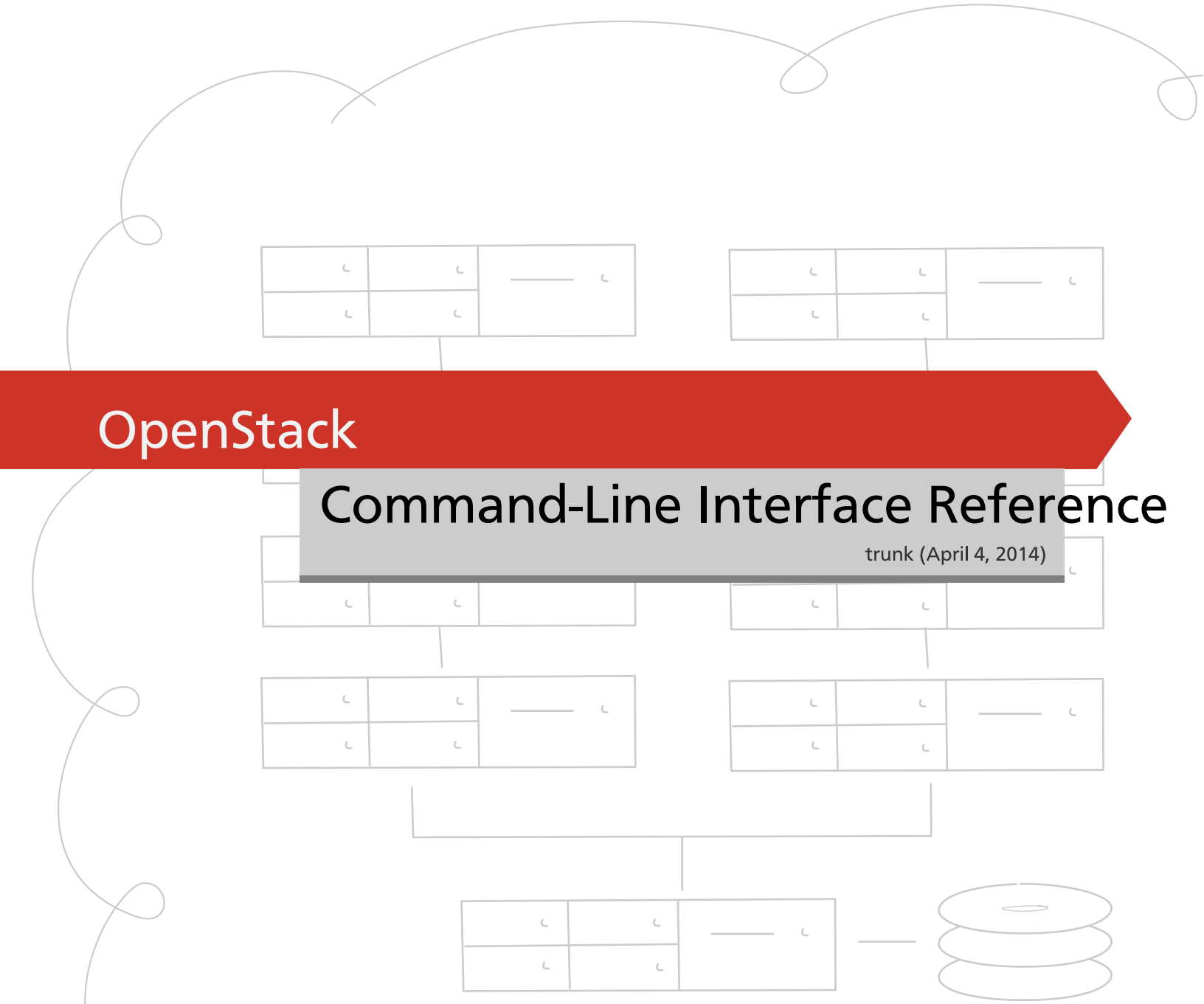

OpenStack

Command-Line Interface Reference

trunk (April 4, 2014)

docs.openstack.org

OpenStack Command-Line Interface Reference

trunk (2014-04-04)

This guide documents the OpenStack command-line clients.

Table of Contents

List of Tables

Preface

Conventions

The OpenStack documentation uses several typesetting conventions:

Admonitions

Admonitions take three forms:

Note

This is a note. The information in a note is usually in the form of a handy tip or reminder.

Important

This is important. The information in an important admonition is something you must be aware of before moving on.

Warning

This is a warning. The information in warnings is critical. Warnings provide additional information about risk of data loss or security issues.

Command prompts

Commands prefixed with the # prompt are to be executed by the `root` user. These examples can also be executed using the **sudo** command, if available.

Commands prefixed with the $ prompt can be executed by any user, including `root`.

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
January 29, 2014	• Initial version.
March 14, 2014	• Added documentation for the neutron-debug command.

1. OpenStack command-line clients

Overview

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

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

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

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

Table 1.1. OpenStack services and clients

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

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

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 1.2. 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. The recommended way to install setuptools on Microsoft Windows is to follow the documentation provided on the setuptools website. Another option is to use the unofficial binary installer maintained by Christoph Gohlke (http://www.lfd.uci.edu/~gohlke/pythonlibs/#setuptools).

pip package	To install the clients on a Linux, Mac OS X or Microsoft Windows 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.
	
	`# easy_install pip`
	
	Microsoft Windows. Make sure that the `C:\Python27\Scripts` directory is defined in the PATH environment variable, and use the **easy_install** command from the setuptools package:
	
	`C:\>easy_install pip`
	
	Another option is to use the unofficial binary installer provided by Christoph Gohlke (http://www.lfd.uci.edu/~gohlke/pythonlibs/#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:

Prerequisite	Description

	`# 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 Linux, Mac OS X or Microsoft Windows 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.sh` file to set required environment variables before you can request OpenStack services through the clients or the APIs.

