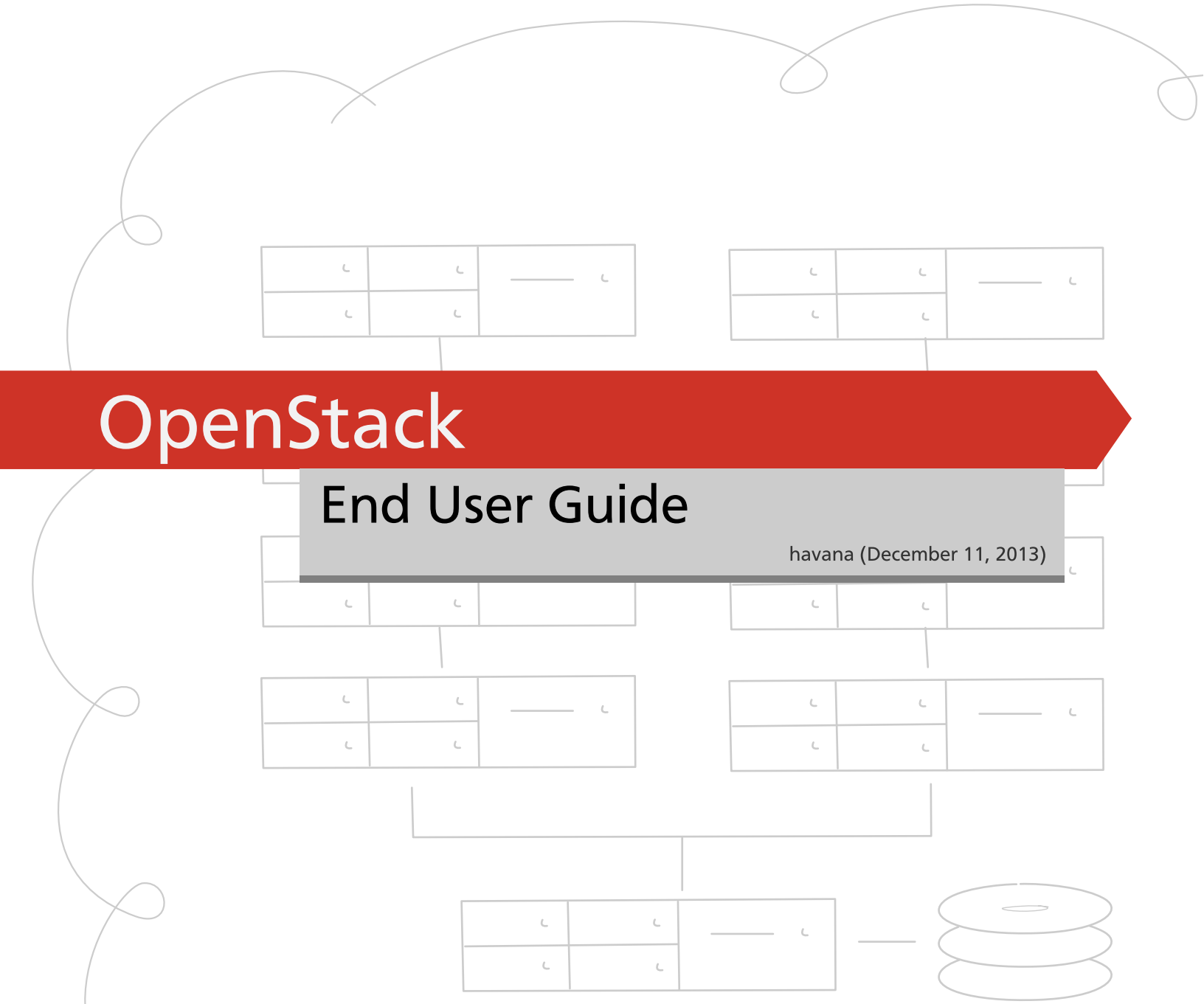
OpenStack

End User Guide

havana (December 11, 2013)



openstack™

docs.openstack.org

OpenStack End User Guide

havana (2013-12-11)

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 and OpenStack client commands.

Table of Contents

How can I use an OpenStack cloud? 6
Document change history 6
1. Dashboard 1
Log in to the dashboard 2
Create and manage images 6
Configure access and security for instances 8
Launch and manage instances 10
Create a network 14
Manage volumes 15
Launch and manage stacks 17
2. OpenStack command-line clients 19
Overview 19
Install the OpenStack command-line clients 21
Get the version for a client 23
Create and source the OpenStack RC file 24
Manage images 25
Configure access and security for instances 31
Launch instances 35
Manage instances and hosts 44
Configure instances at boot 55
Provide user data to instances 59
Store metadata on a configuration drive 60
Create and manage networks 65
Manage objects and containers 69
Create and manage stacks 70
Measure cloud resources 73
Manage volumes 76
A. Command reference 84
keystone commands 84
glance commands 86
neutron commands 88
nova commands 92
cinder commands 97
swift commands 99
heat commands 100
ceilometer commands 101
B. Community support 104
Documentation 104
ask.openstack.org 105
OpenStack mailing lists 105
The OpenStack wiki 105
The Launchpad Bugs area 106
The OpenStack IRC channel 106
Documentation feedback 107
OpenStack distribution packages 107

List of Tables

2.1. Prerequisite software 21
2.2. Disk and CD-ROM bus model values 28
2.3. VIF model values 28
2.4. Description of configuration options for configdrive 64

List of Examples

A.1. Usage 84
A.2. Positional arguments 84
A.3. Optional arguments 85
A.4. Usage 86
A.5. Positional arguments 86
A.6. Optional arguments 87
A.7. Usage 88
A.8. Positional arguments 88
A.9. Optional arguments 91
A.10. Usage 92
A.11. Positional arguments 92
A.12. Optional arguments 96
A.13. Usage 97
A.14. Positional arguments 97
A.15. Optional arguments 98
A.16. Usage 99
A.17. Commands 99
A.18. Examples 99
A.19. Usage 100
A.20. Positional arguments 100
A.21. Optional arguments 100
A.22. Usage 101
A.23. Positional arguments 102
A.24. Optional arguments 102

How can I use an OpenStack cloud?

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
October 17, 2013	• Havana release.
August 19, 2013	• Editorial changes.
July 29, 2013	• First edition of this document.

1. Dashboard

Table of Contents

Log in to the dashboard 2
Create and manage images 6
Configure access and security for instances 8
Launch and manage instances 10
Create a network 14
Manage volumes 15
Launch and manage stacks 17

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 for the dashboard:

