible to members of different projects.

You can also [launch an instance from an image that you have copied to a persistent volume](#). The instance boots from the volume, which is provided by `nova-volume` through iSCSI. When you launch an instance from a volume, especially note the following steps:

- To select from which volume to boot, launch an instance from an arbitrary image. The image you select does not boot. It is replaced by the image on the volume that you choose in the next steps.

To boot a Xen image from a volume, the image you launch in must be the same type, fully virtualized or paravirtualized, as the one on the volume.

- Select the volume or volume snapshot from which to boot. Enter a device name. Enter `vda` for KVM images or `xvda` for Xen images.

Launch an instance from an image

When you launch an instance from an image, OpenStack creates a local copy of the image on the Compute node where the instance starts.

1. Log in to the OpenStack dashboard, choose a project, and click the **Images & Snapshot** category.

The dashboard shows the images that have been uploaded to OpenStack Image Service and are available for this project.

2. Select an image and click **Launch**.
3. In the **Launch Image** window, specify the following values:

Details tab	
Instance Source	Image or snapshot.
Instance Name	The name to assign to the virtual machine.
Flavor	The size of the virtual machine to launch.
Instance Count	To launch multiple instances, enter a value greater than 1. Default is 1.
Access & Security tab	
Keypair	A keypair. In case an image uses a static root password or a static key set (neither is recommended), you do not need to provide a keypair to launch the instance.
Security Groups	Activate the security groups that you want to assign to the instance. Security groups are a kind of cloud firewall that define which incoming network traffic is forwarded to instances. For details, see the section called "Add rules to the default security group" [7] . If you have not created any security groups, you can assign only the default security group to the instance.
Volume Options tab	
Volume Options	Not applicable when you launch an instance from an image. To launch from a volume or volume snapshot, select the appropriate option in the Volume Options drop-down list. Then, choose the volume or snapshot. For information about how to create a bootable volume, see the section called "Launch an instance from a volume" [11] .
Post-Creation tab	
Customization Script	A customization script that runs after your instance launches.

4. Click **Launch**. The instance starts on a Compute node in the cloud.
5. The **Instances** category shows the instance name, its private and public IP addresses, size, status, task, and power state.
6. If you did not provide a keypair, security groups, or rules so far, users can only access the instance from inside the cloud through VNC. Even pinging the instance is not possible. To access the instance through a VNC console, see [the section called "Get a console to an instance" \[42\]](#).

Launch an instance from a volume

You can launch an instance directly from an image that has been copied to a persistent volume.

1. Create a volume that is large enough to store an unzipped image.
2. Create an image.

For details, see [Creating images manually](#) in the *OpenStack Virtual Machine Image Guide*.

3. Launch an instance.
4. Attach the volume to the instance.
5. Assuming that the attached volume is mounted as `/dev/vdb`, use one of the following commands to copy the image to the attached volume:

- For a raw image:

```
$ cat IMAGE >/dev/vdb
```

Alternatively, use `dd`.

- For a non-raw image:

```
$ qemu-img convert -O raw IMAGE /dev/vdb
```

- For a `*.tar.bz2` image:

```
$ tar xfjO IMAGE >/dev/vdb
```

6. Because only *detached* volumes are available for booting, detach the volume.
7. Now, you can launch an instance from an image that has been copied to the volume. The instance is booted from the volume, which is provided by `nova-volume` through iSCSI. To launch an instance from the volume, see [the section called “Launch an instance from an image” \[10\]](#).

SSH in to your instance

To SSH into your instance, you use the downloaded keypair file.



Note

The username is `ubuntu` for the Ubuntu cloud images on TryStack.

1. Copy the IP address for your instance.
2. Use the SSH command to make a secure connection to the instance. For example:

```
$ ssh -i MyKey.pem ubuntu@10.0.0.2
```

3. At the prompt, type `yes`.

Track usage for instances

You can track usage for instances for each tenant, also known as a project. You can track costs per month by showing metrics like number of VCPUs, disks, RAM, and uptime for all your instances.

1. Log in to the OpenStack dashboard, choose a project, and click the **Overview** category.
2. To query the instance usage for a month, select a month and click **Submit**.
3. To download a summary, click **Download CSV Summary**.

Create instance snapshots

1. Log in to the OpenStack dashboard, choose a project, and click the **Instances** category.
2. Select the instance from which to create a snapshot. From the **Actions** drop-down list, select **Create Snapshot**.
3. In the **Create Snapshot** window, enter a name for the snapshot. Click **Create Snapshot**. The **Images & Snapshots** category shows the instance snapshot.
4. To launch an instance from the snapshot, select the snapshot and click **Launch**. Proceed with [the section called "Launch an instance from an image" \[10\]](#).

Control the state of an instance

1. Log in to the OpenStack dashboard, choose a project, and click the **Instances** category.
2. Select an instance.
3. In the **More** drop-down list in the **Actions** column, select the state.

Depending on the current state of the instance, you can choose to pause, resume, suspend, soft or hard reboot, or terminate an instance.

Manage volumes

Volumes are block storage devices that you can attach to instances. They allow for persistent storage. You can attach a volume to a running instance, or detach a volume and attach it to another instance at any time.

Create a volume

1. Log in to the OpenStack dashboard, choose a project, and, click the **Volumes** category.
2. Click **Create Volume**.

In the window that opens, enter a name to assign to a volume, a description (optional), and define the size in GBs.
3. Confirm your changes.
4. The dashboard shows the volume in the **Volumes** category.

Attach volumes to instances

After you create one or more volumes, you can attach them to instances.

1. Log in to the OpenStack dashboard, choose a project, and click the **Volumes** category.
2. Select the volume to add to an instance and click **Edit Attachments**.
3. In the **Manage Volume Attachments** window, select an instance.
4. Enter a device name under which the volume should be accessible on the virtual machine.
5. Click **Attach Volume** to confirm your changes. The dashboard shows the instance to which the volume has been attached and the volume's device name. You can attach a volume to one instance at a time.
6. View the status of a volume in the **Instances & Volumes** category of the dashboard. The volume is either available or In-Use.
7. Now you can log in to the instance, mount the disk, format it, and use it.

Detach a volume from an instance

1. Log in to the OpenStack dashboard, choose a project, and click the **Volumes** category.
2. Select the volume and click **Edit Attachments**.
3. Click **Detach Volume** and confirm your changes.
4. A message indicates whether the action was successful.

Delete volumes

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

1. Log in to the OpenStack dashboard, choose a project, and click the **Volumes** category.
2. Activate the check boxes in front of the volumes that you want to delete.
3. Click **Delete Volumes** and confirm your choice in the pop-up that appears.
4. A message indicates whether the action was successful.

2. OpenStack command-line clients

Overview	15
Install the Openstack command-line clients	16
Get the version for a client	18
Download and source the OpenStack RC file	19
Manage images	20
Configure access and security for instances	24
Launch instances	28
Manage instances and hosts	36
Create and manage networks	48
Create and manage stacks	52
Manage volumes	55

Overview

You can use the OpenStack command-line clients to run simple commands that make API calls and automate tasks by using scripts. Internally, each client command runs cURL commands that embed API requests. The OpenStack APIs are RESTful APIs that use the HTTP protocol, including methods, URIs, media types, and response codes.

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

The following command-line clients are available for the respective services' APIs:

- **cinder** (python-cinderclient). Client for the Block Storage Service API. Use to create and manage volumes.
- **glance** (python-glanceclient). Client for the Image Service API. Use to create and manage images.
- **keystone** (python-keystoneclient). Client for the Identity Service API. Use to create and manage users, tenants, roles, endpoints, and credentials.
- **nova** (python-novaclient). Client for the Compute API and its extensions. Use to create and manage images, instances, and flavors.
- **neutron** (python-neutronclient). Client for the Networking API. Use to configure networks for guest servers. This client was previously known as **quantum**.
- **swift** (python-swiftclient). Client for the Object Storage API. Use to gather statistics, list items, update metadata, upload, download and delete files stored by the object storage service. Provides access to a swift installation for ad hoc processing.
- **heat** (python-heatclient). Client for the Orchestration API. Use to launch stacks from templates, view details of running stacks including events and resources, and update and delete stacks.

An OpenStack common client is in development.

Install the Openstack command-line clients

To install the clients, install the prerequisite software and the Python package for each OpenStack client.

Table 2.1. Prerequisite software

Prerequisite	Description
Python 2.6 or later	Currently, the clients do not support Python 3.
setuptools package	Installed by default on Mac OS X. Many Linux distributions provide packages to make setuptools easy to install. Search your package manager for setuptools to find an installation package. If you cannot find one, download the setuptools package directly from http://pypi.python.org/pypi/setuptools .
pip package	<p>To install the clients on a Mac OS X or Linux system, use pip. It is easy to use and ensures that you get the latest version of the clients from the Python Package Index. Also, it lets you update or remove the packages later on.</p> <p>Install pip through the package manager for your system:</p> <ul style="list-style-type: none"> • Mac OS X <pre>\$ sudo easy_install pip</pre> • Ubuntu 12.04 <p>A packaged version enables you to use dpkg or aptitude to install the python-novaclient.</p> <pre># aptitude install python-novaclient</pre> • Ubuntu <pre># aptitude install python-pip</pre> • RHEL, CentOS, or Fedora <p>A packaged version available in RDO enables you to use yum to install the clients:</p> <pre># yum install python-PROJECTclient</pre> <p>Replace <i>PROJECT</i> with the lower case name of the client to install, such as <i>nova</i>. Repeat this step for each client.</p> <p>Alternatively install pip and use it to manage client installation:</p> <pre># yum install python-pip</pre> • openSUSE 12.2 and earlier <p>A packaged version available in the Open Build Service enables you to use rpm or zypper to install the python-novaclient.</p> <pre># zypper install python-PROJECT</pre> <p>Replace <i>PROJECT</i> with the lowercase name of the client to install, such as <i>nova</i>. Repeat this step for each desired client.</p> <p>Alternatively install pip and use it to manage client installation:</p> <pre># zypper install python-pip</pre> • openSUSE 12.3 <p>A packaged version enables you to use rpm or zypper to install the clients:</p> <pre># zypper install python-PROJECTclient</pre> <p>Replace <i>PROJECT</i> with the lowercase name of the client to install, such as <i>nova</i>. Repeat this step for each desired client.</p>

Install the clients

Use **pip** to install the OpenStack clients on a Mac OS X or Linux system. It is easy and ensures that you get the latest version of the client from the [Python Package Index](#). Also, **pip** lets you update or remove a package. After you install the clients, you must source an `openrc` file to set required environment variables before you can request OpenStack services through the clients or the APIs.

1. You must install each client separately.

Run the following command to install or update a client package:

```
$ sudo pip install [--update] python-PROJECTclient
```

Where *PROJECT* is the project name and has one of the following values:

- `nova`. Compute API and extensions.
- `neutron`. Networking API.
- `keystone`. Identity Service API.
- `glance`. Image Service API.
- `swift`. Object Storage API.
- `cinder`. Block Storage Service API.
- `heat`. Orchestration API.

For example, to install the nova client, run the following command:

```
$ sudo pip install python-novaclient
```

To update the nova client, run the following command:

```
$ sudo pip install --upgrade python-novaclient
```

To remove the nova client, run the following command:

```
$ sudo pip uninstall python-novaclient
```

2. Before you can issue client commands, you must download and source the `openrc` file to set environment variables. Proceed to [the section called "Download and source the OpenStack RC file" \[19\]](#).

Get the version for a client

After you install an OpenStack client, you can get its version number, as follows:

- Run the following command get the version number for a client:

```
$ PROJECT --version
```

Where *PROJECT* is the project name and has one of the following values:

- *nova*. Compute API and extensions.
- *neutron*. Networking API.
- *keystone*. Identity Service API.
- *glance*. Image Service API.
- *swift*. Object Storage API.
- *cinder*. Block Storage Service API.
- *heat*. Orchestration API.

For example, to see the version of the nova client, run the following command:

```
$ nova --version
```

```
2.14.1.17
```

To see the version of the keystone client, run the following command:

```
$ keystone --version
```

```
0.3.1.73
```

Download and source the OpenStack RC file

To set the required environment variables for the OpenStack command-line clients, you must download and source an environment file, `openrc.sh`. It is project-specific and contains the credentials used by OpenStack Compute, Image, and Identity services.

When you source the file and enter the password, environment variables are set for that shell. They allow the commands to communicate to the OpenStack services that run in the cloud.

You can download the file from the OpenStack dashboard as an administrative user or any other user.

1. Log in to the OpenStack dashboard, choose the project for which you want to download the OpenStack RC file, and click **Access & Security**.
2. Click **Download OpenStack RC File** and save the file.
3. Copy the `openrc.sh` file to the machine from where you want to run OpenStack commands.

For example, copy the file to the machine from where you want to upload an image with a glance client command.

4. On any shell from where you want to run OpenStack commands, source the `openrc.sh` file for the respective project.

In this example, we source the `demo-openrc.sh` file for the demo project:

```
$ source demo-openrc.sh
```

5. When you are prompted for an OpenStack password, enter the OpenStack password for the user who downloaded the `openrc.sh` file.
6. When you run OpenStack client commands, you can override some environment variable settings by using the options that are listed at the end of the **nova help** output. For example, you can override the `OS_PASSWORD` setting in the `openrc.sh` file by specifying a password on a nova command, as follows:

```
$ nova --password <password> image-list
```

Where *password* is your password.

Manage images

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

You can upload images through the glance client or the Image Service API. You can also use the nova client to list images, set and delete image metadata, delete images, and take a snapshot of a running instance to create an image. After you upload an image, you cannot change it.

For details about image creation, see the [Virtual Machine Image Guide](#).