1. Install each client separately using:

   - For Mac OS X or Linux:

     ```
     # pip install python-PROJECTclient
     ```

   - For Microsoft Windows:

     ```
     C:\>pip install python-PROJECTclient
     ```

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

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

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

   ```
   # pip install python-novaclient
   ```

   To remove the nova client, run this command:

   ```
   # pip uninstall python-novaclient
   ```

Note

To upgrade a package, add the `--upgrade` option to the **pip** command.

For example, to update the `nova` client, run this command:

```
# pip install --upgrade python-novaclient
```

2. Before you can run client commands, you must create and source the `openrc.sh` file to set environment variables. See the section called “The OpenStack RC file” [6].

Get the version for a client

Run this command get the version number for a client:

```
$ PROJECT --version
```

Where *PROJECT* is a project name:

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

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

```
$ nova --version
2.15.0
```

To see the version number for the **keystone** client, run this command:

```
$ keystone --version
0.4.0
```

The OpenStack RC file

To set the required environment variables for the OpenStack command-line clients, you must create an environment file. If your OpenStack installation provides it, you can download the file from the OpenStack dashboard as an administrative user or any other user. This project-specific environment file contains the credentials that all OpenStack services use.

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

Environment variables on Microsoft Windows

Defining environment variables using an environment file is not a common practice on Microsoft Windows. Environment variables are usually defined in the `Advanced` tab of the `System Properties` dialog.

Download and source the OpenStack RC file

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 on the API Access tab. 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.

Create and source the OpenStack RC file

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.

Override environment variable values

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.

2. Block Storage command-line client

The **cinder** client is the command-line interface (CLI) for the OpenStack Block Storage API and its extensions. This chapter documents **cinder** version 1.0.8.

For help on a specific **cinder** command, enter:

```
$ cinder help COMMAND
```

cinder usage

```
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 <volume-api-ver>]
              [--os-cacert <ca-certificate>] [--retries <retries>]
              <subcommand> ...
```

Subcommands

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 volume(s).
encryption-type-create	Create a new encryption type for a volume type (Admin Only).
encryption-type-delete	Delete the encryption type for a volume type (Admin Only).
encryption-type-list	List encryption type information for all volume types (Admin Only).
encryption-type-show	Show the encryption type information for a volume type (Admin Only).
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 volume(s), regardless of the state(s).
list	List all the volumes.
metadata	Set or Delete metadata on a volume.
metadata-show	Show metadata of given volume.
metadata-update-all	Update all metadata of a volume.
migrate	Migrate the volume to the new host.
qos-associate	Associate qos specs with specific volume type.
qos-create	Create a new qos specs.
qos-delete	Delete a specific qos specs.
qos-disassociate	Disassociate qos specs from specific volume type.
qos-disassociate-all	Disassociate qos specs from all of its associations.
qos-get-association	Get all associations of specific qos specs.
qos-key	Set or unset specifications for a qos spec.
qos-list	Get full list of qos specs.

qos-show	Get a specific qos specs.
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.
quota-usage	List the quota usage for a tenant.
rate-limits	Print a list of rate limits for a user
readonly-mode-update	Update volume read-only access mode read_only.
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-metadata	Set or Delete metadata of a snapshot.
snapshot-metadata-show	Show metadata of given snapshot.
snapshot-metadata-update-all	Update all metadata of a snapshot.
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.

cinder 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 volume 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 <volume-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.

cinder absolute-limits command

```
usage: cinder absolute-limits
```

Print a list of absolute limits for a user

cinder availability-zone-list command

```
usage: cinder availability-zone-list
```

List all the availability zones.

cinder backup-create command

```
usage: cinder backup-create [--container <container>]
                            [--display-name <display-name>]
                            [--display-description <display-description>]
                            <volume>
```

Creates a backup.

Positional arguments

<volume> Name or ID of the volume to backup.

Optional arguments

–container <container> Optional Backup container name. (Default=None)

–display-name <display-name> Optional backup name. (Default=None)

–display-description <display-description> Optional backup description. (Default=None)

cinder backup-delete command

```
usage: cinder backup-delete <backup>
```

Remove a backup.

Positional arguments

<backup> Name or ID of the backup to delete.

cinder backup-list command

```
usage: cinder backup-list
```

List all the backups.

cinder backup-restore command

```
usage: cinder backup-restore [--volume-id <volume>] <backup>
```

Restore a backup.

Positional arguments

<backup> ID of the backup to restore.

Optional arguments

–volume-id <volume> Optional ID(or name) of the volume to restore to.

cinder backup-show command

```
usage: cinder backup-show <backup>
```

Show details about a backup.

Positional arguments

<backup> Name or ID of the backup.

cinder create command

```
usage: cinder create [--snapshot-id <snapshot-id>]
                     [--source-volid <source-volid>] [--image-id <image-id>]
                     [--display-name <display-name>]
                     [--display-description <display-description>]
                     [--volume-type <volume-type>]
                     [--availability-zone <availability-zone>]
                     [--metadata [<key=value> [<key=value> ...]]]
                     <size>
```

Add a new volume.