   ```
   $ 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.

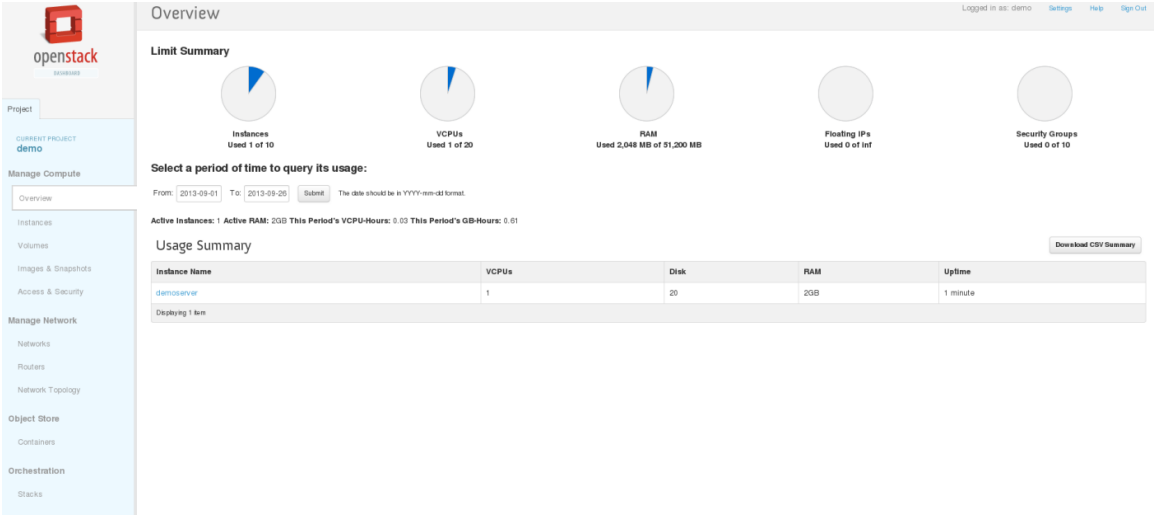
   If you are logged in as an end user, the main screen shows the Project tab.

   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 **CURRENT PROJECT** drop-down list on the left side to view and manage resources in that project.

The **Project** tab displays the details of the selected project.



Access the following tabs:

Manage Compute tab

Overview	View reports for the project.
Instances	View, launch, create a snapshot from, stop, pause, or reboot instances, or connect to them through VNC.
Volumes	View, create, edit, and delete volumes.
Images & Snapshots	View images, instance snapshots, and volume snapshots created project users, plus any images that are publicly available. Create, edit, and delete images, and launch instances from images and snapshots.
Access & Security	Use these tabs to complete these tasks: **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.

Manage Network tab

Networks	Create and manage public and private networks.
Routers	Create and manage subnets.
Network Topology	View the network topology.

Object Store tab

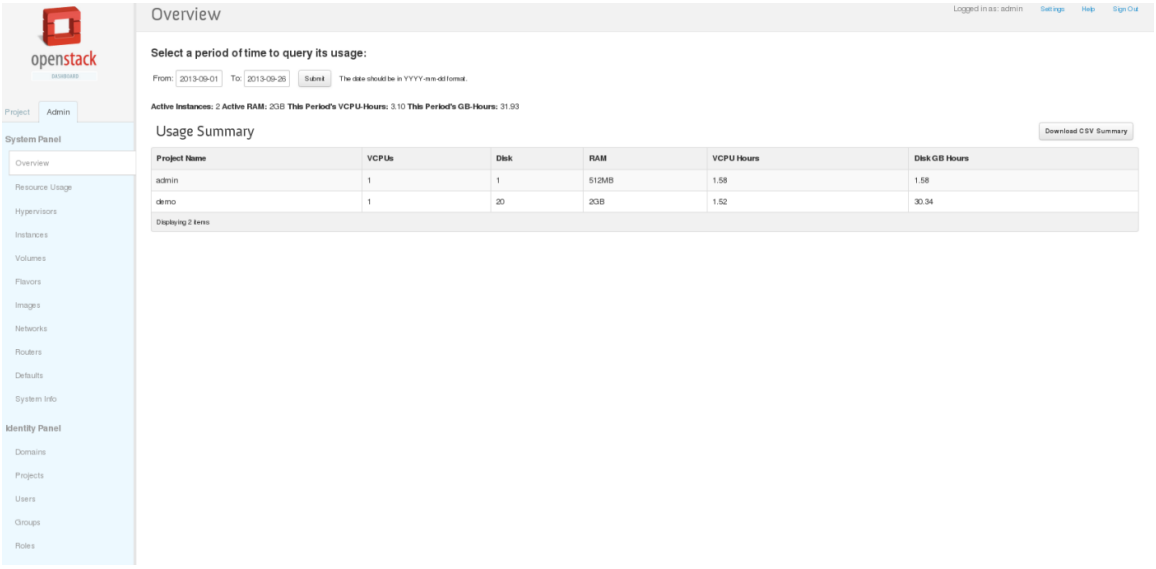
Containers Create and manage object storage.

Orchestration tab

Stacks Use REST API to orchestrate multiple composite cloud applications

OpenStack dashboard—Admin tab

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



Access the following categories to complete these tasks:

Overview View basic reports.

Resource Usage Use these tabs to view the following usages:

Global Disk Usage tab. View the global disk usage for all tenants as an average over the last 30 days.

Global Network Traffic Usage tab. View the global network usage for all tenants as an average over the last 30 days.

Global Object Storage Usage tab. View the global object storage usage for all tenants as an average over the last 30 days.

Global Network Usage tab. View the global network usage for all tenants as an average over the last 30 days.

Stats tab. View the statistics of all resources.

Hypervisors View the hypervisor summary.

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.
Networks	View, create, edit properties for, and delete networks.
Routers	View, create, edit properties for, and delete routers.
Defaults	View default quota values. Quotas are hard-coded in OpenStack Compute and define the maximum allowable size and number of resources.
System Info	Use these tabs to view the service info: **Services tab.** View a list of the services. **Compute Services tab.** View a list of all Compute services. **Availability Zones tab.** View the availability zones. **Host Aggregates tab.** View host aggregates. **Network Agents tab.** View the network agents.
Domains	View domains.
Projects	View, create, assign users to, remove users from, and delete projects.
Users	View, create, enable, disable, and delete users.
Groups	View, create, enable, disable, and delete groups.
Roles	View, create, enable, disable, and delete roles.

Create and manage 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 you have admin privileges, you can create and manage images in the admin project. 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" [25].

Create images

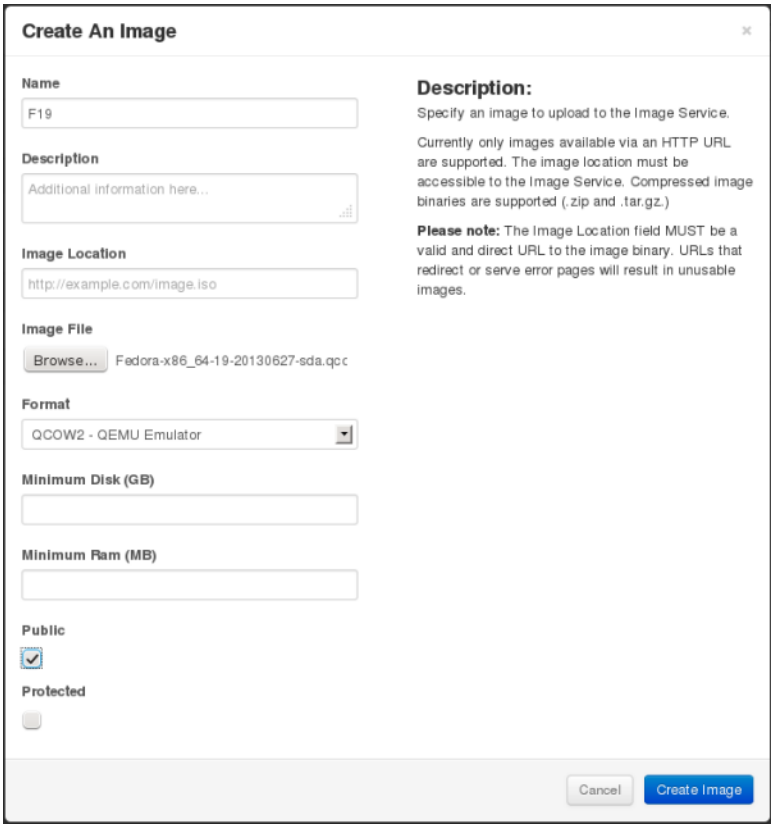
For details about image creation, see the *Virtual Machine Image Guide*.

1. Log in to the dashboard.

   Choose the **admin** project from the **CURRENT PROJECT** drop-down list.

2. On the **Project** tab, click the **Images & Snapshots** category.

3. Click **Create Image**. The **Create An Image** window appears:



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

Name	Enter a name for the image.
Description	Enter a brief description about the image.
Image 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 that only users with permissions can delete it.

5. Click **Create Image**.

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

Update images

1. Log in to the dashboard.

   Choose the **admin** project from the **CURRENT PROJECT** drop-down list.

2. On the **Project** tab, click the **Images & Snapshots** category.

3. Select the image that you want to edit. In the **More** drop-down list, click **Edit**.

4. In the **Update Image** window, you can change the name for the image. Select the **Public** check box to make the image public. Clear this check box to make the image private.

5. Click **Update Image**.

Delete images

1. Log in to the dashboard.

   Choose the **admin** project from the **CURRENT PROJECT** drop-down list.

2. On the **Project** tab, click the **Images & Snapshots** category.

3. Select the images that you want to delete.

4. Click **Delete Images**.

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

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" [9].

Add rules to the default security group

1. Log in to the dashboard, choose a project, and click the **Access & 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 allow ssh access, click **Add Rule**.
4. In the **Add Rule** window, enter the following values:

Rule	SSH
Remote	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**.

   The ssh 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:

Rule	All ICMP
Direction	Ingress
Remote	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 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 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 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" [8]. 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" [12].
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 access an instance" [49].

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” [11].

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 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 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" [11].

Manage an instance

1. Log in to the 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.

   You can resize or rebuild an instance. You can also choose to view the instance console log. Depending on the current state of the instance, you can choose to pause, resume, suspend, soft or hard reboot, or terminate an instance.

Create a network

1. Log in to the dashboard, choose a project, and click the **Networks** category.
2. Click **Create Network**.
3. In the Create Network window, specify the following values.

Network tab	
Network Name	A name to identify the network.
Subnet tab	
Create Subnet	Check this option to create a subnet You do not have to initially specify a subnet (although this will result in the status of 'error' for any attached instance).
Subnet Name	Name for the subnet.
Network Address	IP address for the subnet.
IP Version	IPv4 or IPv6.
Gateway IP	IP address for a specific gateway. This parameter is optional.
Disable Gateway	Check this option to disable gateway IP address.
Subnet Detail tab	
Enable DHCP	Check this option to enable DHCP
Allocation Pools	You can allocate IP address pools.
DNS Name Servers	You can allocate a name for the DNS server.
Host Routes	Include IP address of host routes.

4. Click **Create** to create a network.
5. The dashboard shows the network in the **Networks** category.

Manage volumes

Volumes are block storage devices that you attach to instances to enable persistent storage. You can attach a volume to a running instance or detach a volume and attach it to another instance at any time. You can also create a snapshot from or delete a volume. Only administrative users can create volume types.

Create a volume

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

   In the window that opens, enter a name, an optional description, and the size in GBs for the volume.
3. Click **Create Volume** to 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 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 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.

Create volume snapshots

1. Log in to the dashboard, choose a project, and click the **Instances & Volumes** category.
2. Select a volume from which to create a snapshot.
3. From the **Actions** drop-down list, select **Create Snapshot**.
4. In the window that opens, enter a snapshot name and a description.
5. Confirm your changes.

   The dashboard shows the new volume snapshot in the **Images & Snapshots** category.

Delete volumes

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

1. Log in to the 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.

Launch and manage stacks

Heat is a service used to orchestrate multiple composite cloud applications using the AWS CloudFormation template format, through both an OpenStack-native ReST API and a CloudFormation-compatible Query API.

The purpose of Heat is to provide a template based orchestration for describing a cloud application by executing appropriate OpenStack API calls to generate running cloud applications, integrate other core components of OpenStack into a one-file template system. The templates allow creation of most OpenStack resource types such as instances, floating ips, volumes, security groups, users, etc.

Launch a stack

1. Log in to the dashboard, choose a project, and click **Stacks** in the **Orchestration** category.
2. Click **Launch Stack**.
3. In the **Select Template** window, choose a template source option, **URL**, **File**, or **Direct Input**, from the drop-down list.
4. Enter the URL, browse to the file location, or directly include the template based on your previous selection.
5. In the **Launch Stack** window, specify the following values.

Stack Name	A name to identify the stack.
Creation Timeout (minutes)	Creation Timeout in minutes.
Rollback On Failure	Check this option if you want Heat to rollback on failure.
Password for user "demo"	Password for the user logged in.
DBUsername	Database user name.
LinuxDistribution	Linux Distribution used in the stacks.
DBRootPassword	Database root password.
KeyName	Name of the keypair.
DBName	Database name.
DBPassword	Database password.
InstanceType	Flavor of the instance.

6. Click **Launch** to create a stack.
7. The dashboard shows the stack in the **Stacks** category.

After the stack is created, click on the stack name to see these details:

Topology The topology of the stack created.

Overview The parameters and details of the stack under the following headings: **Info**, **Status**, **Outputs**, **Stack Parameters**, and **Launch Parameters**.

Resources The resources used by the stack.

Events The events related to the stack.

Delete stacks

1. Log in to the dashboard.
2. On the **Project** tab, click the **Stacks** category.
3. Select the stack that you want to delete.
4. Click **Delete Stack**.
5. In the **Confirm Delete Stack** window, click **Delete Stack** to confirm the deletion. You cannot undo this action.

2. OpenStack command-line clients

Table of Contents

Overview 19
Install the OpenStack command-line clients 21
Get the version for a client 23
Create and source the OpenStack RC file 24
Manage images 25
Configure access and security for instances 31
Launch instances 35
Manage instances and hosts 44
Configure instances at boot 55
Provide user data to instances 59
Store metadata on a configuration drive 60
Create and manage networks 65
Manage objects and containers 69
Create and manage stacks 70
Measure cloud resources 73
Manage volumes 76

Overview

You can use the OpenStack command-line clients to run simple commands that make API calls. You can use these commands in scripts to automate tasks. 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.

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

- **ceilometer** (python-ceilometerclient). Client for the Telemetry API that lets you create and collect measurements across OpenStack.
- **cinder** (python-cinderclient). Client for the Block Storage Service API that lets you create and manage volumes.
- **glance** (python-glanceclient). Client for the Image Service API that lets you create and manage images.
- **heat** (python-heatclient). Client for the Orchestration API that lets you launch stacks from templates, view details of running stacks including events and resources, and update and delete stacks.

- **keystone** (python-keystoneclient). Client for the Identity Service API that lets you create and manage users, tenants, roles, endpoints, and credentials.
- **neutron** (python-neutronclient). Client for the Networking API that lets you configure networks for guest servers. This client was previously known as **quantum**.
- **nova** (python-novaclient). Client for the Compute API and its extensions. Use to create and manage images, instances, and flavors.
- **swift** (python-swiftclient). Client for the Object Storage API that lets you 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.

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

Install the OpenStack command-line clients

Install the prerequisite software and the Python package for each OpenStack client.

Note

For each command, replace *PROJECT* with the lower case name of the client to install, such as nova. Repeat for each client.

Table 2.1. Prerequisite software

Prerequisite	Description
Python 2.6 or newer	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	To install the clients on a Mac OS X or Linux system, use pip. It is easy to use, ensures that you get the latest version of the clients from the Python Package Index, and lets you update or remove the packages later on.
	
	Install pip through the package manager for your system:
	
	Mac OS X.
	
	`$ sudo easy_install pip`
	
	Ubuntu 12.04. A packaged version enables you to use dpkg or aptitude to install the python-novaclient:
	
	`# aptitude install python-novaclient`
	
	Ubuntu.
	
	`# aptitude install python-pip`
	
	RHEL, CentOS, or Fedora. A packaged version available in RDO enables you to use yum to install the clients:
	
	`# yum install python-PROJECTclient`
	
	Alternatively, install pip and use it to manage client installation:
	
	`# yum install python-pip`
	
	openSUSE 12.2 and earlier. A packaged version available in the Open Build Service enables you to use rpm or zypper to install the python-novaclient:
	
	`# zypper install python-PROJECT`
	
	Alternatively, install pip and use it to manage client installation:
	
	`# zypper install python-pip`
	
	openSUSE 12.3 and newer. A packaged version enables you to use rpm or zypper to install the clients:
	
	`# zypper install python-PROJECTclient`

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 this command to install or update a client package:

   ```
   $ sudo pip install [--upgrade] 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.
   - `ceilometer`. Telemetry API.