List or get details for images (glance)

- To list the available images:

```
$ glance image-list
```

ID	Name	Disk Format	Container Format	Size	Status
397e713c-b95b-4186-ad46-6126863ea0a9	cirros-0.3.1-x86_64-uec	ami	ami	25165824	active
df430cc2-3406-4061-b635-a51c16e488ac	cirros-0.3.1-x86_64-uec-kernel	aki	aki	4955792	active
3cf852bd-2332-48f4-9ae4-7d926d50945e	cirros-0.3.1-x86_64-uec-ramdisk	ari	ari	3714968	active
7e5142af-1253-4634-bcc6-89482c5f2e8a	myCirrosImage	ami	ami	14221312	active

You can use `grep` to filter the list, as follows:

```
$ glance image-list | grep 'cirros'
```

397e713c-b95b-4186-ad46-6126863ea0a9	cirros-0.3.1-x86_64-uec	ami	ami	25165824	active
df430cc2-3406-4061-b635-a51c16e488ac	cirros-0.3.1-x86_64-uec-kernel	aki	aki	4955792	active
3cf852bd-2332-48f4-9ae4-7d926d50945e	cirros-0.3.1-x86_64-uec-ramdisk	ari	ari	3714968	active

- To get image details, by name or ID:

```
$ glance image-show myCirrosImage
```

Property	Value
Property 'base_image_ref'	397e713c-b95b-4186-ad46-6126863ea0a9
Property 'image_location'	snapshot
Property 'image_state'	available
Property 'image_type'	snapshot
Property 'instance_type_ephemeral_gb'	0
Property 'instance_type_flavorid'	2
Property 'instance_type_id'	5
Property 'instance_type_memory_mb'	2048
Property 'instance_type_name'	m1.small
Property 'instance_type_root_gb'	20
Property 'instance_type_rxtx_factor'	1
Property 'instance_type_swap'	0
Property 'instance_type_vcpu_weight'	None
Property 'instance_type_vcpus'	1
Property 'instance_uuid'	84c6e57d-a6b1-44b6-81eb-fcb36afd31b5
Property 'kernel_id'	df430cc2-3406-4061-b635-a51c16e488ac
Property 'owner_id'	66265572db174a7aa66eba661f58eb9e
Property 'ramdisk_id'	3cf852bd-2332-48f4-9ae4-7d926d50945e
Property 'user_id'	376744b5910b4b4da7d8e6cb483b06a8
checksum	8e4838effal969ad591655d6485c7ba8
container_format	ami
created_at	2013-07-22T19:45:58
deleted	False
disk_format	ami
id	7e5142af-1253-4634-bcc6-89482c5f2e8a
is_public	False
min_disk	0
min_ram	0
name	myCirrosImage
owner	66265572db174a7aa66eba661f58eb9e
protected	False
size	14221312
status	active
updated_at	2013-07-22T19:46:42

Create or update an image (glance)

1. To upload a CentOS 6.3 image in qcow2 format and configure it for public access:

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

2. To update an image by name or ID:

```
$ glance image-update IMAGE
```

To modify image properties, use the following optional arguments:

<code>--name NAME</code>	The name of the image.
<code>--disk-format DISK_FORMAT</code>	The disk format of the image. Acceptable formats are ami, ari, aki, vhd, vmrk, raw, qcow2, vdi, and iso.
<code>--container-format CONTAINER_FORMAT</code>	The container format of the image. Acceptable formats are ami, ari, aki, bare, and ovf.
<code>--owner TENANT_ID</code>	The tenant who should own the image.
<code>--size SIZE</code>	The size of image data, in bytes.
<code>--min-disk DISK_GB</code>	The minimum size of disk needed to boot image, in gigabytes.
<code>--min-ram DISK_RAM</code>	The minimum amount of ram needed to boot image, in megabytes.
<code>--location IMAGE_URL</code>	The URL where the data for this image resides. For example, if the image data is stored in swift, you could specify <code>swift://account:key@example.com/container/obj</code> .
<code>--file FILE</code>	Local file that contains disk image to be uploaded during update. Alternatively, you can pass images to the client through stdin.
<code>--checksum CHECKSUM</code>	Hash of image data to use for verification.
<code>--copy-from IMAGE_URL</code>	Similar to <code>--location</code> in usage, but indicates that the Glance server should immediately copy the data and store it in its configured image store.
<code>--is-public [True/False]</code>	Makes an image accessible to the public.
<code>--is-protected [True/False]</code>	Prevents an image from being deleted.
<code>--property KEY=VALUE</code>	Arbitrary property to associate with image. Can be used multiple times.
<code>--purge-props</code>	Deletes all image properties that are not explicitly set in the update request. Otherwise, those properties not referenced are preserved.
<code>--human-readable</code>	Prints image size in a human-friendly format.

3. To annotate an image with a property that describes the required VIF model:

```
$ glance image-update \
  --property hw_vif_model=e1000 f16-x86_64-openstack-sda
```

If you specify a VIF model that is not supported, the instance fails to launch. See [Table 2.2, "VIF model values" \[22\]](#).

The valid model values depend on the `libvirt_type` setting, as shown in the following table:

Table 2.2. VIF model values

libvirt_type setting	Supported model values
qemu or kvm	<ul style="list-style-type: none"> • virtio • ne2k_pci • pcnet • rtl8139 • e1000
xen	<ul style="list-style-type: none"> • netfront • ne2k_pci • pcnet • rtl8139 • e1000

Create image (nova)

You can use the nova client to list images, set and delete image metadata, delete images, and take a snapshot of a running instance to create an image.

The safest approach is to shut down the instance before you take a snapshot.

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

1. Write any buffered data to disk.

For more information, see [Taking Snapshots](#) in the *OpenStack Operations Guide*.

2. To create the image, list instances to get the server ID:

```
$ nova list
```

```

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

```

In this example, the server is named `myCirrosServer`. Use this server to create a snapshot, as follows:

```
$ nova image-create myCirrosServer myCirrosImage
```

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

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

```
$ nova image-show IMAGE
```

```

+-----+-----+
| Property | Value |
+-----+-----+

```

metadata owner_id	66265572db174a7aa66eba661f58eb9e
minDisk	0
metadata instance_type_name	m1.small
metadata instance_type_id	5
metadata instance_type_memory_mb	2048
id	7e5142af-1253-4634-bcc6-89482c5f2e8a
metadata instance_type_root_gb	20
metadata instance_type_rxtx_factor	1
metadata ramdisk_id	3cf852bd-2332-48f4-9ae4-7d926d50945e
metadata image_state	available
metadata image_location	snapshot
minRam	0
metadata instance_type_vcpus	1
status	ACTIVE
updated	2013-07-22T19:46:42Z
metadata instance_type_swap	0
metadata instance_type_vcpu_weight	None
metadata base_image_ref	397e713c-b95b-4186-ad46-6126863ea0a9
progress	100
metadata instance_type_flavorid	2
OS-EXT-IMG-SIZE:size	14221312
metadata image_type	snapshot
metadata user_id	376744b5910b4b4da7d8e6cb483b06a8
name	myCirrosImage
created	2013-07-22T19:45:58Z
metadata instance_uuid	84c6e57d-a6b1-44b6-81eb-fcb36afd31b5
server	84c6e57d-a6b1-44b6-81eb-fcb36afd31b5
metadata kernel_id	df430cc2-3406-4061-b635-a51c16e488ac
metadata instance_type_ephemeral_gb	0

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

- To launch an instance from your image, include the image ID and flavor ID, as follows:

```
$ nova boot newServer --image 7e5142af-1253-4634-bcc6-89482c5f2e8a \
--flavor 3
```

Property	Value
OS-EXT-STS:task_state	scheduling
image	myCirrosImage
OS-EXT-STS:vm_state	building
OS-EXT-SRV-ATTR:instance_name	instance-00000007
flavor	m1.medium
id	d7efd3e4-d375-46d1-9d57-372b6e4bdb7f
security_groups	[[{'name': 'u'default'}]]
user_id	376744b5910b4b4da7d8e6cb483b06a8
OS-DCF:diskConfig	MANUAL
accessIPv4	
accessIPv6	
progress	0
OS-EXT-STS:power_state	0
OS-EXT-AZ:availability_zone	nova
config_drive	
status	BUILD
updated	2013-07-22T19:58:33Z
hostId	
OS-EXT-SRV-ATTR:host	None
key_name	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
name	newServer
adminPass	jis88nN46RGP
tenant_id	66265572db174a7aa66eba661f58eb9e
created	2013-07-22T19:58:33Z
metadata	{}

Troubleshoot image creation

- You cannot create a snapshot from an instance that has an attached volume. Detach the volume, create the image, and re-mount the volume.
- Make sure the version of qemu you are using is version 0.14 or greater. Older versions of qemu result in an "unknown option -s" error message in the nova-compute.log.
- Examine the /var/log/nova-api.log and /var/log/nova-compute.log log files for error messages.

Configure access and security for instances

When you launch a virtual machine, you can inject a *keypair*, which provides SSH access to your instance. For this to work, the image must contain the `cloud-init` package. Create at least one keypair for each project. If you generate a keypair with an external tool, you can import it into OpenStack. You can use the keypair for multiple instances that belong to that project. In case an image uses a static root password or a static key set – neither is recommended – you must not provide a keypair when you launch the instance.

A *security group* is a named collection of network access rules that you use to limit the types of traffic that have access to instances. When you launch an instance, you can assign one or more security groups to it. If you do not create security groups, new instances are automatically assigned to the default security group, unless you explicitly specify a different security group. The associated *rules* in each security group control the traffic to instances in the group. Any incoming traffic that is not matched by a rule is denied access by default. You can add rules to or remove rules from a security group. You can modify rules for the default and any other security group.

You must modify the rules for the default security group because users cannot access instances that use the default group from any IP address outside the cloud.

You can modify the rules in a security group to allow access to instances through different ports and protocols. For example, you can modify rules to allow access to instances through SSH, to ping them, or to allow UDP traffic – for example, for a DNS server running on an instance. You specify the following parameters for rules:

- **Source of traffic.** Enable traffic to instances from either IP addresses inside the cloud from other group members or from all IP addresses.
- **Protocol.** Choose TCP for SSH, ICMP for pings, or UDP.
- **Destination port on virtual machine.** Defines a port range. To open a single port only, enter the same value twice. ICMP does not support ports: Enter values to define the codes and types of ICMP traffic to be allowed.

Rules are automatically enforced as soon as you create or modify them.

You can also assign a floating IP address to a running instance to make it accessible from outside the cloud. You assign a floating IP address to an instance and attach a block storage device, or volume, for persistent storage. See [the section called “Manage IP addresses” \[36\]](#).

Add a keypair

You can generate a keypair or upload an existing public key.

1. To generate a keypair, run the following command:

```
$ nova keypair-add KEY_NAME > MY_KEY.pem
```

The command generates a keypair named *KEY_NAME*, writes the private key to the *MY_KEY.pem* file, and registers the public key at the Nova database.

2. To set the permissions of the *MY_KEY.pem* file, run the following command:

```
$ chmod 600 MY_KEY.pem
```

The command changes the permissions of the *MY_KEY.pem* file so that only you can read and write to it.

Import a keypair

1. If you have already generated a keypair with the public key located at *~/.ssh/id_rsa.pub*, run the following command to upload the public key:

```
$ nova keypair-add --pub_key ~/.ssh/id_rsa.pub KEY_NAME
```

The command registers the public key at the Nova database and names the keypair *KEY_NAME*.

2. List keypairs to make sure that the uploaded keypair appears in the list:

```
$ nova keypair-list
```

Create and manage security groups

1. To list security groups for the current project, including descriptions, enter the following command:

```
$ nova secgroup-list
```

2. To create a security group with a specified name and description, enter the following command:

```
$ nova secgroup-create SEC_GROUP_NAME GROUP_DESCRIPTION
```

3. To delete a specified group, enter the following command:

```
$ nova secgroup-delete SEC_GROUP_NAME
```



Note

You cannot delete the default security group for a project. Also, you cannot delete a security group that is assigned to a running instance.

Create and manage security group rules

Modify security group rules with the **nova secgroup-*-rule** commands.

1. On a shell, source the OpenStack RC file. For details, see [the section called “Download and source the OpenStack RC file” \[19\]](#).

2. To list the rules for a security group

```
$ nova secgroup-list-rules SEC_GROUP_NAME
```

3. To allow SSH access to the instances, choose one of the following sub-steps:

- a. **Add rule for all IPs**

Either from all IP addresses (specified as IP subnet in CIDR notation as 0.0.0.0/0):

```
$ nova secgroup-add-rule SEC_GROUP_NAME tcp 22 22 0.0.0.0/0
```

- b. **Add rule for security groups**

Alternatively, you can allow only IP addresses from other security groups (source groups) to access the specified port:

```
$ nova secgroup-add-group-rule --ip_proto tcp --from_port 22 \  
--to_port 22 SEC_GROUP_NAME SOURCE_GROUP_NAME
```

4. To allow pinging the instances, choose one of the following sub-steps:

- a. **To allow pinging from IPs**

Specify all IP addresses as IP subnet in CIDR notation: 0.0.0.0/0. This command allows access to all codes and all types of ICMP traffic, respectively:

```
$ nova secgroup-add-rule SEC_GROUP_NAME icmp -1 -1 0.0.0.0/0
```

- b. **To allow pinging from other security groups**

To allow only members of other security groups (source groups) to ping instances:

```
$ nova secgroup-add-group-rule --ip_proto icmp --from_port -1 \  
--to_port -1 SEC_GROUP_NAME SOURCE_GROUP_NAME
```

5. To allow access through a UDP port, such as allowing access to a DNS server that runs on a VM, complete one of the following sub-steps:

- a. To allow UDP access from IPs, specify all IP addresses as IP subnet in CIDR notation: 0.0.0.0/0.

```
$ nova secgroup-add-rule SEC_GROUP_NAME udp 53 53 0.0.0.0/0
```

- b. To allow only IP addresses from other security groups (source groups) to access the specified port:

```
$ nova secgroup-add-group-rule --ip_proto udp --from_port 53 \  
--to_port 53 SEC_GROUP_NAME SOURCE_GROUP_NAME
```

6. To delete a security group rule, specify the same arguments that you used to create the rule.

To delete the security rule that you created in [Step 3.a \[26\]](#):

```
$ nova secgroup-delete-rule SEC_GROUP_NAME tcp 22 22 0.0.0.0/0
```

To delete the security rule that you created in [Step 3.b \[26\]](#):

```
$ nova secgroup-delete-group-rule --ip_proto tcp --from_port 22 \  
  --to_port 22 SEC_GROUP_NAME SOURCE_GROUP_NAME
```

Launch instances

Instances are virtual machines that run inside the cloud.