Positional arguments

<size> Size of volume in GB

Optional arguments

–snapshot-id <snapshot-id> Create volume from snapshot id (Optional, Default=None)

–source-volid <source-volid> Create volume from volume id (Optional, Default=None)

–image-id <image-id>	Create volume from image id (Optional, Default=None)
–display-name <display-name>	Volume name (Optional, Default=None)
–display-description <display-description>	Volume description (Optional, Default=None)
–volume-type <volume-type>	Volume type (Optional, Default=None)
–availability-zone <availability-zone>	Availability zone for volume (Optional, Default=None)
–metadata [<key=value> [<key=value> ...]]	Metadata key=value pairs (Optional, Default=None)

cinder credentials command

```
usage: cinder credentials
```

Show user credentials returned from auth.

cinder delete command

```
usage: cinder delete <volume> [<volume> ...]
```

Remove volume(s).

Positional arguments

<volume> Name or ID of the volume(s) to delete.

cinder encryption-type-create command

```
usage: cinder encryption-type-create [--cipher <cipher>]
                                     [--key_size <key_size>]
                                     [--control_location <control_location>]
                                     <volume_type> <provider>
```

Create a new encryption type for a volume type (Admin Only).

Positional arguments

<volume_type> Name or ID of the volume type

<provider> Class providing encryption support (e.g. LuksEncryptor)

Optional arguments

–cipher <cipher>	Encryption algorithm/mode to use (e.g., aes-xts-plain64) (Optional, Default=None)

–key_size <key_size>	Size of the encryption key, in bits (e.g., 128, 256) (Optional, Default=None)
–control_location <control_location>	Notional service where encryption is performed (e.g., front-end=Nova). Values: 'front-end', 'back-end' (Optional, Default=None)

cinder encryption-type-delete command

```
usage: cinder encryption-type-delete <volume_type>
```

Delete the encryption type for a volume type (Admin Only).

Positional arguments

<volume_type> Name or ID of the volume type

cinder encryption-type-list command

```
usage: cinder encryption-type-list
```

List encryption type information for all volume types (Admin Only).

cinder encryption-type-show command

```
usage: cinder encryption-type-show <volume_type>
```

Show the encryption type information for a volume type (Admin Only).

Positional arguments

<volume_type> Name or ID of the volume type

cinder endpoints command

```
usage: cinder endpoints
```

Discover endpoints that get returned from the authenticate services.

cinder extend command

```
usage: cinder extend <volume> <new-size>
```

Attempt to extend the size of an existing volume.

Positional arguments

<volume> Name or ID of the volume to extend.

<new-size> New size of volume in GB

cinder extra-specs-list command

```
usage: cinder extra-specs-list
```

Print a list of current 'volume types and extra specs' (Admin Only).

cinder force-delete command

```
usage: cinder force-delete <volume> [<volume> ...]
```

Attempt forced removal of volume(s), regardless of the state(s).

Positional arguments

<volume> Name or ID of the volume(s) to delete.

cinder list command

```
usage: cinder list [--all-tenants [<0|1>]] [--display-name <display-name>]
                   [--status <status>]
                   [--metadata [<key=value> [<key=value> ...]]]
```

List all the volumes.

Optional arguments

–all-tenants [<0|1>] Display information from all tenants (Admin only).

–display-name <display-name> Filter results by display-name

–status <status> Filter results by status

–metadata [<key=value> [<key=value> ...]] Filter results by metadata

cinder list-extensions command

```
usage: cinder list-extensions
```

List all the os-api extensions that are available.

cinder metadata command

```
usage: cinder metadata <volume> <action> <key=value> [<key=value> ...]
```

Set or Delete metadata on a volume.

Positional arguments

<volume> Name or ID of the volume to update metadata on.

<action> Actions: 'set' or 'unset'

<key=value> Metadata to set/unset (only key is necessary on unset)

cinder metadata-show command

```
usage: cinder metadata-show <volume>
```

Show metadata of given volume.

Positional arguments

<volume> ID of volume

cinder metadata-update-all command

```
usage: cinder metadata-update-all <volume> <key=value> [<key=value> ...]
```

Update all metadata of a volume.

Positional arguments

<volume> ID of the volume to update metadata on.

<key=value> Metadata entry/entries to update.

cinder migrate command

```
usage: cinder migrate [--force-host-copy <True|False>] <volume> <host>
```

Migrate the volume to the new host.

Positional arguments

<volume> ID of the volume to migrate

<host> Destination host

Optional arguments

–force-host-copy <True|False> Optional flag to force the use of the generic host- based migration mechanism, bypassing driver optimizations (Default=False).

cinder qos-associate command

```
usage: cinder qos-associate <qos_specs> <volume_type_id>
```

Associate qos specs with specific volume type.

Positional arguments

<qos_specs> ID of qos_specs.

<volume_type_id> ID of volume type to be associated with.

cinder qos-create command

```
usage: cinder qos-create <name> <key=value> [<key=value> ...]
```

Create a new qos specs.

Positional arguments

<name> Name of the new QoS specs

<key=value> Specifications for QoS

cinder qos-delete command

```
usage: cinder qos-delete [--force <True|False>] <qos_specs>
```

Delete a specific qos specs.

Positional arguments

<qos_specs> ID of the qos_specs to delete.

Optional arguments

–force <True|False> Optional flag that indicates whether to delete specified qos specs even if it is in-use.

cinder qos-disassociate command

```
usage: cinder qos-disassociate <qos_specs> <volume_type_id>
```

Disassociate qos specs from specific volume type.

Positional arguments

<qos_specs> ID of qos_specs.

<volume_type_id> ID of volume type to be associated with.

cinder qos-disassociate-all command

```
usage: cinder qos-disassociate-all <qos_specs>
```

Disassociate qos specs from all of its associations.

Positional arguments

<qos_specs> ID of qos_specs to be operate on.

cinder qos-get-association command

```
usage: cinder qos-get-association <qos_specs>
```

Get all associations of specific qos specs.