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

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

   To update the nova client, run this command:

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

   To remove the nova client, run this command:

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

2. Before you can run client commands, you must create and source the `openrc` file to set environment variables. See the section called “Create and source the OpenStack RC file” [24].

Get the version for a client

After you install an OpenStack client, you can get its version number.

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

```
$ PROJECT --version
```

Where *PROJECT* is one of the following project names:

- `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.
- `ceilometer`. Telemetry 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
```

Create and source the OpenStack RC file

To set the required environment variables for the OpenStack command-line clients, you must either create or download an environment file, and source it. It is project-specific and contains the credentials used by all OpenStack services.

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

If your OpenStack installation provides it, 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, you 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 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.

Alternatively, you can create the `openrc.sh` file from scratch.

1. Create the `openrc.sh` file and add the authentication information:

   ```
   export OS_USERNAME=USERNAME
   export OS_PASSWORD=PASSWORD
   export OS_TENANT_NAME=PROJECT_NAME
   export OS_AUTH_URL=https://IDENTITY_HOST:PORT/v2.0
   # The following lines can be omitted
   export OS_TENANT_ID=9d792532ffce494583138c495801d164
   export OS_REGION_NAME=RegionOne
   ```
2. On any shell from where you want to run OpenStack commands, source the `openrc.sh` file for the respective project.

```
$ source openrc.sh
```

Note

You are not prompted for the password with this method. The password lives in clear text format in the `openrc.sh` file. Restrict the permissions on this file to avoid security problems. You can also remove the `OS_PASSWORD` variable from the file, and use the *`--password`* parameter with OpenStack client commands.

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)

1. 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 |
```

2. 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                          | 8e4838effa1969ad591655d6485c7ba8     |
| container_format                  | ami                                  |
| created_at                        | 2013-07-22T19:45:58                  |
| deleted                           | False                                |
| disk_format                       | ami                                  |
| id                                | 7e5142af-1253-4634-bcc6-89482c5f2e8a |
| is_public                         | False                                |
| min_disk                          | 0                                    |
| min_ram                           | 0                                    |
| name                              | myCirrosImage                        |
| owner                             | 66265572db174a7aa66eba661f58eb9e     |
| protected                         | False                                |
| size                              | 14221312                             |
| status                            | active                               |
| updated_at                        | 2013-07-22T19:46:42                  |
+-----------------------------------+--------------------------------------+
```

Note

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

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

```
[{
    "id": "b9d69795-5951-4cb0-bb5c-29491e1e2daf",
    "mountpoint": "/var/lib/glance/images/"
}]
```

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

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

For example:

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

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 --file=./centos63.qcow2
```

2. To update an image by name or ID:

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

To modify image properties, use the following optional arguments:

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

3. To annotate an image with a property that describes the `disk_bus`, `cdrom_bus`, and `vif_model`:

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

Currently libvirt will determine the disk/cdrom/vif device models based on the configured hypervisor type (`libvirt_type` in `/etc/nova/nova.conf`). For the sake of optimal performance, it will default to using virtio for both disk and VIF (NIC) models. The downside of this approach is that it is not possible to run operating systems that lack virtio drivers, for example, BSD, Solaris, old Linux, and old Windows.

If you specify a disk or CD-ROM bus model that is not supported, see Table 2.2, “Disk and CD-ROM bus model values” [28]. If you specify a VIF model that is not supported, the instance fails to launch. See Table 2.3, “VIF model values” [28].

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

Table 2.2. Disk and CD-ROM bus model values

libvirt_type setting	Supported model values
qemu or kvm	• virtio • scsi • ide • virtio
xen	• xen • ide

Table 2.3. VIF model values

libvirt_type setting	Supported model values
qemu or kvm	• virtio • ne2k_pci • pcnet • rtl8139 • e1000
xen	• 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.

4. 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                   | [{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                           | 2013-07-22T19:58:33Z                 |
| 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” [44].

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 "Create and source the OpenStack RC file" [24].

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 [33]:

```
$ 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 [33]:

```
$ 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 "Create and source the OpenStack RC file" [24].

2. 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         | True      |
| 2  | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3  | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4  | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
| 42 | m1.nano   | 64        | 0    | 0         |      | 1     | 1.0         | True      |
| 5  | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
| 84 | m1.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

1. 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 can launch an instance from that volume.

Optionally, to configure your volume, see the *OpenStack Configuration Reference*.

1. For a list of images to choose from to create a bootable volume, run this command:

```
$ nova image-list
+--------------------------------------+---------------------------------
+--------+--------+
| ID                                   | Name                            |
 Status | Server |
+--------------------------------------+---------------------------------
+--------+--------+
| e0b7734d-2331-42a3-b19e-067adc0da17d | cirros-0.3.1-x86_64-uec         |
 ACTIVE |        |
| 75bf193b-237b-435e-8712-896c51484de9 | cirros-0.3.1-x86_64-uec-kernel  |
 ACTIVE |        |
| 19eee81c-f972-44e1-a952-1dceee148c47 | cirros-0.3.1-x86_64-uec-ramdisk |
 ACTIVE |        |
+--------------------------------------+---------------------------------
+--------+--------+
```

2. To create a bootable volume from an image, include the image ID in the command:

```
# cinder create --image-id e0b7734d-2331-42a3-b19e-067adc0da17d --display-
name my-boot-vol 8
+---------------------+--------------------------------------+
|       Property      |                Value                 |
+---------------------+--------------------------------------+
|     attachments     |                  []                  |
|  availability_zone  |                 nova                 |
|       bootable      |                false                 |
|      created_at     |      2013-10-16T01:27:48.645390      |
| display_description |                 None                 |
|     display_name    |             my-boot-vol              |
|          id         | 3195a5a7-fd0d-4ac3-b919-7ba6cbe11d46 |
|       image_id      | e0b7734d-2331-42a3-b19e-067adc0da17d |
|       metadata      |                  {}                  |
|         size        |                  8                   |
|     snapshot_id     |                 None                 |
|     source_volid    |                 None                 |
|        status       |              creating                |
|     volume_type     |                 None                 |
+---------------------+--------------------------------------+
```

Before the volume builds, its `bootable` state is `false`.

3. To list volumes, run the following command:

```
$ cinder list
+--------------------------------------+-----------+-----------------
+------+-------------+----------+-------------+
|                  ID                  |   Status  |   Display Name  |
 Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+-----------------
+------+-------------+----------+-------------+
| 3195a5a7-fd0d-4ac3-b919-7ba6cbe11d46 | available |   my-boot-vol   |  8
  |     None    |   true   |             |
```

```
+--------------------------------------+-----------+----------------
+------+-------------+----------+-------------+
```

Note that the `bootable` state is now `true`. Copy the value in the `ID` field for your volume.

4. To launch an instance, include the *`--block_device_mapping`* parameter:

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

The command arguments are:

Parameter	Description
`--flavor FLAVOR`	The flavor ID.

`--block_device_mapping DEVNAME=ID:type:size:delete-on-terminate`	• *`DEVNAME`*. A device name where the volume is attached in the system at `/dev/dev_name`. This value is typically `vda`.
	• *`ID`*. The ID of the volume to boot from, as shown in the output of **nova volume-list**.
	• *`type`*. Either `snap` or any other value, including a blank string. `snap` means that the volume was created from a snapshot.
	• *`size`*. The size of the volume, in GBs. It is safe to leave this blank and have the Compute service infer the size.
	• *`delete-on-terminate`*. Boolean. Indicates whether the volume is deleted when the instance is deleted. You can specify:
	  • `True` or `1`
	  • `False` or `0`

`NAME`	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.

You can also attach a swap disk on boot with the *`--swap`* flag, or you can attach an ephemeral disk on boot with the *`--ephemeral`* flag.

For example, you might enter the following command to boot from a volume. The volume is not deleted when the instance is terminated:

```
$ nova boot --flavor 2 --image e0b7734d-2331-42a3-b19e-067adc0da17d \
    --block_device_mapping vda=3195a5a7-fd0d-4ac3-b919-7ba6cbe11d46:::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                                   | 8ed8b0f9-70de-4662-
a16c-0b51ce7b17b4 |
| security_groups                      | [{u'name': u'default'}]
   |
| user_id                              | 352b37f5c89144d4ad0534139266d51f
   |
| 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                              | 2013-10-16T01:43:26Z
   |
| hostId                               |
   |
| OS-EXT-SRV-ATTR:host                 | None
   |
| key_name                             | None
   |
| OS-EXT-SRV-ATTR:hypervisor_hostname | None
   |
| name                                 | myInstanceFromVolume
   |
| adminPass                            | BULD33uzYwhq
   |
| tenant_id                            | f7ac731cc11f40efbc03a9f9e1d1d21f
   |
| created                              | 2013-10-16T01:43:25Z
   |
| metadata                             | {}
   |
+--------------------------------------
+--------------------------------------+
```

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

```
$ nova volume-list
+--------------------------------------+-----------+-----------------
+------+-------------+--------------------------------------+
```

```
| ID                                   | Status    | Display Name    |
 Size | Volume Type | Attached to                          |
+--------------------------------------+-----------+-----------------
+------+-------------+--------------------------------------+
| 3195a5a7-fd0d-4ac3-b919-7ba6cbe11d46 | in-use    | my-boot-vol     | 8
  | None        | 8ed8b0f9-70de-4662-a16c-0b51ce7b17b4 |
+--------------------------------------+-----------+-----------------
+------+-------------+--------------------------------------+
```

Manage instances and hosts

Instances are virtual machines that run inside the cloud.