Before you can launch an instance, gather the following parameters:

- The **instance source**, which is an image or snapshot. Alternatively, you can boot from a volume, which is block storage, to which you've copied an image or snapshot.
- The **image** or **snapshot**, which represents the operating system.
- A **name** for your instance.
- The **flavor** for your instance, which defines the compute, memory, and storage capacity of nova computing instances. A flavor is an available hardware configuration for a server. It defines the "size" of a virtual server that can be launched.
- `User Data` is a special key in the metadata service that holds a file that cloud-aware applications in the guest instance can access. For example, the `cloudinit` system is an open-source package from Ubuntu that is available on various Linux distributions including Ubuntu, Fedora, and openSUSE and that handles early initialization of a cloud instance that uses `user data`.
- Access and security credentials, which include one or both of the following credentials:
 - A **keypair** for your instance, which are SSH credentials that are injected into images when they are launched. For this to work, the image must contain the `cloud-init` package. Create at least one keypair for each project. If you already have generated a keypair with an external tool, you can import it into OpenStack. You can use the keypair for multiple instances that belong to that project.
 - A **security group**, which defines which incoming network traffic is forwarded to instances. Security groups hold a set of firewall policies, known as *security group rules*.
- If needed, you can assign a **floating (public) IP address** to a running instance and attach a block storage device, or volume, for persistent storage.

After you gather the parameters you need to launch an instance, you can launch it from an [image](#) or a [volume](#).

You can launch an instance directly from one of the available OpenStack images or from an image that you have copied to a persistent volume. The OpenStack Image Service provides a pool of images that are accessible to members of different projects.

Gather parameters to launch an instance

1. On a shell, source the OpenStack RC file. See [the section called “Download and source the OpenStack RC file” \[19\]](#).
2. List the available flavors:

```
$ nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public
1	ml.tiny	512	0	0		1	1.0	True
2	ml.small	2048	20	0		1	1.0	True
3	ml.medium	4096	40	0		2	1.0	True
4	ml.large	8192	80	0		4	1.0	True
42	ml.nano	64	0	0		1	1.0	True
5	ml.xlarge	16384	160	0		8	1.0	True
84	ml.micro	128	0	0		1	1.0	True

Note the ID of the flavor that you want to use for your instance.

3. List the available images:

```
$ nova image-list
```

ID	Name	Status	Server
397e713c-b95b-4186-ad46-6126863ea0a9	cirros-0.3.1-x86_64-uec	ACTIVE	
df430cc2-3406-4061-b635-a51c16e488ac	cirros-0.3.1-x86_64-uec-kernel	ACTIVE	
3cf852bd-2332-48f4-9ae4-7d926d50945e	cirros-0.3.1-x86_64-uec-ramdisk	ACTIVE	

You can also filter the image list by using `grep` to find a specific image, like this:

```
$ nova image-list | grep 'kernel'
```

```
df430cc2-3406-4061-b635-a51c16e488ac | cirros-0.3.1-x86_64-uec-kernel | ACTIVE |
```

Note the ID of the image that you want to boot your instance from.

4. List the available security groups:



Note

If you are an admin user, specify the `--all-tenants` parameter to list groups for all tenants.

```
$ nova secgroup-list --all-tenants
```

Id	Name	Description	Tenant_ID
2	default	default	66265572db174a7aa66eba661f58eb9e
1	default	default	b70d90d65e464582b6b2161cf3603ced

If you have not created any security groups, you can assign the instance to only the default security group.

You can also list rules for a specified security group:

```
$ nova secgroup-list-rules default
```

This example modifies the default security group to allow HTTP traffic on the instance by permitting TCP traffic on Port 80.

5. List the available keypairs.

```
$ nova keypair-list
```

Note the name of the keypair that you use for SSH access.

Launch an instance from an image

- Now you have all parameters required to launch an instance, run the following command and specify the server name, flavor ID, and image ID. Optionally, you can provide a key name for access control and security group for security. You can also include metadata key and value pairs. For example you can add a description for your server by providing the `--meta description="My Server"` parameter.

You can pass user data in a local file at instance launch by using the flag `--user-data USER-DATA-FILE` parameter.

```
$ nova boot --flavor FLAVOR_ID --image IMAGE_ID --key_name KEY_NAME \
--user-data mydata.file --security_group SEC_GROUP NAME_FOR_INSTANCE \
--meta KEY=VALUE --meta KEY=VALUE
```

Depending on the parameters that you provide, the command returns a list of server properties.

A status of `BUILD` indicates that the instance has started, but is not yet online.

A status of `ACTIVE` indicates that the instance is active.

Property	Value
OS-EXT-STS:task_state	scheduling
image	cirros-0.3.1-x86_64-uec
OS-EXT-STS:vm_state	building
OS-EXT-SRV-ATTR:instance_name	instance-00000002
flavor	m1.small
id	b3cdc6c0-85a7-4904-ae85-71918f734048
security_groups	[{'u'name': 'u'default'}]
user_id	376744b5910b4b4da7d8e6cb483b06a8
OS-DCF:diskConfig	MANUAL
accessIPv4	
accessIPv6	
progress	0
OS-EXT-STS:power_state	0
OS-EXT-AZ:availability_zone	nova
config_drive	
status	BUILD
updated	2013-07-16T16:25:34Z
hostId	
OS-EXT-SRV-ATTR:host	None
key_name	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
name	myCirrosServer
adminPass	tVs5pL8HcPGw
tenant_id	66265572db174a7aa66eba661f58eb9e
created	2013-07-16T16:25:34Z
metadata	{}

Copy the server ID value from the `id` field in the output. You use this ID to get details for or delete your server.

Copy the administrative password value from the `adminPass` field. You use this value to log into your server.



Note

Arbitrary local files can also be placed into the instance file system at creation time using the `--file <dst-path>=<src-path>` option. You may store up to 5 files. For example if you have a special `authorized_keys` file named `special_authorized_keysfile` that you want to put on the instance rather than using the regular `ssh` key injection, you can use the following command:

```
$ nova boot --image ubuntu-cloudimage --flavor 1 \  
--file /root/.ssh/authorized_keys=special_authorized_keysfile
```

2. Check if the instance is online:

```
$ nova list
```

The list shows the ID, name, status, and private (and if assigned, public) IP addresses for all instances in the project that you belong to:

ID	Name	Status	Task State	Power State	Networks
84c6e57d-a6b1-44b6-81eb-fcb36afd31b5	myCirrosServer	ACTIVE	None	Running	private=10.0.0.3
8a99547e-7385-4ad1-ae50-4ecfaaad5f42	myInstanceFromVolume	ACTIVE	None	Running	private=10.0.0.4

If the status for the instance is ACTIVE, the instance is online.

To view the available options for the **nova list** command, run the following command:

```
$ nova help list
```

3. If you did not provide a keypair, security groups, or rules, you can only access the instance from inside the cloud through VNC. Even pinging the instance is not possible.

Launch an instance from a volume

After you [create a bootable volume](#), you [launch an instance from the volume](#).

To launch and instance from a volume

1. To create a bootable volume from an image, run the following command:

```
# cinder create --image-id 397e713c-b95b-4186-ad46-6126863ea0a9 \
--display-name my-bootable-vol 8
```



Note

You can also use the `nova volume-create` command to complete the same action:

```
# nova volume-create \
--image-id 397e713c-b95b-4186-ad46-6126863ea0a9 \
--display-name my-bootable-vol 8
```

2. To list volumes, run the following command:

```
$ nova volume-list
```

ID	Status	Display Name	Size	Volume Type	Attached to
bd7cf584-45de-44e3-bf7f-f7b50bf235e3	available	my-bootable-vol	8	None	

Copy the value in the ID field for your volume.

3. To launch an instance, run the `nova boot` command with the `--block_device_mapping` parameter, as follows:

```
$ nova boot --flavor FLAVOR --block_device_mapping \
DEVNAME=ID:TYPE:SIZE:DELETE_ON_TERMINATE NAME
```

The command arguments are:

<code>--flavor FLAVOR</code>	The flavor ID.
<code>--block_device_mapping DEVNAME=ID:type:size:delete-on-terminate</code>	<ul style="list-style-type: none"> • <i>DEVNAME</i>. A device name where the volume is attached in the system at <code>/dev/dev_name</code>. This value is typically <code>vda</code>. • <i>ID</i>. The ID of the volume to boot from, as shown in the output of <code>nova volume-list</code>. • <i>type</i>. Either <code>snap</code> or any other value, including a blank string. <code>snap</code> means that the volume was created from a snapshot. • <i>size</i>. The size of the volume, in GBs. It is safe to leave this blank and have the Compute service infer the size. • <i>delete-on-terminate</i>. Boolean. Indicates whether the volume is deleted when the instance is deleted. You can specify: <ul style="list-style-type: none"> • True or 1 • False or 0
<i>NAME</i>	The name for the server.



Note

You must specify an image when booting from a volume, even though the specified image is not used. Otherwise, the `Attempt to boot from volume - no image supplied` error is returned.

For example, you might enter the following command to boot from a volume with ID `bd7cf584-45de-44e3-bf7f-f7b50bf235e`. The volume is not deleted when the instance is terminated:

```
$ nova boot --flavor 2 --image 397e713c-b95b-4186-ad46-6126863ea0a9
--block_device_mapping vda=bd7cf584-45de-44e3-bf7f-f7b50bf235e:::0
myInstanceFromVolume
```

Property	Value
OS-EXT-STS:task_state	scheduling
image	cirros-0.3.1-x86_64-uec
OS-EXT-STS:vm_state	building
OS-EXT-SRV-ATTR:instance_name	instance-00000003
flavor	m1.small
id	8a99547e-7385-4ad1-ae50-4ecfaaad5f42
security_groups	[{'name': 'u'default'}]
user_id	376744b5910b4b4da7d8e6cb483b06a8
OS-DCF:diskConfig	MANUAL
accessIPv4	
accessIPv6	
progress	0
OS-EXT-STS:power_state	0
OS-EXT-AZ:availability_zone	nova
config_drive	
status	BUILD
updated	2013-07-16T20:14:19Z
hostId	
OS-EXT-SRV-ATTR:host	None
key_name	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
name	myInstanceFromVolume
adminPass	LhwEmx5Ho22M
tenant_id	66265572db174a7aa66eba661f58eb9e
created	2013-07-16T20:14:18Z
metadata	{}

Now when you list volumes, you can see that the volume is attached to a server:

```
$ nova volume-list
```

ID	Status	Display Name	Size	Volume Type	Attached to
bd7cf584-45de-44e3-bf7f-f7b50bf235e3 8a99547e-7385-4ad1-ae50-4ecfaaad5f42	in-use	my-bootable-vol	8	None	

Additionally, when you list servers, you see the server that you booted from a volume:

```
$ nova list
```

ID	Name	Status	Task State	Power State
Networks				

```
+-----+-----+-----+-----+-----+
| b3cdc6c0-85a7-4904-ae85-71918f734048 | myCirrosServer      | ACTIVE | None   | Running
| private=10.0.0.3 |
| 8a99547e-7385-4ad1-ae50-4ecfaaad5f42 | myInstanceFromVolume | ACTIVE | None   | Running
| private=10.0.0.4 |
+-----+-----+-----+-----+-----+
```

Manage instances and hosts

Instances are virtual machines that run inside the cloud.

Manage IP addresses

Floating IP Addresses Overview

Each instance can have a private, or fixed, IP address and a public, or floating, one.

Private IP addresses are used for communication between instances, and public ones are used for communication with the outside world.

When you launch an instance, it is automatically assigned a private IP address that stays the same until you explicitly terminate the instance. Rebooting an instance has no effect on the private IP address.

A pool of floating IPs, configured by the cloud operator, is available in OpenStack Compute.

You can allocate a certain number of these to a project: The maximum number of floating IP addresses per project is defined by the quota.

You can add a floating IP address from this set to an instance of the project. Floating IP addresses can be dynamically disassociated and associated with other instances of the same project at any time.

Before you can assign a floating IP address to an instance, you first must allocate floating IPs to a project. After floating IP addresses have been allocated to the current project, you can assign them to running instances.

You can assign a floating IP address to one instance at a time.

List floating IP address information

1. To list all floating IP addresses:

```
$ nova floating-ip-bulk-list
```

project_id	address	instance_uuid	pool	interface
None	172.24.4.225	None	public	eth0
None	172.24.4.226	None	public	eth0
None	172.24.4.227	None	public	eth0
None	172.24.4.228	None	public	eth0
None	172.24.4.229	None	public	eth0
None	172.24.4.230	None	public	eth0
None	172.24.4.231	None	public	eth0
None	172.24.4.232	None	public	eth0
None	172.24.4.233	None	public	eth0
None	172.24.4.234	None	public	eth0
None	172.24.4.235	None	public	eth0
None	172.24.4.236	None	public	eth0
None	172.24.4.237	None	public	eth0

None	172.24.4.238	None	public	eth0
None	192.168.253.1	None	test	eth0
None	192.168.253.2	None	test	eth0
None	192.168.253.3	None	test	eth0
None	192.168.253.4	None	test	eth0
None	192.168.253.5	None	test	eth0
None	192.168.253.6	None	test	eth0

2. To list all pools that provide floating IP addresses:

```
$ nova floating-ip-pool-list
```

```
+-----+
| name  |
+-----+
| public|
| test  |
+-----+
```

Assign floating IP addresses to projects and instances

To allocate a floating IP address to the current project

1. If more than one pool of IP addresses is available, you can specify the pool from which to allocate the IP address. In this example, the pool name is `public`:

```
$ floating-ip-create public
```

```
+-----+-----+-----+-----+
| Ip          | Instance Id | Fixed Ip | Pool    |
+-----+-----+-----+-----+
| 172.24.4.225 | None        | None     | public  |
+-----+-----+-----+-----+
```

2. To release a floating IP address from the current project:

```
$ nova floating-ip-delete FLOATING_IP
```

The IP address is returned to the pool of IP addresses that are available for all projects. If an IP address is assigned to a running instance, it is disassociated from the instance.

3. To associate an IP address with an instance, at least one floating IP address must be allocated to the current project.

To assign a floating IP address to an instance:

```
$ nova add-floating-ip INSTANCE_NAME_OR_ID FLOATING_IP
```

After you assign the IP address and configure security group rules for the instance, the instance is publicly available at the floating IP address.

4. To remove a floating IP address from an instance, you must specify the same arguments that you used to assign the IP.

To remove a floating IP address from an instance:

```
$ nova remove-floating-ip INSTANCE_NAME_OR_ID FLOATING_IP
```

Change the size of your server

You change the size of a server by changing its flavor.

To change the size of your server

1. List the available flavors:

```
$ nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public
1	m1.tiny	512	0	0		1		1.0
2	m1.small	2048	20	0		1		1.0
3	m1.medium	4096	40	0		2		1.0
4	m1.large	8192	80	0		4		1.0
42	m1.nano	64	0	0		1		1.0
5	m1.xlarge	16384	160	0		8		1.0
84	m1.micro	128	0	0		1		1.0

2. Show information about your server, including its size:

```
$ nova show myCirrosServer
```

Property	Value
status	ACTIVE
updated	2013-07-18T15:08:20Z
OS-EXT-STS:task_state	None
OS-EXT-SRV-ATTR:host	devstack-grizzly
key_name	None
image	cirros-0.3.1-x86_64-uec (397e713c-b95b-4186-ad46-6126863ea0a9)
private network	10.0.0.3
hostId	6e1e69b71ac9b1e6871f91e2dfc9a9b9ceca0f05db68172a81d45385

```

| OS-EXT-STS:vm_state | active
| OS-EXT-SRV-ATTR:instance_name | instance-00000005
| OS-EXT-SRV-ATTR:hypervisor_hostname | devstack-grizzly
| flavor | m1.small (2)
| id | 84c6e57d-a6b1-44b6-81eb-
fcb36afd31b5
| security_groups | [{u'name': u'default'}]
| user_id | 376744b5910b4b4da7d8e6cb483b06a8
| name | myCirrosServer
| created | 2013-07-18T15:07:59Z
| tenant_id | 66265572db174a7aa66eba661f58eb9e
| OS-DCF:diskConfig | MANUAL
| metadata | {u'description': u'Small test
image', u'creator': u'joecool'}
| accessIPv4 |
| accessIPv6 |
| progress | 0
| OS-EXT-STS:power_state | 1
| OS-EXT-AZ:availability_zone | nova
| config_drive |
+-----+
+-----+

```

The size of the server is `m1.small (2)`.

3. To resize the server, pass the server ID and the desired flavor to the nova **resize** command. Include the `--poll` parameter to report the resize progress.

```
$ nova resize myCirrosServer 4 --poll
```

```
Instance resizing... 100% complete
Finished
```

4. Show the status for your server:

```
$ nova list
```

```

+-----+-----+-----+-----+
+-----+
| ID | Name | Status | Networks |
+-----+-----+-----+-----+
+-----+

```

```
| 970e4ca0-f9b7-4c44-80ed-bf0152c96ae1 | resize-demo | RESIZE | private=  
172.16.101.6, public=10.4.113.6 |  
+-----+-----+-----+  
+-----+-----+-----+
```

5. When the resize completes, the status becomes `VERIFY_RESIZE`. To confirm the resize:

```
$ nova resize-confirm 6beefcf7-9de6-48b3-9ba9-e11b343189b3
```

The server status becomes `ACTIVE`.

6. If the resize fails or does not work as expected, you can revert the resize:

```
$ nova resize-revert 6beefcf7-9de6-48b3-9ba9-e11b343189b3
```

The server status becomes `ACTIVE`.

Stop and start an instance

Use one of the following methods to stop and start an instance.

Pause and un-pause an instance

To pause and un-pause a server

- To pause a server, run the following command:

```
$ nova pause SERVER
```

This command stores the state of the VM in RAM. A paused instance continues to run in a frozen state.

To un-pause the server, run the following command:

```
$ nova unpause SERVER
```

Suspend and resume an instance

To suspend and resume a server

Administrative users might want to suspend an infrequently used instance or to perform system maintenance.

1. When you suspend an instance, its VM state is stored on disk, all memory is written to disk, and the virtual machine is stopped. Suspending an instance is similar to placing a device in hibernation; memory and vCPUs become available.

To initiate a hypervisor-level suspend operation, run the following command:

```
$ nova suspend SERVER
```

2. To resume a suspended server:

```
$ nova resume SERVER
```

Reboot an instance

You can perform a soft or hard reboot of a running instance. A soft reboot attempts a graceful shutdown and restart of the instance. A hard reboot power cycles the instance.

To reboot a server

- By default, when you reboot a server, it is a soft reboot.

```
$ nova reboot SERVER
```

To perform a hard reboot, pass the `--hard` parameter, as follows:

```
$ nova reboot --hard SERVER
```

Evacuate instances

If a cloud compute node fails due to a hardware malfunction or another reason, you can evacuate instances to make them available again.

You can choose evacuation parameters for your use case.

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

To evacuate your server

1. To find a different host for the evacuated instance, run the following command to lists hosts:

```
$ nova host-list
```

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

```
$ nova evacuate evacuated_server_name host_b
```

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

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

3. To preserve the user disk data on the evacuated server, deploy OpenStack Compute with shared filesystem. To configure your system, see [Configure migrations guide](#). In this example, the password remains unchanged.

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

Delete an instance

When you no longer need an instance, you can delete it.

To delete an instance

1. List all instances:

```
$ nova list
```

ID	Task State	Power State	Networks	Name	Status
84c6e57d-a6b1-44b6-81eb-fcb36afd31b5	None	Running	private=10.0.0.3	myCirrosServer	ACTIVE
8a99547e-7385-4ad1-ae50-4ecfaaad5f42	None	Running	private=10.0.0.4	myInstanceFromVolume	ACTIVE
d7efd3e4-d375-46d1-9d57-372b6e4bdb7f	None	NOSTATE		newServer	ERROR

2. Use the following command to delete the `newServer` instance, which is in `ERROR` state:

```
$ nova delete newServer
```

3. The command does not notify that your server was deleted.

Instead, run the `nova list` command:

```
$ nova list
```

ID	Task State	Power State	Networks	Name	Status
84c6e57d-a6b1-44b6-81eb-fcb36afd31b5	None	Running	private=10.0.0.3	myCirrosServer	ACTIVE
8a99547e-7385-4ad1-ae50-4ecfaaad5f42	None	Running	private=10.0.0.4	myInstanceFromVolume	ACTIVE

The deleted instance does not appear in the list.

Get a console to an instance

To get a console to an instance

- To get a VNC console to an instance, run the following command:

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

The command returns a URL from which you can access your instance:

```
+-----+
+-----+
+
| Type   | Url
+-----+
+
| xvpvnc | http://166.78.190.96:6081/console?token=
c83ae3a3-15c4-4890-8d45-aefb494a8d6c |
+-----+
+-----+
+
```



Note

To get a non-VNC console, specify the *novnc* parameter instead of the *xvpvnc* parameter.

Manage bare metal nodes

If you use the bare metal driver, you must create and add a network interface to a bare metal node. Then, you can launch an instance from a bare metal image.

You can list and delete bare metal nodes. When you delete a node, any associated network interfaces are removed. You can list and remove network interfaces that are associated with a bare metal node.

Commands

- **baremetal-interface-add**
Adds a network interface to a bare metal node.
- **baremetal-interface-list**
Lists network interfaces associated with a bare metal node.
- **baremetal-interface-remove**
Removes a network interface from a bare metal node.
- **baremetal-node-create**
Creates a bare metal node.
- **baremetal-node-delete**
Removes a bare metal node and any associated interfaces.
- **baremetal-node-list**

Lists available bare metal nodes.

- **baremetal-node-show**

Shows information about a bare metal node.

To manage bare metal nodes

1. Create a bare metal node:

```
$ nova baremetal-node-create --pm_address=1.2.3.4 --pm_user=ipmi --
pm_password=ipmi $(hostname -f) 1 512 10 aa:bb:cc:dd:ee:ff
```

Property	Value
instance_uuid	None
pm_address	1.2.3.4
interfaces	[]
prov_vlan_id	None
cpus	1
memory_mb	512
prov_mac_address	aa:bb:cc:dd:ee:ff
service_host	ubuntu
local_gb	10
id	1
pm_user	ipmi
terminal_port	None

2. Add a network interface to the node:

```
$ nova baremetal-interface-add 1 aa:bb:cc:dd:ee:ff
```

Property	Value
datapath_id	0
id	1
port_no	0
address	aa:bb:cc:dd:ee:ff

3. Launch an instance from a bare metal image:

```
$ nova boot --image my-baremetal-image --flavor my-baremetal-flavor test
```

Property	Value
status	BUILD
id	cc302a8f-cd81-484b-89a8-b75eb3911b1b

... wait for instance to become active ...

4. You can list bare metal nodes and interfaces, as follows:

```
$ nova baremetal-node-list
```


When a node is in use, its status includes the UUID of the instance that runs on it:

```

+-----+-----+-----+-----+-----+-----+
| ID | Host   | CPUs | Memory_MB | Disk_GB | MAC Address |
| VLAN | PM Address | PM Username | PM Password | Terminal Port |
+-----+-----+-----+-----+-----+-----+
| 1 | ubuntu | 1    | 512       | 10      | aa:bb:cc:dd:ee:ff |
| None | 1.2.3.4 | ipmi |           |         | None              |
+-----+-----+-----+-----+-----+-----+

```

5. Show details for a bare metal node:

```
$ nova baremetal-node-show 1
```

```

+-----+-----+
| Property          | Value |
+-----+-----+
| instance_uuid     | cc302a8f-cd81-484b-89a8-b75eb3911b1b |
| pm_address        | 1.2.3.4 |
| interfaces        | [{u'datapath_id': u'0', u'id': 1, u'port_no': 0, u'address': u'aa:bb:cc:dd:ee:ff'}] |
| prov_vlan_id      | None |
| cpus              | 1 |
| memory_mb         | 512 |
| prov_mac_address  | aa:bb:cc:dd:ee:ff |
| service_host      | ubuntu |
| local_gb          | 10 |
| id                | 1 |
| pm_user           | ipmi |
| terminal_port     | None |
+-----+-----+

```

Show usage statistics for hosts and instances

You can show basic statistics on resource usage for hosts and instances.



Note

For more sophisticated monitoring, see the [Ceilometer](#) project, which is under development. You can also use tools, such as [Ganglia](#) or [Graphite](#), to gather more detailed data.

To show host usage statistics

1. List the hosts and the nova-related services that run on them:

```
$ nova host-list
```

```

+-----+-----+-----+
| host_name          | service | zone |
+-----+-----+-----+
| devstack-grizzly  | conductor | internal |
| devstack-grizzly  | compute  | nova |
| devstack-grizzly  | cert     | internal |
+-----+-----+-----+

```

devstack-grizzly	network	internal
devstack-grizzly	scheduler	internal
devstack-grizzly	consoleauth	internal

2. Get a summary of resource usage of all of the instances running on the host.

```
$ nova host-describe devstack-grizzly
```

HOST	PROJECT	cpu	memory_mb	disk_gb
devstack-grizzly	(total)	2	4003	157
devstack-grizzly	(used_now)	3	5120	40
devstack-grizzly	(used_max)	3	4608	40
devstack-grizzly	b70d90d65e464582b6b2161cf3603ced	1	512	0
devstack-grizzly	66265572db174a7aa66eba661f58eb9e	2	4096	40

The `cpu` column shows the sum of the virtual CPUs for instances running on the host.

The `memory_mb` column shows the sum of the memory (in MB) allocated to the instances that run on the hosts.

The `disk_gb` column shows the sum of the root and ephemeral disk sizes (in GB) of the instances that run on the hosts.



Note

These values are computed by using only information about the flavors of the instances that run on the hosts. This command does not query the CPU usage, memory usage, or hard disk usage of the physical host.