Positional arguments

<qos_specs> ID of the qos_specs.

cinder qos-key command

```
usage: cinder qos-key <qos_specs> <action> key=value [key=value ...]
```

Set or unset specifications for a qos spec.

Positional arguments

<qos_specs> ID of qos specs

<action> Actions: 'set' or 'unset'

key=value QoS specs to set/unset (only key is necessary on unset)

cinder qos-list command

```
usage: cinder qos-list
```

Get full list of qos specs.

cinder qos-show command

```
usage: cinder qos-show <qos_specs>
```

Get a specific qos specs.

Positional arguments

<qos_specs> ID of the qos_specs to show.

cinder quota-class-show command

```
usage: cinder quota-class-show <class>
```

List the quotas for a quota class.

Positional arguments

<class> Name of quota class to list the quotas for.

cinder quota-class-update command

```
usage: cinder quota-class-update [--volumes <volumes>]
                                 [--snapshots <snapshots>]
                                 [--gigabytes <gigabytes>]
                                 [--volume-type <volume_type_name>]
                                 <class>
```

Update the quotas for a quota class.

Positional arguments

<class> Name of quota class to set the quotas for.

Optional arguments

–volumes <volumes> New value for the "volumes" quota.

–snapshots <snapshots> New value for the "snapshots" quota.

–gigabytes <gigabytes> New value for the "gigabytes" quota.

–volume-type <volume_type_name> Volume type (Optional, Default=None)

cinder quota-defaults command

```
usage: cinder quota-defaults <tenant_id>
```

List the default quotas for a tenant.

Positional arguments

<tenant_id> UUID of tenant to list the default quotas for.

cinder quota-show command

```
usage: cinder quota-show <tenant_id>
```

List the quotas for a tenant.

Positional arguments

<tenant_id> UUID of tenant to list the quotas for.

cinder quota-update command

```
usage: cinder quota-update [--volumes <volumes>] [--snapshots <snapshots>]
                           [--gigabytes <gigabytes>]
                           [--volume-type <volume_type_name>]
                           <tenant_id>
```

Update the quotas for a tenant.

Positional arguments

<tenant_id> UUID of tenant to set the quotas for.

Optional arguments

–volumes <volumes> New value for the "volumes" quota.

–snapshots <snapshots> New value for the "snapshots" quota.

–gigabytes <gigabytes> New value for the "gigabytes" quota.

–volume-type <volume_type_name> Volume type (Optional, Default=None)

cinder quota-usage command

```
usage: cinder quota-usage <tenant_id>
```

List the quota usage for a tenant.

Positional arguments

<tenant_id> UUID of tenant to list the quota usage for.

cinder rate-limits command

```
usage: cinder rate-limits
```

Print a list of rate limits for a user

cinder readonly-mode-update command

```
usage: cinder readonly-mode-update <volume> <True|true|False|false>
```

Update volume read-only access mode read_only.

Positional arguments

<volume> ID of the volume to update.

<True|true|False|false> Flag to indicate whether to update volume to read-only access mode.

cinder rename command

```
usage: cinder rename [--display-description <display-description>]
                     <volume> [<display-name>]
```

Rename a volume.

Positional arguments

<volume> Name or ID of the volume to rename.

<display-name> New display-name for the volume.

Optional arguments

–display-description <display-description> Optional volume description. (Default=None)

cinder reset-state command

```
usage: cinder reset-state [--state <state>] <volume> [<volume> ...]
```

Explicitly update the state of a volume.

Positional arguments

<volume> Name or ID of the volume to modify.

Optional arguments

–state <state> Indicate which state to assign the volume. Options include available, error, creating, deleting, error_deleting. If no state is provided, available will be used.

cinder service-disable command

```
usage: cinder service-disable <hostname> <binary>
```

```
```

Disable the service.

Positional arguments

<hostname> Name of host.

<binary> Service binary.

cinder service-enable command

```
usage: cinder service-enable <hostname> <binary>
```

Enable the service.

Positional arguments

<hostname> Name of host.

<binary> Service binary.

cinder service-list command

```
usage: cinder service-list [--host <hostname>] [--binary <binary>]
```

List all the services. Filter by host & service binary.

Optional arguments

–host <hostname> Name of host.

–binary <binary> Service binary.

cinder show command

```
usage: cinder show <volume>
```

Show details about a volume.

Positional arguments

<volume> Name or ID of the volume.

cinder snapshot-create command

```
usage: cinder snapshot-create [--force <True|False>]
                              [--display-name <display-name>]
                              [--display-description <display-description>]
                              <volume>
```

```
```

Add a new snapshot.

Positional arguments

<volume> Name or ID of the volume to snapshot

Optional arguments

–force <True|False>	Optional flag to indicate whether to snapshot a volume even if it's attached to an instance. (Default=False)
–display-name <display-name>	Optional snapshot name. (Default=None)
–display-description <display-description>	Optional snapshot description. (Default=None)

cinder snapshot-delete command

```
usage: cinder snapshot-delete <snapshot>
```

Remove a snapshot.

Positional arguments

<snapshot> Name or ID of the snapshot to delete.

cinder snapshot-list command

```
usage: cinder snapshot-list [--all-tenants [<0|1>]]
                            [--display-name <display-name>]
                            [--status <status>] [--volume-id <volume-id>]
```

List all the snapshots.

Optional arguments

–all-tenants [<0|1>]	Display information from all tenants (Admin only).
–display-name <display-name>	Filter results by display-name
–status <status>	Filter results by status
–volume-id <volume-id>	Filter results by volume-id

cinder snapshot-metadata command

```
usage: cinder snapshot-metadata <snapshot> <action> <key=value>
                                [<key=value> ...]
```

Set or Delete metadata of a snapshot.