Manage IP addresses

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

You can assign floating IP addresses to a project or an instance.

1. Allocate a floating IP address to the current project. If more than one IP address pool is available, you can specify the pool from which to allocate the IP address. This example specifies the `public` pool:

```
$ nova 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, specify the same arguments that you used to assign the IP address:

```
$ 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.

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
   | True      |
| 2  | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0
   | True      |
| 3  | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0
   | True      |
| 4  | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0
   | True      |
| 42 | m1.nano   | 64        | 0    | 0         |      | 1     | 1.0
   | True      |
| 5  | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0
   | True      |
| 84 | m1.micro  | 128       | 0    | 0         |      | 1     | 1.0
   | True      |
+----+-----------+-----------+------+-----------+------+-------
+-------------+-----------+
```

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 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

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 soft or hard reboot a running instance. A soft reboot attempts a graceful shut down and restart of the instance. A hard reboot power cycles the instance.

- 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
```

Delete an instance

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

1. List all instances:

```
$ nova list
+--------------------------------------+----------------------+--------
+------------+-------------+------------------+
| 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 |
| d7efd3e4-d375-46d1-9d57-372b6e4bdb7f | newServer            | ERROR  |
 None       | NOSTATE     |                  |
+--------------------------------------+----------------------+--------
+------------+-------------+------------------+
```

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                                   | 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 |
+--------------------------------------+----------------------+--------
+------------+-------------+------------------+
```

   The deleted instance does not appear in the list.

Get a console to access an instance

- To get a VNC console to access 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

The bare metal driver for OpenStack Compute manages provisioning of physical hardware using common cloud APIs and tools such as Orchestration (Heat). The use case for this driver is for single tenant clouds such as a high-performance computing cluster or deploying OpenStack itself. Development efforts are focused on moving the driver out of the Compute code base in the Icehouse release. 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.

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. List bare metal nodes and interfaces:

```
$ 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                                 |
+------------------+--------------------------------------+
```

Note

Set the *--availability_zone* parameter to specify which zone or node to use to start the server. Separate the zone from the host name with a comma. For example:

```
$ nova boot --availability_zone=zone:host,node
```

host is optional for the *--availability_zone* parameter. *zone:,node* also works.

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.

The `used_now` row shows the sum of the resources allocated to the instances that run on the host plus the resources allocated to the virtual machine of the host itself.

The `used_max` row shows the sum of the resources allocated to the instances that run on the host.

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 |
 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 |
```

```
+--------------------------------------+----------------------+--------
+------------+-------------+------------------+
```

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       |
+----------------------------------+-----------+--------------+-----------
+---------------+
```

Configure instances at boot

When you boot instances in an OpenStack cloud, you can use user data or cloud-init to automatically configure instances at boot time. You might want to install some packages, start services, or manage your instance by using a Puppet or Chef server.

User data

User data is the mechanism by which a user can pass information contained in a local file to an instance at launch time. The typical use case is to pass something like a shell script or a configuration file as user data.

User data is sent using the `--user-data` *`/path/to/filename`* option when calling **nova boot**. This example creates a text file and sends its contents as user data to the instance:

```
$ echo "This is some text" > myfile.txt
$ nova boot --user-data ./myfile.txt --image myimage myinstance
```

The instance can get user data by querying the metadata service through either the OpenStack metadata API or the EC2 compatibility API:

```
$ curl http://169.254.169.254/2009-04-04/user-data
```

```
This is some text
```

```
$ curl http://169.254.169.254/openstack/2012-08-10/user_data
```

```
This is some text
```

Note

The Compute service treats user data as a blob. While the previous example uses a text file, user data can be in any format.

For more information, see the section called “Store metadata on a configuration drive” [60].

cloud-init

To do something useful with the user data, you must configure the virtual machine image to run a service on boot that gets user data from the metadata service and takes some action based on the contents of the data. The cloud-init package does exactly this. This package is compatible with the Compute metadata service and the Compute configuration drive.

Note

The cloud-init package supports multiple cloud providers. You can use the same virtual machine image in different clouds without modification. The cloud-init package is an open source project and the source code is available on Launchpad. It is maintained by Canonical, the company that runs the Ubuntu project. All Ubuntu cloud images come pre-installed with cloud-init. However,

cloud-init is not designed to be Ubuntu-specific and has been successfully ported to several other platforms including Fedora.

Even if you do not use user data to configure instance behavior at boot time, install cloud-init on images that you create because this package provides useful functionality. For example, the cloud-init package enables you to copy the public key to an account (the `ubuntu` account by default on Ubuntu instances, the `ec2-user` by default in Fedora instances).

If you do not have cloud-init installed, you must manually configure your image to get the public key from the metadata service on boot and copy it to the appropriate account.

cloud-init supported formats and documentation

Look at the cloud-init doc/userdata.txt file the examples directory and the Ubuntu community documentation for details about how to use cloud-init. Some basic examples are provided here.

cloud-init supports several different input formats for user data. Two commonly used formats are:

- Shell scripts (starts with `#!`)
- Cloud config files (starts with `#cloud-config`)

Run a shell script on boot

Assuming you have cloud-init installed, the simplest way to configure an instance on boot is to pass a shell script as user data. The shell file must begin with `#!` in order for cloud-init to recognize it as a shell script. This example shows a script that creates a `clouduser` account:

```
#!/bin/bash
adduser --disabled-password --gecos "" clouduser
```

Sending a shell script as user data has a similar effect to writing an `/etc/rc.local` script: it executes very late in the boot sequence as root.

Cloud-config format

cloud-init supports a YAML-based config format that allows the user to configure a large number of options on a system. User data that begins with `#cloud-config` will be interpreted by cloud-init as cloud-config format.

Example: Set the host name

This cloud-init user data example sets the hostname and the FQDN, as well as updating `/etc/hosts` on the instance:

```
#cloud-config
hostname: mynode
fqdn: mynode.example.com
manage_etc_hosts: true
```

Example: Configure instances with Puppet

This cloud-init user data example, based on doc/examples/cloud-config-puppet.txt, would configure the instance to contact a Puppet server at puppetmaster.example.org and verify its identity using a certificate.

```
#cloud-config
puppet:
 conf:
   agent:
     server: "puppetmaster.example.org"
   ca_cert: |
     -----BEGIN CERTIFICATE-----
     MIICCTCCAXKgAwIBAgIBATANBgkqhkiG9w0BAQUFADANMQswCQYDVQQDDAJjYTAe
     Fw0xMDAyMTUxNzI5MjFaFw0xNTAyMTQxNzI5MjFaMA0xCzAJBgNVBAMMAmNhMIGf
     MA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCu7Q40sm47/E1Pf+r8AYb/V/FWGPgc
     b014OmNoX7dgCxTDvps/h8Vw555PdAFsW5+QhsGr31IJNI3kSYprFQcYf7A8tNWu
     1MASW2CfaEiOEi9F1R3R4Qlz4ix+iNoHiUDTjazw/tZwEdxaQXQVLwgTGRwVa+aA
     qbutJKi93MILLwIDAQABo3kwdzA4BglghkgBhvhCAQ0EKxYpUHVwcGV0IFJ1Ynkv
     T3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwDwYDVR0TAQH/BAUwAwEB/zAd
     BgNVHQ4EFgQUu4+jHB+GYE5Vxo+ol1OAhevspjAwCwYDVR0PBAQDAgEGMA0GCSqG
     SIb3DQEBBQUAA4GBAH/rxlUIjwNb3n7TXJcDJ6MMHUlwjr03BDJXKb34Ulndkpaf
     +GAlzPXWa7bO908M9I8RnPfvtKnteLbvgTK+h+zX1XCty+S2EQWk29i2AdoqOTxb
     hppiGMp0tT5Havu4aceCXiy2crVcudj3NFciy8X66SoECemW9UYDCb9T5D0d
     -----END CERTIFICATE-----
```

Example: Configure instances with Chef

This cloud-init user data example, based on doc/examples/cloud-config/chef.txt and intended for use in an Ubuntu image, adds the Chef apt repository, installs Chef, connects to a Chef server at `https://chefserver.example.com:4000`, and installs Apache:

```
#cloud-config
apt_sources:
 - source: "deb http://apt.opscode.com/ $RELEASE-0.10 main"
   key: |
     -----BEGIN PGP PUBLIC KEY BLOCK-----
     Version: GnuPG v1.4.9 (GNU/Linux)

     mQGiBEppC7QRBADfsOkZU6KZK+YmKw4wev5mjKJEkVGlus+NxW8wItX5sGa6kdUu
     twAyj7Yr92rF+ICFEP3gGU6+lGo0Nve7KxkN/1W7/m3G4zuk+ccIKmjp8KS3qn99
     dxy64vcji9jIllVa+XXOGIp0G8GEaj7mbkixL/bMeGfdMlv8Gf2XPpp9vwCgn/GC
     JKacfnw7MpLKUHOYSlb//JsEAJqao3ViNfav83jJKEkD8cf59Y8xKia5OpZqTK5W
     ShVnNWS3U5IVQk10ZDH97Qn/YrK387H4CyhLE9mxPXs/ul18ioiaars/q2MEKU2I
     XKfV21eMLO9LYd6Ny/Kqj8o5WQK2J6+NAhSwvthZcIEphcFignIuobP+B5wNFQpe
     DbKfA/0WvN2OwFeWRcmmd3Hz7nHTpcnSF+4QX6yHRF/5BgxkG6IqBIACQbzPn6Hm
     sMtm/SVf11izmDqSsQptCrOZILfLX/mE+YOl+CwWSHhl+YsFts1WOuh1EhQD26aO
     Z84HuHV5HFRWjDLw9LriltBVQcXbpfSrRP5bdr7Wh8vhqJTPjrQnT3BzY29kZSBQ
     YWNrYWdlcyA8cGFja2FnZXNAb3BzY29kZS5jb20+iGAEExECACAFAkppC7QCGwMG
     CwkIBwMCBBUCCAMEFgIDAQIeAQIXgAAKCRApQKupg++Caj8sAKCOXmdG36gWji/K
     +o+XtBfvdMnFYQCfTCEWxRy2BnzLoBBFCjDSK6sJqCu5Ag0ESmkLtBAIAIO2SwlR
     lU5i6gTOp42RHWW7/pmW78CwUqJnYqnXROrt3h9F9xrsGkH0Fh1FRtsnncgzIhvh
     DLQnRHnkXm0ws0jV0PF74ttoUT6BLAUsFi2SPP1zYNJ9H9fhhK/pjijtAcQwdgxu
     wwNJ5xCEscBZCjhSRXm0d30bK1o49Cow8ZIbHtnXVP41c9QWOzX/LaGZsKQZnaMx
     EzDk8dyyctR2f03vRSVyTFGgdpUcpbr9eTFVgikCa6ODEBv+0BnCH6yGTXwBid9g
     w0o1e/2DviKUWCC+AlAUOubLmOIGFBuI4UR+rux9affbHcLIOTiKQXv79lW3P7W8
     AAfniSQKfPWXrrcAAwUH/2XBqD4Uxhbs25HDUUiM/m6Gnlj6EsStg8n0nMggLhuN
     QmPfoNByMPUqvA7sULyfr6xCYzbzRNxABHSpf85FzGQ29RF4xsA4vOOU8RDIYQ9X
```