To show instance usage statistics

1. Get CPU, memory, I/O, and network statistics for an instance.

First, list instances:

```
$ nova list
```

ID	Name	Status
84c6e57d-a6b1-44b6-81eb-fcb36afd31b5	myCirrosServer	ACTIVE

```
| 8a99547e-7385-4ad1-ae50-4ecfaaad5f42 | myInstanceFromVolume | ACTIVE |
None          | Running          | private=10.0.0.4 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Then, get diagnostic statistics:

```
$ nova diagnostics myCirrosServer
```

```
+-----+-----+
| Property          | Value          |
+-----+-----+
| vnet1_rx          | 1210744        |
| cpu0_time         | 19624610000000 |
| vda_read          | 0              |
| vda_write         | 0              |
| vda_write_req     | 0              |
| vnet1_tx          | 863734         |
| vnet1_tx_errors   | 0              |
| vnet1_rx_drop     | 0              |
| vnet1_tx_packets  | 3855           |
| vnet1_tx_drop     | 0              |
| vnet1_rx_errors   | 0              |
| memory            | 2097152        |
| vnet1_rx_packets  | 5485           |
| vda_read_req      | 0              |
| vda_errors        | -1             |
+-----+-----+
```

2. Get summary statistics for each tenant:

```
$ nova usage-list
```

```
Usage from 2013-06-25 to 2013-07-24:
+-----+-----+-----+-----+
+-----+
| Tenant ID          | Instances | RAM MB-Hours | CPU Hours |
| Disk GB-Hours     |           |               |           |
+-----+-----+-----+-----+
+-----+
| b70d90d65e464582b6b2161cf3603ced | 1          | 344064.44    | 672.00   |
| 0.00                |           |               |           |
| 66265572db174a7aa66eba661f58eb9e | 3          | 671626.76    | 327.94   |
| 6558.86              |           |               |           |
+-----+-----+-----+-----+
+-----+
```

Create and manage networks

Before you run commands, set the following environment variables:

```
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://localhost:5000/v2.0
```

Create networks

1. List the extensions of the system:

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

alias	name
agent_scheduler	Agent Schedulers
binding	Port Binding
quotas	Quota management support
agent	agent
provider	Provider Network
router	Neutron L3 Router
lbaas	LoadBalancing service
extraroute	Neutron Extra Route

2. Create a network:

```
$ neutron net-create net1
```

```
Created a new network:
```

Field	Value
admin_state_up	True
id	2d627131-c841-4e3a-ace6-f2dd75773b6d
name	net1
provider:network_type	vlan
provider:physical_network	physnet1
provider:segmentation_id	1001
router:external	False
shared	False
status	ACTIVE
subnets	
tenant_id	3671f46ec35e4bbca6ef92ab7975e463



Note

Some fields of the created network are invisible to non-admin users.

3. Create a network with specified provider network type:

```
$ neutron net-create net2 --provider:network-type local
```

```
Created a new network:
```

Field	Value
admin_state_up	True
id	524e26ea-fad4-4bb0-b504-1ad0dc770e7a
name	net2
provider:network_type	local
provider:physical_network	
provider:segmentation_id	
router:external	False
shared	False
status	ACTIVE
subnets	
tenant_id	3671f46ec35e4bbca6ef92ab7975e463

Just as shown previous, the unknown option `--provider:network-type` is used to create a local provider network.

Create subnets

- Create a subnet:

```
$ neutron subnet-create net1 192.168.2.0/24 --name subnet1
```

Created a new subnet:

Field	Value
allocation_pools	{"start": "192.168.2.2", "end": "192.168.2.254"}
cidr	192.168.2.0/24
dns_nameservers	
enable_dhcp	True
gateway_ip	192.168.2.1
host_routes	
id	15a09f6c-87a5-4d14-b2cf-03d97cd4b456
ip_version	4
name	subnet1
network_id	2d627131-c841-4e3a-ace6-f2dd75773b6d
tenant_id	3671f46ec35e4bbca6ef92ab7975e463

In the previous command, `net1` is the network name, `192.168.2.0/24` is the subnet's CIDR. They are positional arguments. `--name subnet1` is an unknown option, which specifies the subnet's name.

Create ports

1. Create a port with specified IP address:

```
$ neutron port-create net1 --fixed-ip ip_address=192.168.2.40
```

Created a new port:

Field	Value

```

+-----+
+-----+
+
| admin_state_up      | True
|
| binding:capabilities | {"port_filter": false}
|
| binding:vif_type    | ovs
|
| device_id           |
|
| device_owner        |
|
| fixed_ips            | {"subnet_id": "15a09f6c-87a5-4d14-
b2cf-03d97cd4b456", "ip_address": "192.168.2.40"} |
| id                   | f7a08fe4-e79e-4b67-bbb8-a5002455a493
|
| mac_address         | fa:16:3e:97:e0:fc
|
| name                 |
|
| network_id          | 2d627131-c841-4e3a-ace6-f2dd75773b6d
|
| status               | DOWN
|
| tenant_id           | 3671f46ec35e4bbca6ef92ab7975e463
|
+-----+
+-----+
+

```

In the previous command, `net1` is the network name, which is a positional argument. `--fixed-ip ip_address=192.168.2.40` is an option, which specifies the port's fixed IP address we wanted.

2. Create a port without specified IP address:

```
$ neutron port-create net1
```

```
Created a new port:
```

```

+-----+
+-----+
+
| Field| Value
|
+-----+
+-----+
+
| admin_state_up      | True
|
| binding:capabilities | {"port_filter": false}
|
| binding:vif_type    | ovs
|
| device_id           |
|
| device_owner        |
|
| fixed_ips            | {"subnet_id": "15a09f6c-87a5-4d14-
b2cf-03d97cd4b456", "ip_address": "192.168.2.2"} |

```

```

| id | baf13412-2641-4183-9533-de8f5b91444c
| mac_address | fa:16:3e:f6:ec:c7
| name |
| network_id | 2d627131-c841-4e3a-ace6-f2dd75773b6d
| status | DOWN
| tenant_id | 3671f46ec35e4bbca6ef92ab7975e463
+-----+
+-----+
+

```

We can see that the system will allocate one IP address if we don't specify the IP address in command line.

3. Query ports with specified fixed IP addresses:

```
$ neutron port-list --fixed-ips ip_address=192.168.2.2 ip_address=192.168.2.40
```

```

+-----+-----+-----+
+-----+-----+-----+
+
| id | name | mac_address |
+-----+-----+-----+
| fixed_ips |
+-----+-----+-----+
+
| baf13412-2641-4183-9533-de8f5b91444c | | fa:16:3e:f6:ec:c7 |
| {"subnet_id": "15a09f6c-87a5-4d14-b2cf-03d97cd4b456", "ip_address": "192.168.2.2"} |
| f7a08fe4-e79e-4b67-bbb8-a5002455a493 | | fa:16:3e:97:e0:fc |
| {"subnet_id": "15a09f6c-87a5-4d14-b2cf-03d97cd4b456", "ip_address": "192.168.2.40"} |
+-----+-----+-----+
+-----+-----+-----+
+

```

`--fixed-ips ip_address=192.168.2.2 ip_address=192.168.2.40` is one unknown option.

How to find unknown options? The unknown options can be easily found by watching the output of `create_xxx` or `show_xxx` command. For example, in the port creation command, we see the `fixed_ips` fields, which can be used as an unknown option.

Create and manage stacks

To create a stack from an example template file

1. To create a stack, or template, from an [example template file](#), run following command:

```
$ heat stack-create mystack --template-file=/path/to/heat/templates/WordPress_Single_Instance.template --parameters="InstanceType=m1.large;DBUsername=wp;DBPassword=verybadpassword;KeyName=heat_key;LinuxDistribution=F17"
```

The `--parameters` values that you specify depend on which parameters are defined in the template. If the template file is hosted on a website, you can specify the URL with `--template-url` parameter instead of the `--template-file` parameter.

The command returns the following output:

```
+-----+-----+-----+
+-----+-----+-----+
| id                | stack_name | stack_status |
| creation_time    |           |             |
+-----+-----+-----+
+-----+-----+-----+
| 4c712026-dcd5-4664-90b8-0915494c1332 | mystack    |             |
| CREATE_IN_PROGRESS | 2013-04-03T23:22:08Z |             |
+-----+-----+-----+
+-----+-----+-----+
```

2. You can also use the `stack-create` command to validate a template file without creating a stack from it.

To do so, run the following command:

```
$ heat stack-create mystack --template-file=/path/to/heat/templates/WordPress_Single_Instance.template
```

If validation fails, the response returns an error message.

To list stacks

- To see which stacks are visible to the current user, run the following command:

```
$ heat stack-list
```

```
+-----+-----+-----+
+-----+-----+-----+
| id                | stack_name | stack_status |
| creation_time    |           |             |
+-----+-----+-----+
+-----+-----+-----+
| 4c712026-dcd5-4664-90b8-0915494c1332 | mystack    | CREATE_COMPLETE |
| 2013-04-03T23:22:08Z |           |             |
| 7edc7480-bda5-4e1c-9d5d-f567d3b6a050 | my-otherstack | CREATE_FAILED |
| 2013-04-03T23:28:20Z |           |             |
+-----+-----+-----+
+-----+-----+-----+
```


To view stack details

To explore the state and history of a particular stack, you can run a number of commands.

1. To show the details of a stack, run the following command:

```
$ heat stack-show mystack
```

2. A stack consists of a collection of resources. To list the resources, including their status, in a stack, run the following command:

```
$ heat resource-list mystack
```

```
+-----+-----+-----+
+-----+
| logical_resource_id | resource_type      | resource_status |
| updated_time       |                    |                 |
+-----+-----+-----+
+-----+
| WikiDatabase       | AWS::EC2::Instance | CREATE_COMPLETE |
| 2013-04-03T23:25:56Z |                    |                 |
+-----+-----+-----+
+-----+
```

3. To show the details for the specified resource in a stack, run the following command:

```
$ heat resource-show mystack WikiDatabase
```

Some resources have associated metadata which can change throughout the life-cycle of a resource:

```
$ heat resource-metadata mystack WikiDatabase
```

4. A series of events is generated during the life-cycle of a stack. This command will display those events.

```
$ heat event-list mystack
```

```
+-----+-----+-----+-----+
+-----+
| logical_resource_id | id | resource_status_reason | resource_status |
| event_time         |    |                        |                 |
+-----+-----+-----+-----+
+-----+
| WikiDatabase       | 1  | state changed         | IN_PROGRESS    |
| 2013-04-03T23:22:09Z |    |                        |                 |
| WikiDatabase       | 2  | state changed         | CREATE_COMPLETE |
| 2013-04-03T23:25:56Z |    |                        |                 |
+-----+-----+-----+-----+
+-----+
```

5. To show the details for a particular event, run the following command:

```
$ heat event-show WikiDatabase 1
```

To update a stack

- To update an existing stack from a modified template file, run a command like the following command:

```
$ heat stack-update mystack --template-file=/path/to/heat/templates/  
WordPress_Single_Instance_v2.template  
--parameters="InstanceType=m1.large;DBUsername=wp;DBPassword=  
verybadpassword;KeyName=heat_key;LinuxDistribution=F17"
```

```
+-----+-----+-----+  
+-----+  
| id                | stack_name  | stack_status  |  
creation_time      |             |               |  
+-----+-----+-----+  
+-----+  
| 4c712026-dcd5-4664-90b8-0915494c1332 | mystack     | UPDATE_COMPLETE |  
2013-04-03T23:22:08Z |             |                 |  
| 7edc7480-bda5-4e1c-9d5d-f567d3b6a050 | my-otherstack | CREATE_FAILED   |  
2013-04-03T23:28:20Z |             |                 |  
+-----+-----+-----+  
+-----+
```

Some resources are updated in-place, while others are replaced with new resources.

Manage volumes

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

This example creates a volume named `my-volume` based on an image.

To list volumes

1. To determine which image to use for your volume, list images:

```
$ nova image-list
```

ID	Server	Name
397e713c-b95b-4186-ad46-6126863ea0a9		cirros-0.3.1-x86_64-uec
ACTIVE		
df430cc2-3406-4061-b635-a51c16e488ac		cirros-0.3.1-x86_64-uec-kernel
ACTIVE		
3cf852bd-2332-48f4-9ae4-7d926d50945e		cirros-0.3.1-x86_64-uec-ramdisk
ACTIVE		
7e5142af-1253-4634-bcc6-89482c5f2e8a		myCirrosImage
ACTIVE	84c6e57d-a6b1-44b6-81eb-fcb36afd31b5	
89bcd424-9d15-4723-95ec-61540e8a1979		mynsnapshot
ACTIVE	f51ebd07-c33d-4951-8722-1df6aa8afaa4	

Note the ID of the image that you want to use.

2. To determine which availability zone are available in which to create your volume, list the availability zones:

```
$ nova availability-zone-list
```

Name	Status
internal	available
- devstack-grizzly	
- nova-conductor	enabled (-) 2013-07-25T16:50:44.000000
- nova-consoleauth	enabled (-) 2013-07-25T16:50:44.000000
- nova-scheduler	enabled (-) 2013-07-25T16:50:44.000000
- nova-cert	enabled (-) 2013-07-25T16:50:44.000000
- nova-network	enabled (-) 2013-07-25T16:50:44.000000
nova	available
- devstack-grizzly	
- nova-compute	enabled (-) 2013-07-25T16:50:39.000000

Note the name of an available availability zone that you want to use.

3. Create a volume with 8 GBs of space in the desired availability zone and based on the desired image, as follows:

```
$ cinder create 8 --display-name my-new-volume --image-id 397e713c-b95b-4186-ad46-6126863ea0a9 --availability-zone nova
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2013-07-25T17:02:12.472269
display_description	None
display_name	my-new-volume
id	573e024d-5235-49ce-8332-be1576d323f8
image_id	397e713c-b95b-4186-ad46-6126863ea0a9
metadata	{}
size	8
snapshot_id	None
source_volid	None
status	creating
volume_type	None

4. To verify that your volume was created successfully, list the available volumes:

```
$ cinder list
```

Size	Volume ID	Type	Bootable	Attached to	Status	Display Name	
8	573e024d-5235-49ce-8332-be1576d323f8	None	true		available	my-new-volume	8
8	bd7cf584-45de-44e3-bf7f-f7b50bf235e3	None	true		available	my-bootable-vol	8

If your volume was created successfully, its status is `available`. If its status is `error`, you might have tried to create a volume outside of your quota.

5. Attach your volume to a server:

```
$ nova volume-attach 84c6e57d-a6b1-44b6-81eb-fcb36afd31b5 573e024d-5235-49ce-8332-be1576d323f8 /dev/vdb
```

Property	Value
device	/dev/vdb
serverId	84c6e57d-a6b1-44b6-81eb-fcb36afd31b5
id	573e024d-5235-49ce-8332-be1576d323f8
volumeId	573e024d-5235-49ce-8332-be1576d323f8

Note the ID of your volume.