Positional arguments

<snapshot> ID of the snapshot to update metadata on.

<action> Actions: 'set' or 'unset'

<key=value> Metadata to set/unset (only key is necessary on unset)

cinder snapshot-metadata-show command

```
usage: cinder snapshot-metadata-show <snapshot>
```

Show metadata of given snapshot.

Positional arguments

<snapshot> ID of snapshot

cinder snapshot-metadata-update-all command

```
usage: cinder snapshot-metadata-update-all <snapshot> <key=value>
                                           [<key=value> ...]
```

Update all metadata of a snapshot.

Positional arguments

<snapshot> ID of the snapshot to update metadata on.

<key=value> Metadata entry/entries to update.

cinder snapshot-rename command

```
usage: cinder snapshot-rename [--display-description <display-description>]
                              <snapshot> [<display-name>]
```

Rename a snapshot.

Positional arguments

<snapshot> Name or ID of the snapshot.

<display-name> New display-name for the snapshot.

Optional arguments

–display-description <display-description> Optional snapshot description. (Default=None)

cinder snapshot-reset-state command

```
usage: cinder snapshot-reset-state [--state <state>]
                                   <snapshot> [<snapshot> ...]
```

Explicitly update the state of a snapshot.

Positional arguments

<snapshot> Name or ID of the snapshot to modify.

Optional arguments

–state <state> Indicate which state to assign the snapshot. Options include available, error, creating, deleting, error_deleting. If no state is provided, available will be used.

cinder snapshot-show command

```
usage: cinder snapshot-show <snapshot>
```

Show details about a snapshot.

Positional arguments

<snapshot> Name or ID of the snapshot.

cinder transfer-accept command

```
usage: cinder transfer-accept <transfer> <auth_key>
```

Accepts a volume transfer.

Positional arguments

<transfer> ID of the transfer to accept.

<auth_key> Auth key of the transfer to accept.

cinder transfer-create command

```
usage: cinder transfer-create [--display-name <display-name>] <volume>
```

Creates a volume transfer.

Positional arguments

<volume> Name or ID of the volume to transfer.

Optional arguments

–display-name <display-name> Optional transfer name. (Default=None)

cinder transfer-delete command

```
usage: cinder transfer-delete <transfer>
```

Undo a transfer.

Positional arguments

<transfer> Name or ID of the transfer to delete.

cinder transfer-list command

```
usage: cinder transfer-list
```

List all the transfers.

cinder transfer-show command

```
usage: cinder transfer-show <transfer>
```

Show details about a transfer.

Positional arguments

<transfer> Name or ID of the transfer to accept.

cinder type-create command

```
usage: cinder type-create <name>
```

Create a new volume type.

Positional arguments

<name> Name of the new volume type

cinder type-delete command

```
usage: cinder type-delete <id>
```

Delete a specific volume type.

Positional arguments

<id> Unique ID of the volume type to delete

cinder type-key command

```
usage: cinder type-key <vtype> <action> [<key=value> [<key=value> ...]]
```

Set or unset extra_spec for a volume type.

Positional arguments

<vtype> Name or ID of the volume type

<action> Actions: 'set' or 'unset'

<key=value> Extra_specs to set/unset (only key is necessary on unset)

cinder type-list command

```
usage: cinder type-list
```

Print a list of available 'volume types'.

cinder upload-to-image command

```
usage: cinder upload-to-image [--force <True|False>]
                              [--container-format <container-format>]
                              [--disk-format <disk-format>]
                              <volume> <image-name>
```

Upload volume to image service as image.

Positional arguments

<volume> Name or ID of the volume to upload to an image

<image-name> Name for created image

Optional arguments

–force <True|False> Optional flag to indicate whether to upload a volume even if it's attached to an instance. (Default=False)

–container-format <container-format> Optional type for container format (Default=bare)

–disk-format <disk-format> Optional type for disk format (Default=raw)

3. Compute command-line client

The **nova** client is the command-line interface (CLI) for the OpenStack Compute API and its extensions. This chapter documents **nova** version 2.17.0.

For help on a specific **nova** command, enter:

```
$ nova help COMMAND
```

nova usage

```
usage: nova [--version] [--debug] [--os-cache] [--timings]
            [--timeout <seconds>] [--os-auth-token OS_AUTH_TOKEN]
            [--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> ...
```

Subcommands

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	*DEPRECATED*, use floating-ip-associate instead.
add-secgroup	Add a Security Group to a server.
agent-create	Create new agent build.
agent-delete	Delete existing agent build.
agent-list	List all builds.
agent-modify	Modify 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.
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 server by creating 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.
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	Retrieve 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-associate	Associate a floating IP address to a server.
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-disassociate	Disassociate a floating IP address from a server.
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-rdp-console	Get a rdp console to 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 servers 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 a server.
interface-detach	Detach a network interface from a server.
interface-list	List interfaces attached to a server.
keypair-add	Create a new key pair for use with servers.
keypair-delete	Delete keypair given by its name.
keypair-list	Print a list of keypairs for a user
keypair-show	Show details about the given keypair.

list	List active servers.
list-secgroup	List Security Group(s) of a server.
live-migration	Migrate running server 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/user so their quota will Revert back to default.
quota-show	List the quotas for a tenant/user.
quota-update	Update the quotas for a tenant/user.
rate-limits	Print a list of rate limits for a user
reboot	Reboot a server.
rebuild	Shutdown, re-image, and re-boot a server.
refresh-network	Refresh server network information.
remove-fixed-ip	Remove an IP address from a server.
remove-floating-ip	*DEPRECATED*, use floating-ip-disassociate instead.
remove-secgroup	Remove a Security Group from a server.
rename	Rename a server.