```
    Q8NqqR6pydprRFqWe47hsAN7BoYuhWqTtOLSBmnAnzTR5pURoqcquWYiiEavZixJ
    3ZRAq/HMGioJEtMFrvsZjGXuzef7f0ytfR1zYeLVWnL9Bd32CueBlI7dhYwkFe+V
    Ep5jWOCj02C1wHcwt+uIRDJV6TdtbIiBYAdOMPk15+VBdweBXwMuYXr76+A7VeDL
    zIhi7tKFo6WiwjKZq0dzctsJJjtIfr4K4vbiD9Ojg1iISQQYEQIACQUCSmkLtAIb
    DAAKCRApQKupg++CauISAJ9CxYPOKhOxalBnVTLeNUkAHGg2gACeIsbobtaD4ZHG
    0GLl8EkfA8uhluM=
    =zKAm
    -----END PGP PUBLIC KEY BLOCK-----

chef:
 install_type: "packages"
 server_url: "https://chefserver.example.com:4000"
 node_name: "your-node-name"
 environment: "production"
 validation_name: "yourorg-validator"
 validation_key: |
    -----BEGIN RSA PRIVATE KEY-----
    YOUR-ORGS-VALIDATION-KEY-HERE
    -----END RSA PRIVATE KEY-----
 run_list:
  - "recipe[apache2]"
  - "role[db]"
 initial_attributes:
    apache:
      prefork:
        maxclients: 100
      keepalive: "off"
```

Provide user data to instances

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 a Ubuntu open source package that handles early initialization of a cloud instance and that makes use of `user data`.

You can place user data in a local file and pass it through the `--user-data <user-data-file>` parameter at instance creation:

```
$ nova boot --image ubuntu-cloudimage --flavor 1 --user-data mydata.file
```

Store metadata on a configuration drive

You can configure OpenStack to write metadata to a special configuration drive that attaches to the instance when it boots. The instance can mount this drive and read files from it to get information that is normally available through the metadata service.

One use case for the configuration drive is to pass a networking configuration when you do not use DHCP to assign IP addresses to instances. For example, you might pass the IP configuration for the instance through the configuration drive, which the instance can mount and access before the you configure the network settings for the instance.

Any modern guest operating system that is capable of mounting an ISO9660 or VFAT file system can use the configuration drive.

Requirements and guidelines

Compute host requirements

- The following hypervisors support the configuration drive: libvirt, xenserver, hyper-v, and vmware.
- To use configuration drive with libvirt, xenserver, or vmware, you must first install the genisoimage package on each Compute host. Otherwise, instances do not boot properly.

  Use the `mkisofs_cmd` flag to set the path where you install the genisoimage program. If genisoimage is in same path as the `nova-compute` service, you do not need to set this flag.

  Note

  By default, Ubuntu packages do not install this package. See bug #1165174.
- To use configuration drive with hyper-v, you must set the `mkisofs_cmd` value to the full path to an `mkisofs.exe` installation. Additionally, you must set the `qemu_img_cmd` value in the `hyperv` configuration section to the full path to an `qemu-img` command installation.

Image requirements

- An image built with a recent version of the cloud-init package can automatically access metadata passed through the configuration drive. The cloud-init package version 0.7.1 works with Ubuntu and Fedora-based images, such as RHEL.
- If an image does not have the cloud-init package installed, you must customize the image to run a script that mounts the configuration drive on boot, reads the data from the drive, and takes appropriate action such as adding the public key to an account. See the section called "Configuration drive contents" [62] for details on how data is organized on the configuration drive.
- If you use Xen with a configuration drive, use the `xenapi_disable_agent` configuration parameter to disable the agent.

Guidelines

- Do not rely on the presence of the EC2 metadata present in the configuration drive, as this content might be removed in a future release. For example, do not rely on files in the `ec2` directory.
- When you create images that access configuration drive data and multiple directories are under the `openstack` directory, always select the highest API version by date that your consumer supports. For example, if your guest image supports the 2012-03-05, 2012-08-05, 2013-04-13 versions, try 2013-04-13 first and fall back to a previous version if 2013-04-13 is not present.

Enable and access the configuration drive

1. To enable the configuration drive, pass the `--config-drive=true` parameter to the **nova boot** command.

   This example enables the configuration drive and passes user data, two files, and two key/value metadata pairs, all of which are accessible from the configuration drive:

   ```
   $ nova boot --config-drive=true --image my-image-name --key-name mykey --
   flavor 1 --user-data ./my-user-data.txt myinstance --file /etc/network/
   interfaces=/home/myuser/instance-interfaces --file known_hosts=/home/
   myuser/.ssh/known_hosts --meta role=webservers --meta essential=false
   ```

   You can also configure the Compute service to always create a configuration drive.

   Set this option in the `/etc/nova/nova.conf` file:

   ```
   force_config_drive=true
   ```

   Note

   If a user passes the *`--config-drive=true`* flag to the **nova boot** command, an administrator cannot disable the configuration drive.

2. The configuration drive has the `config-2` volume label. If your guest operating system supports accessing disk by label, you can mount the configuration drive as the `/dev/disk/by-label/config-2` device.

   For example:

   ```
   # mkdir -p /mnt/config
   # mount /dev/disk/by-label/config-2 /mnt/config
   ```

   Note

   Make sure that you use at least version 0.3.1 of CirrOS for configuration drive support.

   If your guest operating system does not use `udev`, the `/dev/disk/by-label` directory is not present.

You can use the **blkid** command to identify the block device that corresponds to the configuration drive. For example, when you boot the CirrOS image with the `m1.tiny` flavor, the device is `/dev/vdb`:

```
# blkid -t LABEL="config-2" -odevice
```

```
/dev/vdb
```

Once identified, you can mount the device:

```
# mkdir -p /mnt/config
# mount /dev/vdb /mnt/config
```

Configuration drive contents

In this example, the contents of the configuration drive are:

```
ec2/2009-04-04/meta-data.json
ec2/2009-04-04/user-data
ec2/latest/meta-data.json
ec2/latest/user-data
openstack/2012-08-10/meta_data.json
openstack/2012-08-10/user_data
openstack/content
openstack/content/0000
openstack/content/0001
openstack/latest/meta_data.json
openstack/latest/user_data
```

The files that appear on the configuration drive depend on the arguments that you pass to the **nova boot** command.

OpenStack metadata format

The following example shows the contents of the `openstack/2012-08-10/meta_data.json` and `openstack/latest/meta_data.json` files. These files are identical. The file contents are formatted for readability:

```
{
   "availability_zone":"nova",
   "files":[
      {
         "content_path":"/content/0000",
         "path":"/etc/network/interfaces"
      },
      {
         "content_path":"/content/0001",
         "path":"known_hosts"
      }
   ],
   "hostname":"test.novalocal",
   "launch_index":0,
   "name":"test",
   "meta":{
      "role":"webservers",
      "essential":"false"
   },
```

```
   "public_keys":{
      "mykey":"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDBqUfVvCSez0/
Wfpd8dLLgZXV9GtXQ7hnMN+Z0OWQUyebVEHey1CXuin0uY1cAJMhUq8j98SiW
+cU0sU4J3x5l2+xi1bodDm1BtFWVeLIOQINpfV1n8fKjHB
+ynPpe1F6tMDvrFGUlJs44t30BrujMXBe8Rq44cCk6wqyjATA3rQ== Generated by Nova\n"
   },
   "uuid":"83679162-1378-4288-a2d4-70e13ec132aa"
}
```

Note the effect of the `--file /etc/network/interfaces=/home/myuser/instance-interfaces` argument that was passed to the **nova boot** command. The contents of this file are contained in the `openstack/content/0000` file on the configuration drive, and the path is specified as `/etc/network/interfaces` in the `meta_data.json` file.

EC2 metadata format

The following example shows the contents of the `ec2/2009-04-04/meta-data.json`, `latest/meta-data.json` files. These files are identical. The file contents are formatted to improve readability:

```
{
   "ami-id":"ami-00000001",
   "ami-launch-index":0,
   "ami-manifest-path":"FIXME",
   "block-device-mapping":{
      "ami":"sda1",
      "ephemeral0":"sda2",
      "root":"/dev/sda1",
      "swap":"sda3"
   },
   "hostname":"test.novalocal",
   "instance-action":"none",
   "instance-id":"i-00000001",
   "instance-type":"m1.tiny",
   "kernel-id":"aki-00000002",
   "local-hostname":"test.novalocal",
   "local-ipv4":null,
   "placement":{
      "availability-zone":"nova"
   },
   "public-hostname":"test.novalocal",
   "public-ipv4":"",
   "public-keys":{
      "0":{
         "openssh-key":"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDBqUfVvCSez0/
Wfpd8dLLgZXV9GtXQ7hnMN+Z0OWQUyebVEHey1CXuin0uY1cAJMhUq8j98SiW
+cU0sU4J3x5l2+xi1bodDm1BtFWVeLIOQINpfV1n8fKjHB
+ynPpe1F6tMDvrFGUlJs44t30BrujMXBe8Rq44cCk6wqyjATA3rQ== Generated by Nova\n"
      }
   },
   "ramdisk-id":"ari-00000003",
   "reservation-id":"r-7lfps8wj",
   "security-groups":[
      "default"
   ]
}
```

User data

The `openstack/2012-08-10/user_data`, `openstack/latest/user_data`, `ec2/2009-04-04/user-data`, and `ec2/latest/user-data` file are present only if the `--user-data` flag and the contents of the user data file are passed to the **nova boot** command.

Configuration drive format

The default format of the configuration drive as an ISO 9660 file system. To explicitly specify the ISO 9660 format, add the following line to the `/etc/nova/nova.conf` file:

```
config_drive_format=iso9660
```

By default, you cannot attach the configuration drive image as a CD drive instead of as a disk drive. To attach a CD drive, add this line to the `/etc/nova/nova.conf` file:

```
config_drive_cdrom=true
```

For legacy reasons, you can configure the configuration drive to use VFAT format instead of ISO 9660. It is unlikely that you would require VFAT format because ISO 9660 is widely supported across operating systems. However, to use the VFAT format, add the following line to the `/etc/nova/nova.conf` file:

```
config_drive_format=vfat
```

If you choose VFAT, the configuration drive is 64 MBs.

Configuration drive reference

The following table shows the configuration options for the configuration drive:

Table 2.4. Description of configuration options for configdrive

Configuration option=Default value	Description
config_drive_cdrom=False	(BoolOpt) Attaches the Config Drive image as a cdrom drive instead of a disk drive
config_drive_format=iso9660	(StrOpt) Config drive format. One of iso9660 (default) or vfat
config_drive_inject_password=False	(BoolOpt) Sets the admin password in the config drive image
config_drive_skip_versions=1.0 2007-01-19 2007-03-01 2007-08-29 2007-10-10 2007-12-15 2008-02-01 2008-09-01	(StrOpt) List of metadata versions to skip placing into the config drive
config_drive_tempdir=None	(StrOpt) Where to put temporary files associated with config drive creation
force_config_drive=None	(StrOpt) Set to force injection to take place on a config drive (if set, valid options are: always)
mkisofs_cmd=genisoimage	(StrOpt) Name and optionally path of the tool used for ISO image creation

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.

Manage objects and containers

The OpenStack Object Storage Service provides the **swift** client, which is a command-line interface (CLI). Use this client to list objects and containers, upload objects to containers, and download or delete objects from containers. You can also gather statistics and update metadata for accounts, containers, and objects.

This client is based on the native swift client library, `client.py`, which seamlessly re-authenticates if the current token expires during processing, retries operations multiple times, and provides a processing concurrency of 10.

Users have roles on accounts. For example, a user with the admin role has full access to all containers and objects in an account. You can set access control lists (ACLs) at the container level and support lists for read and write access, which you set with the `X-Container-Read` and `X-Container-Write` header, respectively.

To give a user read access, use the **swift post** command with the *`-r`* parameter. To give a user write access, use the *`-w`* parameter.

The following example enables the `testuser` user to read objects in the container:

```
$ swift post -r 'testuser'
```

You can also use this command with a list of users.

If you use StaticWeb middleware to enable Object Storage to serve public web content, use `.r:`, followed by a list of allowed referrers.

The following command gives object access to all referring domains:

```
$ swift post -r '.r:*'
```

Create and manage stacks

Create a stack from an example template file

1. To create a stack, or template, from an example template file, run the following command:

```
$ heat stack-create mystack --template-file=/PATH_TO_HEAT_TEMPLATES/
WordPress_Single_Instance.template
     --parameters="InstanceType=m1.
large;DBUsername=USERNAME;DBPassword=PASSWORD;KeyName=HEAT_KEY;LinuxDistribution=
F17"
```

   The `--parameters` values that you specify depend on the parameters that are defined in the template. If a website hosts the template file, you can specify the URL with the `--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.

Get information about stacks

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

- 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 show the details of a stack, run the following command:

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

- A stack consists of a collection of resources.

  To list the resources and their status, 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 |
+----------------------+--------------------+-----------------
+----------------------+
```

- 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
```

- A series of events is generated during the life-cycle of a stack.

  To display life-cycle events, run::

```
$ 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 |
+----------------------+----+-----------------------+-----------------
+----------------------+
```

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

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

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.

Measure cloud resources

The OpenStack Telemetry service measures cloud resources within OpenStack.

It collects information about how much, who, what, and when with regards to billing. For Havana, metering is available through only the **ceilometer** command-line interface (CLI).

The following example uses the **heat** client to create an auto-scaling stack and the **ceilometer** client to measure resources.

1. Create an auto-scaling stack:

```
$ heat stack-create -f cfn/F17/AutoScalingCeilometer.yaml -P "KeyName=
heat_key"
```

2. List the heat resources that were created:

```
$ heat resource-list
+-------------------------+-----------------------------------------
+-----------------+---------------------+
| resource_name           | resource_type                           |
resource_status | updated_time         |
+-------------------------+-----------------------------------------
+-----------------+---------------------+
| CfnUser                 | AWS::IAM::User                          |
CREATE_COMPLETE | 2013-10-02T05:53:41Z |
| WebServerKeys           | AWS::IAM::AccessKey                     |
CREATE_COMPLETE | 2013-10-02T05:53:42Z |
| LaunchConfig            | AWS::AutoScaling::LaunchConfiguration   |
CREATE_COMPLETE | 2013-10-02T05:53:43Z |
| ElasticLoadBalancer     | AWS::ElasticLoadBalancing::LoadBalancer |
UPDATE_COMPLETE | 2013-10-02T05:55:58Z |
| WebServerGroup          | AWS::AutoScaling::AutoScalingGroup      |
CREATE_COMPLETE | 2013-10-02T05:55:58Z |
| WebServerScaleDownPolicy | AWS::AutoScaling::ScalingPolicy        |
CREATE_COMPLETE | 2013-10-02T05:56:00Z |
| WebServerScaleUpPolicy   | AWS::AutoScaling::ScalingPolicy        |
CREATE_COMPLETE | 2013-10-02T05:56:00Z |
| CPUAlarmHigh            | OS::Ceilometer::Alarm                   |
CREATE_COMPLETE | 2013-10-02T05:56:02Z |
| CPUAlarmLow             | OS::Ceilometer::Alarm                   |
CREATE_COMPLETE | 2013-10-02T05:56:02Z |
+-------------------------+-----------------------------------------
+-----------------+---------------------+
```

3. List the alarms that are set:

```
$ ceilometer alarm-list
+----------------------------------+-----------------------------
+------------------+---------+------------
+--------------------------------+
| Alarm ID                         | Name                         |
 State             | Enabled | Continuous | Alarm condition
 |
+----------------------------------+-----------------------------
+------------------+---------+------------
+--------------------------------+
```

```
| 4f896b40-0859-460b-9c6a-b0d329814496 | as-CPUAlarmLow-i6qqgkf2fubs  |
 insufficient data | True    | False      | cpu_util < 15.0 during 1x 60s
 |
| 75d8ecf7-afc5-4bdc-95ff-19ed9ba22920 | as-CPUAlarmHigh-sf4muyfruy5m |
 insufficient data | True    | False      | cpu_util > 50.0 during 1x 60s
 |
+--------------------------------------+-----------------------------
+------------------+---------+-----------+-----------------------------+
```

4. List the meters that are set:

```
$ ceilometer meter-list
+-------------------------+------------+----------
+--------------------------------------------------------------------------
+-----------------------------------+
| Name                      | Type       | Unit     | Resource ID
               | User ID                          | Project ID
           |
+-------------------------+------------+----------
+--------------------------------------------------------------------------
+-----------------------------------+
| cpu                       | cumulative | ns       | 3965b41b-81b0-4386-
bea5-6ec37c8841c1 | d1a2996d3b1f4e0e8645ba9650308011 |
 bf03bf32e3884d489004ac995ff7a61c |
| cpu                       | cumulative | ns       | 62520a83-73c7-4084-
be54-275fe770ef2c | d1a2996d3b1f4e0e8645ba9650308011 |
 bf03bf32e3884d489004ac995ff7a61c |
| cpu_util                  | gauge      | %        | 3965b41b-81b0-4386-
bea5-6ec37c8841c1 | d1a2996d3b1f4e0e8645ba9650308011 |
 bf03bf32e3884d489004ac995ff7a61c |
+-------------------------+------------+----------
+--------------------------------------------------------------------------
+-----------------------------------+
```

5. List samples:

```
$ ceilometer sample-list -m cpu_util
+--------------------------------------+----------+-------+--------------
+------+---------------------+
| Resource ID                          | Name     | Type  | Volume
 | Unit | Timestamp           |
+--------------------------------------+----------+-------+--------------
+------+---------------------+
| 3965b41b-81b0-4386-bea5-6ec37c8841c1 | cpu_util | gauge | 3.98333333333
 | %    | 2013-10-02T10:50:12 |
+--------------------------------------+----------+-------+--------------
+------+---------------------+
```

6. View statistics:

```
$ ceilometer statistics -m cpu_util
+--------+---------------------+---------------------+-------
+---------------+---------------+---------------+---------------
+----------+---------------------+---------------------+
| Period | Period Start        | Period End          | Count | Min
    | Max           | Sum           | Avg           | Duration | Duration
 Start      | Duration End        |
+--------+---------------------+---------------------+-------
+---------------+---------------+---------------+---------------
+----------+---------------------+---------------------+
```

```
| 0      | 2013-10-02T10:50:12 | 2013-10-02T10:50:12 | 1     | 3.
98333333333 | 3.98333333333 | 3.98333333333 | 3.98333333333 | 0.0      |
 2013-10-02T10:50:12 | 2013-10-02T10:50:12 |
+--------+---------------------+---------------------+-------
+---------------+---------------+---------------+---------------
+----------+---------------------+---------------------+
```

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 `my-new-volume` volume based on an image.

Create a volume

1. List images, and note the ID of the image to use for your volume:

```
$ 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 |
                       |
| 7e5142af-1253-4634-bcc6-89482c5f2e8a | myCirrosImage                   | ACTIVE | 84c6e57d-
a6b1-44b6-81eb-fcb36afd31b5 |
| 89bcd424-9d15-4723-95ec-61540e8a1979 | mysnapshot                      | ACTIVE | f51ebd07-
c33d-4951-8722-1df6aa8afaa4 |
+--------------------------------------+---------------------------------+--------
+--------------------------------------+
```

2. List the availability zones, and note the ID of the availability zone in which to create your volume:

```
$ 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 |
+----------------------+----------------------------------------+
```

3. Create a volume with 8 GBs of space. Specify the availability zone and image:

```
$ 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
```

```
+--------------------------------------+-----------+-----------------+------+-------------
+----------+-------------+
|                  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   |             |
+--------------------------------------+-----------+-----------------+------+-------------
+----------+-------------+
```

If your volume was created successfully, its status is `available`. If its status is `error`, you might have exceeded your quota.

Attach a volume to an instance

1. 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.

2. Show information for your volume:

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

```
+-----------------------------
+-------------------------------------------------------------------------------------------------------------------------------
+
|           Property           |
                                      Value
                                                 |
+-----------------------------
+-------------------------------------------------------------------------------------------------------------------------------
+
|         attachments          |             [{u'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
                                                    |
+------------------------------
+-----------------------------------------------------------------------------------------------------------------------
+
```

The output shows that the volume is attached to the server with ID `84c6e57d-a6b1-44b6-81eb-fcb36afd31b5`, is in the nova availability zone, and is bootable.

Resize a volume

1. To resize 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:

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

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

2. 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.

3. Resize the volume by passing the volume ID and the new size (a value greater than the old one) as parameters:

```
$ cinder extend 573e024d-5235-49ce-8332-be1576d323f8 10
```

The **extend** command does not return any output.

Delete a volume

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

   To detach the volume from your server and check for the list of existing volumes, see steps 1 and 2 mentioned in the section called "Resize a volume" [78].

2. Delete the volume:

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

The delete command does not return any output.

3. 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   |             |
+--------------------------------------+-----------+-----------------+------+-------------
+----------+-------------+
```

Transfer a volume

You can transfer a volume from one owner to another by using the **cinder transfer*** commands. The volume donor, or original owner, creates a transfer request and sends the created transfer ID and authorization key to the volume recipient. The volume recipient, or new owner, accepts the transfer by using the ID and key.