6. Show information for your volume:

```
$ cinder show 573e024d-5235-49ce-8332-be1576d323f8
```

Property	Value
attachments	[[{'device': u'/dev/vdb', u'server_id': u'84c6e57d-a6b1-44b6-81eb-fcb36afd31b5', u'id': u'573e024d-5235-49ce-8332-be1576d323f8', u'volume_id': u'573e024d-5235-49ce-8332-be1576d323f8'}]]
availability_zone	nova
bootable	true
created_at	2013-07-25T17:02:12.000000
display_description	None
display_name	my-new-volume
id	573e024d-5235-49ce-8332-be1576d323f8
metadata	{}
os-vol-host-attr:host	devstack-grizzly
os-vol-tenant-attr:tenant_id	66265572db174a7aa66eba661f58eb9e
size	8
snapshot_id	None

```

|          source_volid          |          |          None          |
|          status          |          |          in-use          |
|          volume_image_metadata          | {u'kernel_id': u'df430cc2-3406-4061- |
b635-a51c16e488ac', u'image_id': u'397e713c-b95b-4186-ad46-6126863ea0a9', |
u'ramdisk_id': u'3cf852bd-2332-48f4-9ae4-7d926d50945e', u'image_name': |
u'cirros-0.3.1-x86_64-uec'} |          |
|          volume_type          |          |          None          |
+-----+-----+-----+
+-----+-----+-----+
+

```

From the output, you can see that the volume is attached to the server with ID 84c6e57d-a6b1-44b6-81eb-fcb36afd31b5, is in the nova availability zone, and is bootable.

- To delete your volume, you must first detach it from the server.

To detach the volume from your server, pass the server ID and volume ID to the command, as follows:

```
$ nova volume-detach 84c6e57d-a6b1-44b6-81eb-fcb36afd31b5
573e024d-5235-49ce-8332-be1576d323f8
```

The **volume-detach** command does not return any output.

- List volumes:

```
$ cinder list
```

```

+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
|          ID          |          Status          |          Display Name          |
Size | Volume Type | Bootable | Attached to |
+-----+-----+-----+-----+-----+-----+
| 573e024d-5235-49ce-8332-be1576d323f8 | available | my-new-volume | 8
|          None          | true          |          |
| bd7cf584-45de-44e3-bf7f-f7b50bf235e3 | available | my-bootable-vol | 8
|          None          | true          |          |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+

```

Note that the volume is now available.

- Now you can delete the volume, as follows:

```
$ cinder delete my-new-volume
```

The delete command does not return any output.

10. List the volumes again, and note that the status of your volume is deleting:

```
$ cinder list
```

ID	Status	Display Name	Size	Volume Type	Bootable	Attached to
573e024d-5235-49ce-8332-be1576d323f8	deleting	my-new-volume	8	None	true	
bd7cf584-45de-44e3-bf7f-f7b50bf235e3	available	my-bootable-vol	8	None	true	

When the volume is fully deleted, it disappears from the list of volumes:

```
$ cinder list
```

ID	Status	Display Name	Size	Volume Type	Bootable	Attached to
bd7cf584-45de-44e3-bf7f-f7b50bf235e3	available	my-bootable-vol	8	None	true	

Appendix A. Command reference

keystone commands	60
glance commands	62
neutron commands	64
nova commands	67
cinder commands	71
swift commands	73
heat commands	74

keystone commands

The keystone client is the command-line interface (CLI) for the OpenStack Identity API.

For help on a specific keystone command, enter:

```
keystone help COMMAND
```

Example A.1. Usage

```
keystone [--version] [--timeout <seconds>]
  [--os-username <auth-user-name>]
  [--os-password <auth-password>]
  [--os-tenant-name <auth-tenant-name>]
  [--os-tenant-id <tenant-id>] [--os-auth-url <auth-url>]
  [--os-region-name <region-name>]
  [--os-identity-api-version <identity-api-version>]
  [--os-token <service-token>]
  [--os-endpoint <service-endpoint>]
  [--os-cacert <ca-certificate>] [--insecure]
  [--os-cert <certificate>] [--os-key <key>] [--os-cache]
  [--force-new-token] [--stale-duration <seconds>]
  <subcommand> ...
```

Example A.2. Positional arguments

```
<subcommand>
  catalog          List service catalog, possibly filtered by service.
  ec2-credentials-create
                    Create EC2-compatible credentials for user per tenant
  ec2-credentials-delete
                    Delete EC2-compatible credentials
  ec2-credentials-get
                    Display EC2-compatible credentials
  ec2-credentials-list
                    List EC2-compatible credentials for a user
  endpoint-create  Create a new endpoint associated with a service
  endpoint-delete  Delete a service endpoint
  endpoint-get     Find endpoint filtered by a specific attribute or
                    service type
  endpoint-list    List configured service endpoints
  password-update Update own password
  role-create      Create new role
  role-delete      Delete role
  role-get         Display role details
  role-list        List all roles
```


service-create	Add service to Service Catalog
service-delete	Delete service from Service Catalog
service-get	Display service from Service Catalog
service-list	List all services in Service Catalog
tenant-create	Create new tenant
tenant-delete	Delete tenant
tenant-get	Display tenant details
tenant-list	List all tenants
tenant-update	Update tenant name, description, enabled status
token-get	Display the current user token
user-create	Create new user
user-delete	Delete user
user-get	Display user details.
user-list	List users
user-password-update	Update user password
user-role-add	Add role to user
user-role-list	List roles granted to a user
user-role-remove	Remove role from user
user-update	Update user's name, email, and enabled status
discover	Discover Keystone servers, supported API versions and extensions.
bootstrap	Grants a new role to a new user on a new tenant, after creating each.
bash-completion	Prints all of the commands and options to stdout.
help	Display help about this program or one of its subcommands.

Example A.3. Optional arguments

--version	Shows the client version and exits
--timeout <seconds>	Set request timeout (in seconds)
--os-username <auth-user-name>	Name used for authentication with the OpenStack Identity service. Defaults to env[OS_USERNAME]
--os-password <auth-password>	Password used for authentication with the OpenStack Identity service. Defaults to env[OS_PASSWORD]
--os-tenant-name <auth-tenant-name>	Tenant to request authorization on. Defaults to env[OS_TENANT_NAME]
--os-tenant-id <tenant-id>	Tenant to request authorization on. Defaults to env[OS_TENANT_ID]
--os-auth-url <auth-url>	Specify the Identity endpoint to use for authentication. Defaults to env[OS_AUTH_URL]
--os-region-name <region-name>	Defaults to env[OS_REGION_NAME]
--os-identity-api-version <identity-api-version>	Defaults to env[OS_IDENTITY_API_VERSION] or 2.0
--os-token <service-token>	Specify an existing token to use instead of retrieving one via authentication (e.g. with username & password). Defaults to env[OS_SERVICE_TOKEN]
--os-endpoint <service-endpoint>	Specify an endpoint to use instead of retrieving one from the service catalog (via authentication). Defaults to env[OS_SERVICE_ENDPOINT]
--os-cacert <ca-certificate>	

```

Specify a CA bundle file to use in verifying a TLS
(https) server certificate. Defaults to env[OS_CACERT]
--insecure          Explicitly allow keystoneclient to perform "insecure"
                    TLS (https) requests. The server's certificate will
                    not be verified against any certificate authorities.
                    This option should be used with caution.
--os-cert <certificate>
                    Defaults to env[OS_CERT]
--os-key <key>      Defaults to env[OS_KEY]
--os-cache          Use the auth token cache. Defaults to env[OS_CACHE]
--force-new-token   If the keyring is available and in use, token will
                    always be stored and fetched from the keyring until
                    the token has expired. Use this option to request a
                    new token and replace the existing one in the keyring.
--stale-duration <seconds>
                    Stale duration (in seconds) used to determine whether
                    a token has expired when retrieving it from keyring.
                    This is useful in mitigating process or network
                    delays. Default is 30 seconds.

```

glance commands

The glance client is the command-line interface (CLI) for the OpenStack Image Service API.

For help on a specific glance command, enter:

```
glance help COMMAND
```

Example A.4. Usage

```

glance [--version] [-d] [-v] [-k] [--cert-file CERT_FILE]
  [--key-file KEY_FILE] [--os-cacert <ca-certificate-file>]
  [--ca-file OS_CACERT] [--timeout TIMEOUT] [--no-ssl-compression]
  [-f] [--dry-run] [--ssl] [-H ADDRESS] [-p PORT]
  [--os-username OS_USERNAME] [-I OS_USERNAME]
  [--os-password OS_PASSWORD] [-K OS_PASSWORD]
  [--os-tenant-id OS_TENANT_ID] [--os-tenant-name OS_TENANT_NAME]
  [-T OS_TENANT_NAME] [--os-auth-url OS_AUTH_URL] [-N OS_AUTH_URL]
  [--os-region-name OS_REGION_NAME] [-R OS_REGION_NAME]
  [--os-auth-token OS_AUTH_TOKEN] [-A OS_AUTH_TOKEN]
  [--os-image-url OS_IMAGE_URL] [-U OS_IMAGE_URL]
  [--os-image-api-version OS_IMAGE_API_VERSION]
  [--os-service-type OS_SERVICE_TYPE]
  [--os-endpoint-type OS_ENDPOINT_TYPE] [-S OS_AUTH_STRATEGY]
  <subcommand> ...

```

Example A.5. Positional arguments

```

<subcommand>
  add          DEPRECATED! Use image-create instead.
  clear        DEPRECATED!
  delete       DEPRECATED! Use image-delete instead.
  details      DEPRECATED! Use image-list instead.
  image-create Create a new image.
  image-delete Delete specified image(s).
  image-download Download a specific image.
  image-list   List images you can access.
  image-members DEPRECATED! Use member-list instead.
  image-show   Describe a specific image.

```

```

image-update      Update a specific image.
index             DEPRECATED! Use image-list instead.
member-add        DEPRECATED! Use member-create instead.
member-create     Share a specific image with a tenant.
member-delete     Remove a shared image from a tenant.
member-images     DEPRECATED! Use member-list instead.
member-list       Describe sharing permissions by image or tenant.
members-replace  DEPRECATED!
show              DEPRECATED! Use image-show instead.
update           DEPRECATED! Use image-update instead.
help             Display help about this program or one of its
                subcommands.

```

Example A.6. Optional arguments

```

--version          show program's version number and exit
-d, --debug        Defaults to env[GLANCECLIENT_DEBUG]
-v, --verbose      Print more verbose output
-k, --insecure     Explicitly allow glanceclient to perform "insecure
                  SSL" (https) requests. The server's certificate will
                  not be verified against any certificate authorities.
                  This option should be used with caution.

--cert-file CERT_FILE
                  Path of certificate file to use in SSL connection.
                  This file can optionally be prepended with the private
                  key.

--key-file KEY_FILE
                  Path of client key to use in SSL connection. This
                  option is not necessary if your key is prepended to
                  your cert file.

--os-cacert <ca-certificate-file>
                  Path of CA TLS certificate(s) used to verify the
                  remote server's certificate. Without this option
                  glance looks for the default system CA certificates.

--ca-file OS_CACERT
                  DEPRECATED! Use --os-cacert.
--timeout TIMEOUT
                  Number of seconds to wait for a response
--no-ssl-compression
                  Disable SSL compression when using https.
-f, --force        Prevent select actions from requesting user
                  confirmation.

--dry-run          DEPRECATED! Only used for deprecated legacy commands.
--ssl              DEPRECATED! Send a fully-formed endpoint using --os-
                  image-url instead.

-H ADDRESS, --host ADDRESS
                  DEPRECATED! Send a fully-formed endpoint using --os-
                  image-url instead.

-p PORT, --port PORT
                  DEPRECATED! Send a fully-formed endpoint using --os-
                  image-url instead.

--os-username OS_USERNAME
                  Defaults to env[OS_USERNAME]
-I OS_USERNAME     DEPRECATED! Use --os-username.
--os-password OS_PASSWORD
                  Defaults to env[OS_PASSWORD]
-K OS_PASSWORD     DEPRECATED! Use --os-password.
--os-tenant-id OS_TENANT_ID
                  Defaults to env[OS_TENANT_ID]
--os-tenant-name OS_TENANT_NAME
                  Defaults to env[OS_TENANT_NAME]
-T OS_TENANT_NAME  DEPRECATED! Use --os-tenant-name.
--os-auth-url OS_AUTH_URL
                  Defaults to env[OS_AUTH_URL]
-N OS_AUTH_URL     DEPRECATED! Use --os-auth-url.

```

```

--os-region-name OS_REGION_NAME
    Defaults to env[OS_REGION_NAME]
-R OS_REGION_NAME    DEPRECATED! Use --os-region-name.
--os-auth-token OS_AUTH_TOKEN
    Defaults to env[OS_AUTH_TOKEN]
-A OS_AUTH_TOKEN, --auth_token OS_AUTH_TOKEN
    DEPRECATED! Use --os-auth-token.
--os-image-url OS_IMAGE_URL
    Defaults to env[OS_IMAGE_URL]
-U OS_IMAGE_URL, --url OS_IMAGE_URL
    DEPRECATED! Use --os-image-url.
--os-image-api-version OS_IMAGE_API_VERSION
    Defaults to env[OS_IMAGE_API_VERSION] or 1
--os-service-type OS_SERVICE_TYPE
    Defaults to env[OS_SERVICE_TYPE]
--os-endpoint-type OS_ENDPOINT_TYPE
    Defaults to env[OS_ENDPOINT_TYPE]
-S OS_AUTH_STRATEGY, --os_auth_strategy OS_AUTH_STRATEGY
    DEPRECATED! This option is completely ignored.