rescue	Rescue a server.
reset-network	Reset network of a server.
reset-state	Reset the state of a server.
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	Delete 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.
shelve	Shelve a server.
shelve-offload	Remove a shelved server from the compute node.
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.
unshelve	Unshelve 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 volume(s).
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	Fetch the x509 root cert.
bash-completion	Prints all of the commands and options to stdout so that the nova.bash_completion script doesn't have to hard code them.
help	Display help about this program or one of its subcommands.
force-delete	Force delete a server.
restore	Restore a soft-deleted server.

net	Show a network
net-create	Create a network
net-delete	Delete a network
net-list	List networks
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 list of available baremetal nodes.
baremetal-node-show	Show information about a baremetal node.
host-evacuate	Evacuate all instances from failed host to specified one.
instance-action	Show an action.
instance-action-list	List actions on a server.
migration-list	Print a list of migrations.
host-servers-migrate	Migrate all instances of the specified host to other available hosts.
cell-capacities	Get cell capacities for all cells or a given cell.
cell-show	Show details of a given cell.
host-meta	Set or Delete metadata on all instances of a host.
list-extensions	List all the os-api extensions that are available.

nova optional arguments

–version	show program's version number and exit
–debug	Print debugging output
–os-cache	Use the auth token cache. Defaults to False if `env[OS_CACHE]` is not set.
–timings	Print call timing info
–timeout <seconds>	Set HTTP call timeout (in seconds)

–os-auth-token OS_AUTH_TOKEN	Defaults to `env[OS_AUTH_TOKEN]`
–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 or 3, 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

nova absolute-limits command

```
usage: nova absolute-limits [--tenant [<tenant>]] [--reserved]
```

Print a list of absolute limits for a user

Optional arguments

–tenant [<tenant>]	Display information from single tenant (Admin only).
–reserved	Include reservations count.

nova add-fixed-ip command

```
usage: nova add-fixed-ip <server> <network-id>
```

Add new IP address on a network to server.

Positional arguments

<server> Name or ID of server.

<network-id> Network ID.

nova add-secgroup command

```
usage: nova add-secgroup <server> <secgroup>
```

Add a Security Group to a server.

Positional arguments

<server> Name or ID of server.

<secgroup> Name of Security Group.

nova agent-create command

```
usage: nova agent-create <os> <architecture> <version> <url> <md5hash>
                         <hypervisor>
```

Create new agent build.

Positional arguments

<os> type of os.

<architecture> type of architecture

<version> version

<url> url

<md5hash> md5 hash

<hypervisor> type of hypervisor.

nova agent-delete command

```
usage: nova agent-delete <id>
```

Delete existing agent build.

Positional arguments

<id> id of the agent-build

nova agent-list command

```
usage: nova agent-list [--hypervisor <hypervisor>]
```

List all builds.

Optional arguments

–hypervisor <hypervisor> type of hypervisor.

nova agent-modify command

```
usage: nova agent-modify <id> <version> <url> <md5hash>
```

Modify existing agent build.

Positional arguments

<id> id of the agent-build

<version> version

<url> url

<md5hash> md5hash

nova aggregate-add-host command

```
usage: nova aggregate-add-host <aggregate> <host>
```

Add the host to the specified aggregate.

Positional arguments

<aggregate> Name or ID of aggregate.

<host> The host to add to the aggregate.

nova aggregate-create command

```
usage: nova aggregate-create <name> [<availability-zone>]
```

Create a new aggregate with the specified details.

Positional arguments

<name> Name of aggregate.

<availability-zone> The availability zone of the aggregate (optional).

nova aggregate-delete command

```
usage: nova aggregate-delete <aggregate>
```

Delete the aggregate.

Positional arguments

<aggregate> Name or ID of aggregate to delete.

nova aggregate-details command

```
usage: nova aggregate-details <aggregate>
```

Show details of the specified aggregate.

Positional arguments

<aggregate> Name or ID of aggregate.

nova aggregate-list command

```
usage: nova aggregate-list
```

Print a list of all aggregates.

nova aggregate-remove-host command

```
usage: nova aggregate-remove-host <aggregate> <host>
```

Remove the specified host from the specified aggregate.

Positional arguments

<aggregate> Name or ID of aggregate.

<host> The host to remove from the aggregate.

nova aggregate-set-metadata command

```
usage: nova aggregate-set-metadata <aggregate> <key=value> [<key=value> ...]
```

```
```

Update the metadata associated with the aggregate.

Positional arguments

<aggregate> Name or ID of aggregate to update.

<key=value> Metadata to add/update to aggregate

nova aggregate-update command

```
usage: nova aggregate-update <aggregate> <name> [<availability-zone>]
```

Update the aggregate's name and optionally availability zone.

Positional arguments

<aggregate> Name or ID of aggregate to update.

<name> Name of aggregate.

<availability-zone> The availability zone of the aggregate.

nova availability-zone-list command

```
usage: nova availability-zone-list
```

List all the availability zones.

nova backup command

```
usage: nova backup <server> <name> <backup-type> <rotation>
```

Backup a server by creating a 'backup' type snapshot.

Positional arguments

<server> Name or ID of server.

<name> Name of the backup image.

<backup-type> The backup type, like "daily" or "weekly".

<rotation> Int parameter representing how many backups to keep around.

nova baremetal-interface-add command

```
usage: nova baremetal-interface-add [--datapath_id <datapath_id>]
```

```
                                    [--port_no <port_no>]
                                    <node> <address>
```

Add a network interface to a baremetal node.