Use cases include:

- Create a custom bootable volume or a volume with a large data set and transfer it to the end customer.
- For bulk import of data to the cloud, the data ingress system creates a new Block Storage volume, copies data from the physical device, and transfers device ownership to the end user.

Create a volume transfer request

1. While logged in as the volume donor, list available volumes:

```
$ cinder list
```

```
+--------------------------------------+-----------+--------------+------
+-------------+----------+-------------+
|                  ID                  |   Status  | Display Name | Size |
 Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+--------------+------
+-------------+----------+-------------+
| 72bfce9f-cacf-477a-a092-bf57a7712165 |   error   |     None     |  1   |
     None    |  false   |             |
| a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f | available |     None     |  1   |
     None    |  false   |             |
+--------------------------------------+-----------+--------------+------
+-------------+----------+-------------+
```

2. As the volume donor, request a volume transfer authorization code for a specific volume:

```
$ cinder transfer-create volumeID
```

The volume must be in an 'available' state or the request will be denied. If the transfer request is valid in the database (that is, it has not expired or been deleted), the volume is placed in an `awaiting transfer` state. For example:

```
$ cinder transfer-create a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f
```

```
+------------+--------------------------------------+
|  Property  |                Value                 |
+------------+--------------------------------------+
|  auth_key  |           b2c8e585cbc68a80           |
| created_at |      2013-10-14T15:20:10.121458      |
|     id     | 6e4e9aa4-bed5-4f94-8f76-df43232f44dc |
|    name    |                 None                 |
| volume_id  | a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f |
+------------+--------------------------------------+
```

Note

Optionally, you can specify a name for the transfer by using the `--display-name` *displayName* parameter.

3. Send the volume transfer ID and authorization key to the new owner (for example, by email).
4. View pending transfers:

```
$ cinder transfer-list
```

```
+--------------------------------------
+--------------------------------------+------+
|               ID                     |             VolumeID
    | Name |
+--------------------------------------
+--------------------------------------+------+
| 6e4e9aa4-bed5-4f94-8f76-df43232f44dc | a1cdace0-08e4-4dc7-
b9dc-457e9bcfe25f | None |
+--------------------------------------
+--------------------------------------+------+
```

5. After the volume recipient, or new owner, accepts the transfer, you can see that the transfer is no longer available:

```
$ cinder transfer-list
```

```
+----+-----------+------+
| ID | Volume ID | Name |
+----+-----------+------+
+----+-----------+------+
```

Accept a volume transfer request

1. As the volume recipient, you must first obtain the transfer ID and authorization key from the original owner.

2. Display the transfer request details using the ID:

```
$ cinder transfer-show transferID
```

For example:

```
$ cinder transfer-show 6e4e9aa4-bed5-4f94-8f76-df43232f44dc
```

```
+------------+--------------------------------------+
|  Property  |                Value                 |
+------------+--------------------------------------+
| created_at |      2013-10-14T15:20:10.000000      |
|     id     | 6e4e9aa4-bed5-4f94-8f76-df43232f44dc |
|    name    |                 None                 |
| volume_id  | a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f |
+------------+--------------------------------------+
```

3. Accept the request:

```
$ cinder transfer-accept transferID authKey
```

For example:

```
$ cinder transfer-accept 6e4e9aa4-bed5-4f94-8f76-df43232f44dc
 b2c8e585cbc68a80
```

```
+-----------+--------------------------------------+
|  Property |                Value                 |
+-----------+--------------------------------------+
|     id    | 6e4e9aa4-bed5-4f94-8f76-df43232f44dc |
|    name   |                 None                 |
| volume_id | a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f |
+-----------+--------------------------------------+
```

Note

If you do not have a sufficient quota for the transfer, the transfer is refused.

Delete a volume transfer

1. List available volumes and their statuses:

```
$ cinder list
```

```
+--------------------------------------+-------------------+--------------
+------+-------------+----------+-------------+
|                  ID                  |       Status      | Display Name
 | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-------------------+--------------
+------+-------------+----------+-------------+
| 72bfce9f-cacf-477a-a092-bf57a7712165 |       error       |     None
 |  1   |     None    |  false   |             |
| a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f | awaiting-transfer |     None
 |  1   |     None    |  false   |             |
+--------------------------------------+-------------------+--------------
+------+-------------+----------+-------------+
```

2. Find the matching transfer ID:

```
$ cinder transfer-list
```

```
+--------------------------------------
+--------------------------------------+------+
|               ID                     |             VolumeID
    | Name |
+--------------------------------------
+--------------------------------------+------+
| a6da6888-7cdf-4291-9c08-8c1f22426b8a | a1cdace0-08e4-4dc7-
b9dc-457e9bcfe25f | None |
+--------------------------------------
+--------------------------------------+------+
```

3. Delete the volume:

```
$ cinder transfer-delete transferID
```

For example:

```
$ cinder transfer-delete a6da6888-7cdf-4291-9c08-8c1f22426b8a
```

4. The transfer list is now empty and the volume is again available for transfer:

```
$ cinder transfer-list
```

```
+----+-----------+------+
| ID | Volume ID | Name |
+----+-----------+------+
+----+-----------+------+
```

```
$ cinder list
```

```
+--------------------------------------+-----------+--------------+------
+-------------+----------+-------------+
|                  ID                  |   Status  | Display Name | Size |
 Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+--------------+------
+-------------+----------+-------------+
| 72bfce9f-cacf-477a-a092-bf57a7712165 |   error   |     None     |  1   |
     None    |  false   |             |
| a1cdace0-08e4-4dc7-b9dc-457e9bcfe25f | available |     None     |  1   |
     None    |  false   |             |
+--------------------------------------+-----------+--------------+------
+-------------+----------+-------------+
```

Appendix A. Command reference

Table of Contents

keystone commands 84
glance commands 86
neutron commands 88
nova commands 92
cinder commands 97
swift commands 99
heat commands 100
ceilometer commands 101

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-tenant-id <auth-tenant-id>]
        [--os-username <auth-username>] [--os-password <auth-password>]
        [--os-region-name <auth-region-name>] [--os-token <token>]
        [--endpoint-type <endpoint-type>] [--os-url <url>]
        [--os-cacert <ca-certificate>] [--insecure]
```

Example A.8. Positional arguments

```
 agent-delete                  Delete a given agent.
 agent-list                    List agents.
 agent-show                    Show information of a given agent.
 agent-update                  Update a given agent.
 cisco-credential-create       Creates a credential.
 cisco-credential-delete       Delete a  given credential.
 cisco-credential-list         List credentials that belong to a given
tenant.
 cisco-credential-show         Show information of a given credential.
 cisco-network-profile-create  Creates a network profile.
 cisco-network-profile-delete  Delete a given network profile.
```

```
 cisco-network-profile-list     List network profiles that belong to a given
tenant.
 cisco-network-profile-show     Show information of a given network profile.
 cisco-network-profile-update   Update network profile's information.
 cisco-policy-profile-list      List policy profiles that belong to a given
tenant.
 cisco-policy-profile-show      Show information of a given policy profile.
 cisco-policy-profile-update    Update policy profile's information.
 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.
 firewall-create                Create a firewall.
 firewall-delete                Delete a given firewall.
 firewall-list                  List firewalls that belong to a given tenant.
 firewall-policy-create         Create a firewall policy.
 firewall-policy-delete         Delete a given firewall policy.
 firewall-policy-insert-rule    Insert a rule into a given firewall policy.
 firewall-policy-list           List firewall policies that belong to a given
tenant.
 firewall-policy-remove-rule    Remove a rule from a given firewall policy.
 firewall-policy-show           Show information of a given firewall policy.
 firewall-policy-update         Update a given firewall policy.
 firewall-rule-create           Create a firewall rule.
 firewall-rule-delete           Delete a given firewall rule.
 firewall-rule-list             List firewall rules that belong to a given
tenant.
 firewall-rule-show             Show information of a given firewall rule.
 firewall-rule-update           Update a given firewall rule.
 firewall-show                  Show information of a given firewall.
 firewall-update                Update a given firewall.
 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
 ipsec-site-connection-create   Create an IPsecSiteConnection.
 ipsec-site-connection-delete   Delete a given IPsecSiteConnection.
 ipsec-site-connection-list     List IPsecSiteConnections that belong to a
given tenant.
 ipsec-site-connection-show     Show information of a given
IPsecSiteConnection.
 ipsec-site-connection-update   Update a given IPsecSiteConnection.
 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-agent-hosting-pool          Get loadbalancer agent hosting a pool.
 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-list-on-agent          List the pools on a loadbalancer agent.
 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              List external networks that belong to a given
tenant.
 net-gateway-connect            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           Remove an external network gateway from a
router.
 router-gateway-set             Set the external network gateway for a
router.
 router-interface-add           Add an internal network interface to a
router.
 router-interface-delete        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                List ports that belong to a given tenant,
with specified router.
 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             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        List security group rules that belong to a
given tenant.
 security-group-rule-show        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.
 service-provider-list           List service providers.
 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.
 vpn-ikepolicy-create            Create an IKEPolicy.
 vpn-ikepolicy-delete            Delete a given IKE Policy.
 vpn-ikepolicy-list              List IKEPolicies that belong to a tenant.
 vpn-ikepolicy-show              Show information of a given IKEPolicy.
 vpn-ikepolicy-update            Update a given IKE Policy.
 vpn-ipsecpolicy-create          Create an ipsecpolicy.
 vpn-ipsecpolicy-delete          Delete a given ipsecpolicy.
 vpn-ipsecpolicy-list            List ipsecpolicies that belongs to a given
tenant connection.
 vpn-ipsecpolicy-show            Show information of a given ipsecpolicy.
 vpn-ipsecpolicy-update          Update a given ipsec policy.
 vpn-service-create              Create a VPNService.
 vpn-service-delete              Delete a given VPNService.
 vpn-service-list                List VPNService configurations that belong to
a given tenant.
 vpn-service-show                Show information of a given VPNService.
 vpn-service-update              Update a given VPNService.
```

Example A.9. 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-tenant-id <auth-tenant-id>
                       Authentication tenant name (Env: OS_TENANT_ID)
 --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.
```

nova commands

The nova client is the command-line interface for the 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-tenant-id <auth-tenant-id>] [--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.
    list-secgroup       List Security Group(s) of a server.
    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'.
    volume-update       Update volume attachment.
    x509-create-cert    Create x509 cert for a user in tenant.
    x509-get-root-cert  Fetches the x509 root cert.
    bash-completion     Prints out a list of all possible commands and flags
                        that the nova command can accept. Can be used by bash
                        for tab-completion of commands.
    help                Display help about this program or one of its
                        subcommands.
    cell-capacities     Get cell capacities for all cells or a given cell.
    cell-show           Show details of a given cell.
    host-servers-migrate
                        Migrate all instances of the specified host to other
                        available hosts.
    net                 Show a network
```