```

neutron commands

The neutron client is the command-line interface (CLI) for the OpenStack Networking API.

For help on a specific neutron command, enter:

```
neutron help COMMAND
```

Example A.7. Usage

```

neutron [--version] [-v] [-q] [-h] [--debug]
    [--os-auth-strategy <auth-strategy>] [--os-auth-url <auth-url>]
    [--os-tenant-name <auth-tenant-name>]
    [--os-username <auth-username>] [--os-password <auth-password>]
    [--os-region-name <auth-region-name>] [--os-token <token>]
    [--endpoint-type <endpoint-type>] [--os-url <url>] [--insecure]

```

Example A.8. Optional arguments

```

--version          show program's version number and exit
-v, --verbose     Increase verbosity of output. Can be repeated.
-q, --quiet       suppress output except warnings and errors
-h, --help        show this help message and exit
--debug           show tracebacks on errors
--os-auth-strategy <auth-strategy>
                  Authentication strategy (Env: OS_AUTH_STRATEGY,
                  default keystone). For now, any other value will
                  disable the authentication
--os-auth-url <auth-url>
                  Authentication URL (Env: OS_AUTH_URL)
--os-tenant-name <auth-tenant-name>
                  Authentication tenant name (Env: OS_TENANT_NAME)
--os-username <auth-username>
                  Authentication username (Env: OS_USERNAME)
--os-password <auth-password>
                  Authentication password (Env: OS_PASSWORD)
--os-region-name <auth-region-name>
                  Authentication region name (Env: OS_REGION_NAME)
--os-token <token>
                  Defaults to env[OS_TOKEN]

```

```

--endpoint-type <endpoint-type>
    Defaults to env[OS_ENDPOINT_TYPE] or publicURL.
--os-url <url>
    Defaults to env[OS_URL]
--insecure
    Explicitly allow neutronclient to perform "insecure"
    SSL (https) requests. The server's certificate will
    not be verified against any certificate authorities.
    This option should be used with caution.

```

Example A.9. Commands

agent-delete	Delete a given agent.
agent-list	List agents.
agent-show	Show information of a given agent.
agent-update	Update a given agent.
dhcp-agent-list-hosting-net	List DHCP agents hosting a network.
dhcp-agent-network-add	Add a network to a DHCP agent.
dhcp-agent-network-remove	Remove a network from a DHCP agent.
ext-list	List all extensions.
ext-show	Show information of a given resource.
floatingip-associate	Create a mapping between a floating ip and a
fixed ip.	
floatingip-create	Create a floating ip for a given tenant.
floatingip-delete	Delete a given floating ip.
floatingip-disassociate	Remove a mapping from a floating ip to a
fixed ip.	
floatingip-list	List floating ips that belong to a given
tenant.	
floatingip-show	Show information of a given floating ip.
help	print detailed help for another command
l3-agent-list-hosting-router	List L3 agents hosting a router.
l3-agent-router-add	Add a router to a L3 agent.
l3-agent-router-remove	Remove a router from a L3 agent.
lb-healthmonitor-associate	Create a mapping between a health monitor and
a pool.	
lb-healthmonitor-create	Create a healthmonitor.
lb-healthmonitor-delete	Delete a given healthmonitor.
lb-healthmonitor-disassociate	Remove a mapping from a health monitor to a
pool.	
lb-healthmonitor-list	List healthmonitors that belong to a given
tenant.	
lb-healthmonitor-show	Show information of a given healthmonitor.
lb-healthmonitor-update	Update a given healthmonitor.
lb-member-create	Create a member.
lb-member-delete	Delete a given member.
lb-member-list	List members that belong to a given tenant.
lb-member-show	Show information of a given member.
lb-member-update	Update a given member.
lb-pool-create	Create a pool.
lb-pool-delete	Delete a given pool.
lb-pool-list	List pools that belong to a given tenant.
lb-pool-show	Show information of a given pool.
lb-pool-stats	Retrieve stats for a given pool.
lb-pool-update	Update a given pool.
lb-vip-create	Create a vip.
lb-vip-delete	Delete a given vip.
lb-vip-list	List vips that belong to a given tenant.
lb-vip-show	Show information of a given vip.
lb-vip-update	Update a given vip.
net-create	Create a network for a given tenant.
net-delete	Delete a given network.

net-external-list tenant.	List external networks that belong to a given tenant.
net-gateway-connect router.	Add an internal network interface to a router.
net-gateway-create	Create a network gateway.
net-gateway-delete	Delete a given network gateway.
net-gateway-disconnect	Remove a network from a network gateway.
net-gateway-list	List network gateways for a given tenant.
net-gateway-show	Show information of a given network gateway.
net-gateway-update	Update the name for a network gateway.
net-list	List networks that belong to a given tenant.
net-list-on-dhcp-agent	List the networks on a DHCP agent.
net-show	Show information of a given network.
net-update	Update network's information.
port-create	Create a port for a given tenant.
port-delete	Delete a given port.
port-list	List ports that belong to a given tenant.
port-show	Show information of a given port.
port-update	Update port's information.
queue-create	Create a queue.
queue-delete	Delete a given queue.
queue-list	List queues that belong to a given tenant.
queue-show	Show information of a given queue.
quota-delete	Delete defined quotas of a given tenant.
quota-list	List defined quotas of all tenants.
quota-show	Show quotas of a given tenant
quota-update	Define tenant's quotas not to use defaults.
router-create	Create a router for a given tenant.
router-delete	Delete a given router.
router-gateway-clear router.	Remove an external network gateway from a router.
router-gateway-set router.	Set the external network gateway for a router.
router-interface-add router.	Add an internal network interface to a router.
router-interface-delete router.	Remove an internal network interface from a router.
router-list	List routers that belong to a given tenant.
router-list-on-l3-agent	List the routers on a L3 agent.
router-port-list with specified router.	List ports that belong to a given tenant,
router-show	Show information of a given router.
router-update	Update router's information.
security-group-create	Create a security group.
security-group-delete	Delete a given security group.
security-group-list tenant.	List security groups that belong to a given tenant.
security-group-rule-create	Create a security group rule.
security-group-rule-delete	Delete a given security group rule.
security-group-rule-list given tenant.	List security group rules that belong to a given tenant.
security-group-rule-show rule.	Show information of a given security group rule.
security-group-show	Show information of a given security group.
security-group-update	Update a given security group.
subnet-create	Create a subnet for a given tenant.
subnet-delete	Delete a given subnet.
subnet-list	List networks that belong to a given tenant.
subnet-show	Show information of a given subnet.
subnet-update	Update subnet's information.

nova commands

The nova client is the command-line interface for the OpenStack Compute API and its extensions.

For help on a specific nova command, enter:

```
nova help COMMAND
```

Example A.10. Usage

```
nova [--version] [--debug] [--os-cache] [--timings]
      [--timeout <seconds>] [--os-username <auth-user-name>]
      [--os-password <auth-password>]
      [--os-tenant-name <auth-tenant-name>] [--os-auth-url <auth-url>]
      [--os-region-name <region-name>] [--os-auth-system <auth-system>]
      [--service-type <service-type>] [--service-name <service-name>]
      [--volume-service-name <volume-service-name>]
      [--endpoint-type <endpoint-type>]
      [--os-compute-api-version <compute-api-ver>]
      [--os-cacert <ca-certificate>] [--insecure]
      [--bypass-url <bypass-url>]
      <subcommand> ...
```

Example A.11. Positional arguments

```
<subcommand>
  absolute-limits      Print a list of absolute limits for a user
  add-fixed-ip         Add new IP address on a network to server.
  add-floating-ip     Add a floating IP address to a server.
  add-secgroup        Add a Security Group to a server.
  agent-create        Creates a new agent build.
  agent-delete        Deletes an existing agent build.
  agent-list          List all builds
  agent-modify        Modify an existing agent build.
  aggregate-add-host  Add the host to the specified aggregate.
  aggregate-create    Create a new aggregate with the specified details.
  aggregate-delete    Delete the aggregate by its id.
  aggregate-details   Show details of the specified aggregate.
  aggregate-list      Print a list of all aggregates.
  aggregate-remove-host
                      Remove the specified host from the specified
                      aggregate.
  aggregate-set-metadata
                      Update the metadata associated with the aggregate.
  aggregate-update    Update the aggregate's name and optionally
                      availability zone.
  availability-zone-list
                      List all the availability zones.
  backup              Backup a instance by create a
'backup' type snapshot
  boot                Boot a new server.
  clear-password      Clear password for a server.
  cloudpipe-configure
                      Update the VPN IP/port of a cloudpipe instance
  cloudpipe-create    Create a cloudpipe instance for the given project
  cloudpipe-list      Print a list of all cloudpipe instances.
  console-log         Get console log output of a server.
  coverage-report     Generate a coverage report
```

coverage-reset	Reset coverage data.
coverage-start	Start Nova coverage reporting
coverage-stop	Stop Nova coverage reporting
credentials	Show user credentials returned from auth
delete	Immediately shut down and delete specified server(s).
diagnostics	Retrieve server diagnostics.
dns-create	Create a DNS entry for domain, name and ip.
dns-create-private-domain	Create the specified DNS domain.
dns-create-public-domain	Create the specified DNS domain.
dns-delete	Delete the specified DNS entry.
dns-delete-domain	Delete the specified DNS domain.
dns-domains	Print a list of available dns domains.
dns-list	List current DNS entries for domain and ip or domain and name.
endpoints	Discover endpoints that get returned from the authenticate services
evacuate	Evacuate server from failed host to specified one.
fixed-ip-get	Get info on a fixed ip
fixed-ip-reserve	Reserve a fixed ip
fixed-ip-unreserve	Unreserve a fixed ip
flavor-access-add	Add flavor access for the given tenant.
flavor-access-list	Print access information about the given flavor.
flavor-access-remove	Remove flavor access for the given tenant.
flavor-create	Create a new flavor
flavor-delete	Delete a specific flavor
flavor-key	Set or unset extra_spec for a flavor.
flavor-list	Print a list of available 'flavors' (sizes of servers).
flavor-show	Show details about the given flavor.
floating-ip-bulk-create	Bulk create floating ips by range
floating-ip-bulk-delete	Bulk delete floating ips by range
floating-ip-bulk-list	List all floating ips
floating-ip-create	Allocate a floating IP for the current tenant.
floating-ip-delete	De-allocate a floating IP.
floating-ip-list	List floating ips for this tenant.
floating-ip-pool-list	List all floating ip pools.
get-password	Get password for a server.
get-spice-console	Get a spice console to a server.
get-vnc-console	Get a vnc console to a server.
host-action	Perform a power action on a host.
host-describe	Describe a specific host
host-list	List all hosts by service
host-update	Update host settings.
hypervisor-list	List hypervisors.
hypervisor-servers	List instances belonging to specific hypervisors.
hypervisor-show	Display the details of the specified hypervisor.
hypervisor-stats	Get hypervisor statistics over all compute nodes.
hypervisor-uptime	Display the uptime of the specified hypervisor.
image-create	Create a new image by taking a snapshot of a running server.
image-delete	Delete specified image(s).
image-list	Print a list of available images to boot from.
image-meta	Set or Delete metadata on an image.

image-show	Show details about the given image.
interface-attach	Attach a network interface to an instance.
interface-detach	Detach a network interface from an instance.
interface-list	List interfaces attached to an instance.
keypair-add	Create a new key pair for use with instances
keypair-delete	Delete keypair by its name
keypair-list	Print a list of keypairs for a user
list	List active servers.
live-migration	Migrates a running instance to a new machine.
lock	Lock a server.
meta	Set or Delete metadata on a server.
migrate	Migrate a server. The new host will be selected by the scheduler.
network-associate-host	Associate host with network.
network-associate-project	Associate project with network.
network-create	Create a network.
network-disassociate	Disassociate host and/or project from the given network.
network-list	Print a list of available networks.
network-show	Show details about the given network.
pause	Pause a server.
quota-class-show	List the quotas for a quota class.
quota-class-update	Update the quotas for a quota class.
quota-defaults	List the default quotas for a tenant.
quota-delete	Delete quota for a tenant so their quota will revert back to default.
quota-show	List the quotas for a tenant.
quota-update	Update the quotas for a tenant.
rate-limits	Print a list of rate limits for a user
reboot	Reboot a server.
rebuild	Shutdown, re-image, and re-boot a server.
remove-fixed-ip	Remove an IP address from a server.
remove-floating-ip	Remove a floating IP address from a server.
remove-secgroup	Remove a Security Group from a server.
rename	Rename a server.
rescue	Rescue a server.
reset-network	Reset network of an instance.
reset-state	Reset the state of an instance
resize	Resize a server.
resize-confirm	Confirm a previous resize.
resize-revert	Revert a previous resize (and return to the previous VM).
resume	Resume a server.
root-password	Change the root password for a server.
scrub	Deletes data associated with the project
secgroup-add-group-rule	Add a source group rule to a security group.
secgroup-add-rule	Add a rule to a security group.
secgroup-create	Create a security group.
secgroup-delete	Delete a security group.
secgroup-delete-group-rule	Delete a source group rule from a security group.
secgroup-delete-rule	Delete a rule from a security group.
secgroup-list	List security groups for the current tenant.
secgroup-list-rules	List rules for a security group.