Positional arguments

<node> ID of node

<address> MAC address of interface

Optional arguments

–datapath_id <datapath_id> OpenFlow Datapath ID of interface

–port_no <port_no> OpenFlow port number of interface

nova baremetal-interface-list command

```
usage: nova baremetal-interface-list <node>
```

List network interfaces associated with a baremetal node.

Positional arguments

<node> ID of node

nova baremetal-interface-remove command

```
usage: nova baremetal-interface-remove <node> <address>
```

Remove a network interface from a baremetal node.

Positional arguments

<node> ID of node

<address> MAC address of interface

nova baremetal-node-create command

```
usage: nova baremetal-node-create [--pm_address <pm_address>]
                                  [--pm_user <pm_user>]
                                  [--pm_password <pm_password>]
                                  [--terminal_port <terminal_port>]
                                  <service_host> <cpus> <memory_mb> <local_gb>
                                  <prov_mac_address>
```

Create a baremetal node.

Positional arguments

<service_host>	Name of nova compute host which will control this baremetal node
<cpus>	Number of CPUs in the node
<memory_mb>	Megabytes of RAM in the node
<local_gb>	Gigabytes of local storage in the node
<prov_mac_address>	MAC address to provision the node

Optional arguments

–pm_address <pm_address>	Power management IP for the node
–pm_user <pm_user>	Username for the node's power management
–pm_password <pm_password>	Password for the node's power management
–terminal_port <terminal_port>	ShellInABox port?

nova baremetal-node-delete command

```
usage: nova baremetal-node-delete <node>
```

Remove a baremetal node and any associated interfaces.

Positional arguments

<node> ID of the node to delete.

nova baremetal-node-list command

```
usage: nova baremetal-node-list
```

Print list of available baremetal nodes.

nova baremetal-node-show command

```
usage: nova baremetal-node-show <node>
```

Show information about a baremetal node.

Positional arguments

<node> ID of node

nova boot command

```
usage: nova boot [--flavor <flavor>] [--image <image>]
                 [--image-with <key=value>] [--boot-volume <volume_id>]
                 [--snapshot <snapshot_id>] [--num-instances <number>]
                 [--meta <key=value>] [--file <dst-path=src-path>]
                 [--key-name <key-name>] [--user-data <user-data>]
                 [--availability-zone <availability-zone>]
                 [--security-groups <security-groups>]
                 [--block-device-mapping <dev-name=mapping>]
                 [--block-device key1=value1[,key2=value2...]]
                 [--swap <swap_size>]
                 [--ephemeral size=<size>[,format=<format>]]
                 [--hint <key=value>]
                 [--nic <net-id=net-uuid,v4-fixed-ip=ip-addr,port-id=port-
uuid>]
                 [--config-drive <value>] [--poll]
                 <name>
```

Boot a new server.

Positional arguments

<name> Name for the new server

Optional arguments

–flavor <flavor>	Name or ID of flavor (see 'nova flavor-list').
–image <image>	Name or ID of image (see 'nova image-list').
–image-with <key=value>	Image metadata property (see 'nova image-show').
–boot-volume <volume_id>	Volume ID to boot from.
–snapshot <snapshot_id>	Snapshot ID to boot from (will create a volume).
–num-instances <number>	boot multiple servers at a time (limited by quota).
–meta <key=value>	Record arbitrary key/value metadata to /meta.js on the new server. Can be specified multiple times.
–file <dst-path=src-path>	Store arbitrary files from <src-path> locally to <dst-path> on the new server. You may store up to 5 files.
–key-name <key-name>	Key name of keypair that should be created earlier with the command keypair-add
–user-data <user-data>	user data file to pass to be exposed by the metadata server.
–availability-zone <availability-zone>	The availability zone for server placement.

–security-groups <security-groups>	Comma separated list of security group names.
–block-device-mapping <dev-name=mapping>	Block device mapping in the format <dev-name>=<id>:<type>:<size(GB)>:<delete-on-terminate>.
–block-device	key1=value1[,key2=value2...] Block device mapping with the keys: id=image_id, snapshot_id or volume_id, source=source type (image, snapshot, volume or blank), dest=destination type of the block device (volume or local), bus=device's bus, device=name of the device (e.g. vda, xda, ...), size=size of the block device in GB, format=device will be formatted (e.g. swap, ext3, ntfs, ...), bootindex=integer used for ordering the boot disks, type=device type (e.g. disk, cdrom, ...) and shutdown=shutdown behaviour (either preserve or remove).
–swap <swap_size>	Create and attach a local swap block device of <swap_size> MB.
–ephemeral	size=<size>[,format=<format>] Create and attach a local ephemeral block device of <size> GB and format it to <format>.
–hint <key=value>	Send arbitrary key/value pairs to the scheduler for custom use.
–nic <net-id=net-uuid,v4-fixed-ip=ip-addr,port-id=port-uuid>	Create a NIC on the server. Specify option multiple times to create multiple NICs. net-id: attach NIC to network with this UUID (required if no port-id), v4 -fixed-ip: IPv4 fixed address for NIC (optional), port-id: attach NIC to port with this UUID (required if no net-id)
–config-drive <value>	Enable config drive
–poll	Blocks while server builds so progress can be reported.

nova cell-capacities command

```
usage: nova cell-capacities [--cell <cell-name>]
```

Get cell capacities for all cells or a given cell.

Optional arguments

–cell <cell-name>	Name of the cell to get the capacities.

nova cell-show command

```
usage: nova cell-show <cell-name>
```

Show details of a given cell.

Positional arguments

<cell-name> Name of the cell.

nova clear-password command

```
usage: nova clear-password <server>
```

Clear password for a server.