```
    net-create          Create a network
    net-delete          Delete a network
    net-list            List networks
    force-delete        Force delete a server.
    restore             Restore a soft-deleted server.
    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.
    migration-list      Print a list of migrations.
    host-meta           Set or Delete metadata on all instances of a host.
    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.
    instance-action     Show an action.
    instance-action-list
                        List actions on a server.
```

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-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].
 --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 Block Storage 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
    availability-zone-list
                        List all the availability zones.
    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.
    extend              Attempt to extend the size of an existing volume.
    extra-specs-list    Print a list of current 'volume types and extra specs'
                        (Admin Only).
    force-delete        Attempt forced removal of a volume, regardless of its
                        state.
    list                List all the volumes.
```

```
  metadata            Set or Delete metadata on a volume.
  migrate             Migrate the volume to the new host.
  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-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.
  reset-state         Explicitly update the state of a volume.
  service-disable     Disable the service.
  service-enable      Enable the service.
  service-list        List all the services. Filter by host & service
                      binary.
  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-reset-state
                      Explicitly update the state of a snapshot.
  snapshot-show       Show details about a snapshot.
  transfer-accept     Accepts a volume transfer.
  transfer-create     Creates a volume transfer.
  transfer-delete     Undo a transfer.
  transfer-list       List all the transfers.
  transfer-show       Show details about a transfer.
  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 [--version] [--help] [--snet] [--verbose]
             [--debug] [--quiet] [--auth <auth_url>]
             [--auth-version <auth_version>] [--user <username>]
             [--key <api_key>] [--retries <num_retries>]
             [--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-auth-token <auth-token>]
             [--os-storage-url <storage-url>] [--os-region-name <region-name>]
             [--os-service-type <service-type>]
             [--os-endpoint-type <endpoint-type>]
             [--os-cacert <ca-certificate>] [--insecure]
             [--no-ssl-compression]
             <subcommand> ...
```

Example A.17. Commands

```
<subcommand>
    delete               Delete a container or objects within a container
    download             Download objects from containers
    list                 Lists the containers for the account or the objects
                         for a container
    post                 Updates meta information for the account, container,
                         or object
    stat                 Displays information for the account, container,
                         or object
    upload               Uploads files or directories to the given container
```

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>
    action-resume       Resume the stack.
    action-suspend      Suspend the stack.
    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.
```

ceilometer commands

The ceilometer client is the command-line interface for the OpenStack Telemetry API.

For help on a specific ceilometer command, enter:

```
$ ceilometer help COMMAND
```

Example A.22. Usage

```
ceilometer [--version] [-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]
                  [--ceilometer-url CEILOMETER_URL]
                  [--ceilometer-api-version CEILOMETER_API_VERSION]
                  [--os-service-type OS_SERVICE_TYPE]
                  [--os-endpoint-type OS_ENDPOINT_TYPE]
```

```
                  <subcommand> ...
```

Example A.23. Positional arguments

```
<subcommand>
    alarm-create        Create a new alarm.
    alarm-delete        Delete an alarm.
    alarm-list          List the user's alarms.
    alarm-show          Show an alarm.
    alarm-update        Update an existing alarm.
    meter-list          List the user's meters.
    resource-list       List the resources.
    resource-show       Show the resource.
    sample-create       Create a sample.
    sample-list         List the samples for this meters.
    statistics          List the statistics for this meter.
    help                Display help about this program or one of its
                        subcommands.
```

Example A.24. Optional arguments

```
  -d, --debug           Defaults to env[HEATCLIENT_DEBUG]
  --version             show program's version number and exit
  -d, --debug           Defaults to env[CEILOMETERCLIENT_DEBUG]
  -v, --verbose         Print more verbose output
  -k, --insecure        Explicitly allow ceilometerclient 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 certificate. Without this option
                        ceilometer 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]
  --ceilometer-url CEILOMETER_URL
                        Defaults to env[CEILOMETER_URL]
  --ceilometer-api-version CEILOMETER_API_VERSION
                        Defaults to env[CEILOMETER_API_VERSION] or 2
```

```
--os-service-type OS_SERVICE_TYPE
                      Defaults to env[OS_SERVICE_TYPE]
--os-endpoint-type OS_ENDPOINT_TYPE
                      Defaults to env[OS_ENDPOINT_TYPE]
```

Appendix B. Community support

Table of Contents

Documentation 104
ask.openstack.org 105
OpenStack mailing lists 105
The OpenStack wiki 105
The Launchpad Bugs area 106
The OpenStack IRC channel 106
Documentation feedback 107
OpenStack distribution packages 107

To help you run and use OpenStack, many resources are available. Many OpenStack community members can answer questions and help with bug suspicions. We are constantly improving and adding to the main features of OpenStack, but if you have any problems, do not hesitate to ask. Use the following resources to get OpenStack support and troubleshoot your existing installations.

Documentation

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

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

The following books explain how to install an OpenStack cloud and its components:

- *Installation Guide for Debian 7.0*
- *Installation Guide for openSUSE and SUSE Linux Enterprise Server*
- *Installation Guide for Red Hat Enterprise Linux, CentOS, and Fedora*
- *Installation Guide for Ubuntu 12.04 (LTS)*

The following books explain how to configure and run an OpenStack cloud:

- *Cloud Administrator Guide*
- *Configuration Reference*
- *Operations Guide*
- *High Availability Guide*
- *Security Guide*
- *Virtual Machine Image Guide*

The following books explain how to use the OpenStack dashboard and command-line clients:

- *API Quick Start*
- *End User Guide*
- *Admin User Guide*

The following documentation provides reference and guidance information for the OpenStack APIs:

- OpenStack API Reference
- *OpenStack Block Storage Service API v2 Reference*
- *OpenStack Compute API v2 and Extensions Reference*
- *OpenStack Identity Service API v2.0 Reference*
- *OpenStack Image Service API v2 Reference*
- *OpenStack Networking API v2.0 Reference*
- *OpenStack Object Storage API v1 Reference*

ask.openstack.org

During set up or testing, you might have questions about how to do something or be in a situation where a feature does not work correctly. Use the ask.openstack.org site to ask questions and get answers. When you visit the http://ask.openstack.org site, scan the recently asked questions to see whether your question was already answered. If not, ask a new question. Be sure to give a clear, concise summary in the title and provide as much detail as possible in the description. Paste in your command output or stack traces, link to screen shots, and so on.

OpenStack mailing lists

A great way to get answers and insights is to post your question or scenario to the OpenStack mailing list. You can learn from and help others who might have the same scenario as you. To subscribe or view the archives, go to http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack. You might be interested in the other mailing lists for specific projects or development, which you can find on the wiki. A description of all mailing lists is available at http://wiki.openstack.org/MailingLists.

The OpenStack wiki

The OpenStack wiki contains content on a broad range of topics but some of it sits a bit below the surface. Fortunately, the wiki search feature enables you to search by title or content. If you search for specific information, such as about networking or nova, you can find lots of content. More is being added all the time, so be sure to check back often. You can find the search box in the upper right corner of any OpenStack wiki page.

The Launchpad Bugs area

So you think you've found a bug. That's great! Seriously, it is. The OpenStack community values your set up and testing efforts and wants your feedback. To log a bug, you must sign up for a Launchpad account at https://launchpad.net/+login. You can view existing bugs and report bugs in the Launchpad Bugs area. Use the search feature to determine whether the bug was already reported (or even better, already fixed). If it still seems like your bug is unreported, fill out a bug report.

Some tips:

- Give a clear, concise summary!
- Provide as much detail as possible in the description. Paste in your command output or stack traces, link to screen shots, and so on.
- Be sure to include the software version that you are using, especially if you are using a development branch, such as, `"Grizzly release" vs git commit bc79c3ecc55929bac585d04a03475b72e06a3208`.
- Any deployment specific information is helpful, such as Ubuntu 12.04 or multi-node install.

The Launchpad Bugs areas are available here:

- Bugs: OpenStack Compute (nova)
- Bugs : OpenStack Object Storage (swift)
- Bugs : OpenStack Image Service (glance)
- Bugs : OpenStack Identity (keystone)
- Bugs : OpenStack Dashboard (horizon)
- Bugs : OpenStack Networking (neutron)
- Bugs : OpenStack Orchestration (heat)
- Bugs : OpenStack Telemetry (ceilometer)

The OpenStack IRC channel

The OpenStack community lives and breathes in the #openstack IRC channel on the Freenode network. You can hang out, ask questions, or get immediate feedback for urgent and pressing issues. To install an IRC client or use a browser-based client, go to http://webchat.freenode.net/. You can also use Colloquy (Mac OS X, http://colloquy.info/), mIRC (Windows, http://www.mirc.com/), or XChat (Linux). When you are in the IRC channel and want to share code or command output, the generally accepted method is to use a Paste Bin. The OpenStack project has one at http://paste.openstack.org. Just paste your longer amounts of text or logs in the web form and you get a URL you can paste into the channel. The OpenStack IRC channel is: `#openstack` on `irc.freenode.net`. You can find a list of all OpenStack-related IRC channels at https://wiki.openstack.org/wiki/IRC.

Documentation feedback

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

OpenStack distribution packages

The following Linux distributions provide community-supported packages for OpenStack:

- **Debian:** http://wiki.debian.org/OpenStack
- **CentOS, Fedora, and Red Hat Enterprise Linux:** http://openstack.redhat.com/
- **openSUSE and SUSE Linux Enterprise Server:** http://en.opensuse.org/Portal:OpenStack
- **Ubuntu:** https://wiki.ubuntu.com/ServerTeam/CloudArchive