secgroup-update	Update a security group.
service-disable	Disable the service
service-enable	Enable the service
service-list	Show a list of all running services. Filter by host & binary.
show	Show details about the given server.
ssh	SSH into a server.
start	Start a server.
stop	Stop a server.
suspend	Suspend a server.
unlock	Unlock a server.
unpause	Unpause a server.
unrescue	Unrescue a server.
usage	Show usage data for a single tenant
usage-list	List usage data for all tenants
volume-attach	Attach a volume to a server.
volume-create	Add a new volume.
volume-delete	Remove a volume.
volume-detach	Detach a volume from a server.
volume-list	List all the volumes.
volume-show	Show details about a volume.
volume-snapshot-create	Add a new snapshot.
volume-snapshot-delete	Remove a snapshot.
volume-snapshot-list	List all the snapshots.
volume-snapshot-show	Show details about a snapshot.
volume-type-create	Create a new volume type.
volume-type-delete	Delete a specific flavor.
volume-type-list	Print a list of available 'volume types'.
x509-create-cert	Create x509 cert for a user in tenant
x509-get-root-cert	Fetches the x509 root cert.
bash-completion	Prints all of the commands and options to stdout so that the
help	Display help about this program or one of its subcommands.
baremetal-interface-add	Add a network interface to a baremetal node.
baremetal-interface-list	List network interfaces associated with a baremetal node.
baremetal-interface-remove	Remove a network interface from a baremetal node.
baremetal-node-create	Create a baremetal node.
baremetal-node-delete	Remove a baremetal node and any associated interfaces.
baremetal-node-list	Print a list of available baremetal nodes.
baremetal-node-show	Show information about a baremetal node.
host-evacuate	Evacuate all instances from failed host to specified one.
instance-action	Show an action.
instance-action-list	List actions on a server.
list-extensions	List all the os-api extensions that are available.
host-meta	Set or Delete metadata on all instances of a host.

```
net          Show a network.
net-create  Create a network.
net-delete  Delete a network.
net-list    List networks
```

Example A.12. Optional arguments

```
--version      Show program's version number and exit.
--debug        Print debugging output.
--force        Force quota-update.
--os-cache     Use the auth token cache.
--timings      Print call timing info.
--timeout <seconds> Set HTTP call timeout (in seconds).
--os-username <auth-user-name>
                Defaults to env[OS_USERNAME].
--os-password <auth-password>
                Defaults to env[OS_PASSWORD].
--os-tenant-name <auth-tenant-name>
                Defaults to env[OS_TENANT_NAME].
--os-auth-url <auth-url>
                Defaults to env[OS_AUTH_URL].
--os-region-name <region-name>
                Defaults to env[OS_REGION_NAME].
--os-auth-system <auth-system>
                Defaults to env[OS_AUTH_SYSTEM].
--service-type <service-type>
                Defaults to compute for most actions.
--service-name <service-name>
                Defaults to env[NOVA_SERVICE_NAME].
--volume-service-name <volume-service-name>
                Defaults to env[NOVA_VOLUME_SERVICE_NAME].
--endpoint-type <endpoint-type>
                Defaults to env[NOVA_ENDPOINT_TYPE] or publicURL.
--os-compute-api-version <compute-api-ver>
                Accepts 1.1, defaults to env[OS_COMPUTE_API_VERSION].
--os-cacert <ca-certificate>
                Specify a CA bundle file to use in verifying a TLS
                (https) server certificate. Defaults to env[OS_CACERT]
--insecure     Explicitly allow novaclient to perform "insecure" SSL
                (https) requests. The server's certificate will not be
                verified against any certificate authorities. This
                option should be used with caution.
--bypass-url <bypass-url>
                Use this API endpoint instead of the Service Catalog.
```

cinder commands

The cinder client is the command-line interface for the OpenStack Identity API.

For help on a specific cinder command, enter:

```
cinder help COMMAND
```

Example A.13. Usage

```
cinder [--version] [--debug] [--os-username <auth-user-name>]
        [--os-password <auth-password>]
        [--os-tenant-name <auth-tenant-name>]
```

```

[--os-tenant-id <auth-tenant-id>] [--os-auth-url <auth-url>]
[--os-region-name <region-name>] [--service-type <service-type>]
[--service-name <service-name>]
[--volume-service-name <volume-service-name>]
[--endpoint-type <endpoint-type>]
[--os-volume-api-version <compute-api-ver>]
[--os-cacert <ca-certificate>] [--retries <retries>]
<subcommand> ...

```

Example A.14. Positional arguments

<subcommand>	
absolute-limits	Print a list of absolute limits for a user
backup-create	Creates a backup.
backup-delete	Remove a backup.
backup-list	List all the backups.
backup-restore	Restore a backup.
backup-show	Show details about a backup.
create	Add a new volume.
credentials	Show user credentials returned from auth.
delete	Remove a volume.
endpoints	Discover endpoints that get returned from the authenticate services.
extra-specs-list	Print a list of current 'volume types and extra specs'
force-delete	(Admin Only). Attempt forced removal of a volume, regardless of its
list	state.
list	List all the volumes.
metadata	Set or Delete metadata on a volume.
quota-class-show	List the quotas for a quota class.
quota-class-update	Update the default quotas for a quota class.
quota-defaults	List the default quotas for a tenant.
quota-show	List the quotas for a tenant.
quota-update	Update the quotas for a tenant.
rate-limits	Print a list of rate limits for a user
rename	Rename a volume.
show	Show details about a volume.
snapshot-create	Add a new snapshot.
snapshot-delete	Remove a snapshot.
snapshot-list	List all the snapshots.
snapshot-rename	Rename a snapshot.
snapshot-show	Show details about a snapshot.
type-create	Create a new volume type.
type-delete	Delete a specific volume type.
type-key	Set or unset extra_spec for a volume type.
type-list	Print a list of available 'volume types'.
upload-to-image	Upload volume to image service as image.
bash-completion	Print arguments for bash_completion.
help	Display help about this program or one of its subcommands.
list-extensions	List all the os-api extensions that are available.

Example A.15. Optional arguments

```

--version          show program's version number and exit
--debug           Print debugging output
--os-username <auth-user-name>

```

```

        Defaults to env[OS_USERNAME].
--os-password <auth-password>
        Defaults to env[OS_PASSWORD].
--os-tenant-name <auth-tenant-name>
        Defaults to env[OS_TENANT_NAME].
--os-tenant-id <auth-tenant-id>
        Defaults to env[OS_TENANT_ID].
--os-auth-url <auth-url>
        Defaults to env[OS_AUTH_URL].
--os-region-name <region-name>
        Defaults to env[OS_REGION_NAME].
--service-type <service-type>
        Defaults to compute for most actions
--service-name <service-name>
        Defaults to env[CINDER_SERVICE_NAME]
--volume-service-name <volume-service-name>
        Defaults to env[CINDER_VOLUME_SERVICE_NAME]
--endpoint-type <endpoint-type>
        Defaults to env[CINDER_ENDPOINT_TYPE] or publicURL.
--os-volume-api-version <compute-api-ver>
        Accepts 1 or 2,defaults to env[OS_VOLUME_API_VERSION].
--os-cacert <ca-certificate>
        Specify a CA bundle file to use in verifying a TLS
        (https) server certificate. Defaults to env[OS_CACERT]
--retries <retries>      Number of retries.

```

swift commands

The swift client is the command-line interface for the OpenStack Object Storage API.

For help on a specific swift command, enter:

```
swift help COMMAND
```

Example A.16. Usage

```
swift COMMAND [options] [args]
```

Example A.17. Commands

```

stat [container] [object]
    Displays information for the account, container, or object depending on
    the
    args given (if any).

list [options] [container]
    Lists the containers for the account or the objects for a container. -p or
    --prefix is an option that will only list items beginning with that
    prefix.
    -l produces output formatted like 'ls -l' and --lh like 'ls -lh'.
    -d or --delimiter is option (for container listings only) that will roll
    up
    items with the given delimiter (see http://docs.openstack.org/
    api/openstack-object-storage/1.0/content/list-objects.html)

upload [options] container file_or_directory [file_or_directory] [...]
    Uploads to the given container the files and directories specified by the
    remaining args. -c or --changed is an option that will only upload files

```

```

that have changed since the last upload. -S <size> or --segment-size
<size>
will upload the files in segments no larger than size. -C <container> or
--segment-container <container> will specify the location of the segments
to <container>. --leave-segments are options as well (see --help for
more).

post [options] [container] [object]
Updates meta information for the account, container, or object depending
on
the args given. If the container is not found, it will be created
automatically; but this is not true for accounts and objects. Containers
also allow the -r (or --read-acl) and -w (or --write-acl) options. The -m
or --meta option is allowed on all and used to define the user meta data
items to set in the form Name:Value. This option can be repeated. Example:
post -m Color:Blue -m Size:Large

download --all OR download container [options] [object] [object] ...
Downloads everything in the account (with --all), or everything in a
container, or a list of objects depending on the args given. For a single
object download, you may use the -o [--output] <filename> option to
redirect the output to a specific file or if "-" then just redirect to
stdout.

delete [options] --all OR delete container [options] [object] [object] ...
Deletes everything in the account (with --all), or everything in a
container, or a list of objects depending on the args given. Segments of
manifest objects will be deleted as well, unless you specify the
--leave-segments option.

```

Example A.18. Examples

```

$ swift -A https://auth.api.rackspacecloud.com/v1.0 -U user -K key stat

$ swift --os-auth-url https://api.example.com/v2.0 --os-tenant-name tenant \
--os-usernameuser --os-password password list

$ swift --os-auth-token 6ee5eb33efad4e45ab46806eac010566 \
--os-storage-url https://10.1.5.2:8080/v1/AUTH_ced809b6a4baea7aeab61a \
list

$ swift list --lh

```

heat commands

The heat client is the command-line interface for the OpenStack Orchestration API.

For help on a specific heat command, enter:

```
heat help COMMAND
```

Example A.19. Usage

```

heat [-d] [-v] [-k] [--cert-file CERT_FILE] [--key-file KEY_FILE]
[--ca-file CA_FILE] [--timeout TIMEOUT]
[--os-username OS_USERNAME] [--os-password OS_PASSWORD]
[--os-tenant-id OS_TENANT_ID] [--os-tenant-name OS_TENANT_NAME]
[--os-auth-url OS_AUTH_URL] [--os-region-name OS_REGION_NAME]
[--os-auth-token OS_AUTH_TOKEN] [--os-no-client-auth]

```

```

[--heat-url HEAT_URL] [--heat-api-version HEAT_API_VERSION]
[--os-service-type OS_SERVICE_TYPE]
[--os-endpoint-type OS_ENDPOINT_TYPE] [-t]
<subcommand> ...

```

Example A.20. Positional arguments

```

<subcommand>
  create           DEPRECATED! Use stack-create instead
  delete          DEPRECATED! Use stack-delete instead
  describe        DEPRECATED! Use stack-show instead
  event           DEPRECATED! Use event-show instead
  event-list      List events for a stack
  event-show      Describe the event
  gettemplate     DEPRECATED! Use template-show instead
  list            DEPRECATED! Use stack-list instead
  resource        DEPRECATED! Use resource-show instead
  resource-list   Show list of resources belonging to a stack
  resource-metadata List resource metadata
  resource-show   Describe the resource
  stack-create    Create the stack
  stack-delete    Delete the stack
  stack-list      List the user's stacks
  stack-show      Describe the stack
  stack-update    Update the stack
  template-show   Get the template for the specified stack
  template-validate Validate a template with parameters
  update          DEPRECATED! Use stack-update instead
  validate        DEPRECATED! Use template-validate instead
  help            Display help about this program or one of its
                  subcommands.

```

Example A.21. Optional arguments

```

-d, --debug           Defaults to env[HEATCLIENT_DEBUG]
-v, --verbose         Print more verbose output
-k, --insecure        Explicitly allow the client to perform "insecure" SSL
                      (https) requests. The server's certificate will not be
                      verified against any certificate authorities. This
                      option should be used with caution.

--cert-file CERT_FILE Path of certificate file to use in SSL connection.
                      This file can optionally be prepended with the private
                      key.

--key-file KEY_FILE   Path of client key to use in SSL connection. This
                      option is not necessary if your key is prepended to
                      your cert file.

--ca-file CA_FILE     Path of CA SSL certificate(s) used to verify the
                      remote server's certificate. Without this option the
                      client looks for the default system CA certificates.

--timeout TIMEOUT    Number of seconds to wait for a response

--os-username OS_USERNAME
                      Defaults to env[OS_USERNAME]

--os-password OS_PASSWORD
                      Defaults to env[OS_PASSWORD]

--os-tenant-id OS_TENANT_ID
                      Defaults to env[OS_TENANT_ID]

--os-tenant-name OS_TENANT_NAME
                      Defaults to env[OS_TENANT_NAME]

--os-auth-url OS_AUTH_URL

```

```

        Defaults to env[OS_AUTH_URL]
--os-region-name OS_REGION_NAME
        Defaults to env[OS_REGION_NAME]
--os-auth-token OS_AUTH_TOKEN
        Defaults to env[OS_AUTH_TOKEN]
--os-no-client-auth Do not contact heat for a token. Defaults to
        env[OS_NO_CLIENT_AUTH]
--heat-url HEAT_URL Defaults to env[HEAT_URL]
--heat-api-version HEAT_API_VERSION
        Defaults to env[HEAT_API_VERSION] or 1
--os-service-type OS_SERVICE_TYPE
        Defaults to env[OS_SERVICE_TYPE]
--os-endpoint-type OS_ENDPOINT_TYPE
        Defaults to env[OS_ENDPOINT_TYPE]
-t, --token-only Only send a token for auth, do not send username and
        password as well.
```


. Resources

For the available OpenStack documentation, see docs.openstack.org.

The following books are of interest to administrators:

- *[Block Storage Service Administration Guide](#)*
- *[Compute Administration Guide](#)*
- *[High Availability Guide](#)*
- *[Networking Administration Guide](#)*
- *[Object Storage Administration Guide](#)*
- *[Operations Guide](#)*
- *[Security Guide](#)*
- *[Virtual Machine Image Guide](#)*

For assistance with OpenStack, go to ask.openstack.org.

To provide feedback on documentation, join and use the [<openstack-docs@lists.openstack.org>](mailto:openstack-docs@lists.openstack.org) mailing list at [OpenStack Documentation Mailing List](#), or [report a bug](#).