Positional arguments

<server> Name or ID of server.

nova cloudpipe-configure command

```
usage: nova cloudpipe-configure <ip address> <port>
```

Update the VPN IP/port of a cloudpipe instance.

Positional arguments

<ip address> New IP Address.

<port> New Port.

nova cloudpipe-create command

```
usage: nova cloudpipe-create <project_id>
```

Create a cloudpipe instance for the given project.

Positional arguments

<project_id> UUID of the project to create the cloudpipe for.

nova cloudpipe-list command

```
usage: nova cloudpipe-list
```

Print a list of all cloudpipe instances.

nova console-log command

```
usage: nova console-log [--length <length>] <server>
```

```
```

Get console log output of a server.

Positional arguments

<server> Name or ID of server.

Optional arguments

–length <length> Length in lines to tail.

nova credentials command

```
usage: nova credentials [--wrap <integer>]
```

Show user credentials returned from auth.

Optional arguments

–wrap <integer> wrap PKI tokens to a specified length, or 0 to disable

nova delete command

```
usage: nova delete <server> [<server> ...]
```

Immediately shut down and delete specified server(s).

Positional arguments

<server> Name or ID of server(s).

nova diagnostics command

```
usage: nova diagnostics <server>
```

Retrieve server diagnostics.

Positional arguments

<server> Name or ID of server.

nova dns-create command

```
usage: nova dns-create [--type <type>] <ip> <name> <domain>
```

Create a DNS entry for domain, name and ip.

Positional arguments

<ip> ip address

<name> DNS name

<domain> DNS domain

Optional arguments

–type <type> dns type (e.g. "A")

nova dns-create-private-domain command

```
usage: nova dns-create-private-domain
                                     [--availability-zone <availability-
zone>]
                                     <domain>
```

Create the specified DNS domain.

Positional arguments

<domain> DNS domain

Optional arguments

–availability-zone <availability-zone> Limit access to this domain to servers in the specified availability zone.

nova dns-create-public-domain command

```
usage: nova dns-create-public-domain [--project <project>] <domain>
```

Create the specified DNS domain.

Positional arguments

<domain> DNS domain

Optional arguments

–project <project> Limit access to this domain to users of the specified project.

nova dns-delete command

```
usage: nova dns-delete <domain> <name>
```

Delete the specified DNS entry.

Positional arguments

<domain> DNS domain

<name> DNS name

nova dns-delete-domain command

```
usage: nova dns-delete-domain <domain>
```

Delete the specified DNS domain.

Positional arguments

<domain> DNS domain

nova dns-domains command

```
usage: nova dns-domains
```

Print a list of available dns domains.

nova dns-list command

```
usage: nova dns-list [--ip <ip>] [--name <name>] <domain>
```

List current DNS entries for domain and ip or domain and name.

Positional arguments

<domain> DNS domain

Optional arguments

–ip <ip> ip address

–name <name> DNS name

nova endpoints command

```
usage: nova endpoints
```

Discover endpoints that get returned from the authenticate services.

nova evacuate command

```
usage: nova evacuate [--password <password>] [--on-shared-storage]
                     <server> <host>
```

```
```

Evacuate server from failed host to specified one.

Positional arguments

<server> Name or ID of server.

<host> Name or ID of target host.

Optional arguments

–password <password> Set the provided password on the evacuated server. Not applicable with on-shared-storage flag

–on-shared-storage Specifies whether server files are located on shared storage

nova fixed-ip-get command

```
usage: nova fixed-ip-get <fixed_ip>
```

Retrieve info on a fixed ip.

Positional arguments

<fixed_ip> Fixed IP Address.

nova fixed-ip-reserve command

```
usage: nova fixed-ip-reserve <fixed_ip>
```

Reserve a fixed IP.

Positional arguments

<fixed_ip> Fixed IP Address.

nova fixed-ip-unreserve command

```
usage: nova fixed-ip-unreserve <fixed_ip>
```

Unreserve a fixed IP.

Positional arguments

<fixed_ip> Fixed IP Address.

nova flavor-access-add command

```
usage: nova flavor-access-add <flavor> <tenant_id>
```

```
```

Add flavor access for the given tenant.

Positional arguments

<flavor> Flavor name or ID to add access for the given tenant.

<tenant_id> Tenant ID to add flavor access for.

nova flavor-access-list command

```
usage: nova flavor-access-list [--flavor <flavor>] [--tenant <tenant_id>]
```

Print access information about the given flavor.

Optional arguments

–flavor <flavor> Filter results by flavor name or ID.

–tenant <tenant_id> Filter results by tenant ID.

nova flavor-access-remove command

```
usage: nova flavor-access-remove <flavor> <tenant_id>
```

Remove flavor access for the given tenant.

Positional arguments

<flavor> Flavor name or ID to remove access for the given tenant.

<tenant_id> Tenant ID to remove flavor access for.

nova flavor-create command

```
usage: nova flavor-create [--ephemeral <ephemeral>] [--swap <swap>]
                          [--rxtx-factor <factor>] [--is-public <is-public>]
                          <name> <id> <ram> <disk> <vcpus>
```

Create a new flavor

Positional arguments

<name> Name of the new flavor

<id> Unique ID (integer or UUID) for the new flavor. If specifying 'auto', a UUID will be generated as id

<ram> Memory size in MB

<disk> Disk size in GB

<vcpus> Number of vcpus

Optional arguments

–ephemeral <ephemeral> Ephemeral space size in GB (default 0)

–swap <swap> Swap space size in MB (default 0)

–rxtx-factor <factor> RX/TX factor (default 1)

–is-public <is-public> Make flavor accessible to the public (default true)

nova flavor-delete command

```
usage: nova flavor-delete <flavor>
```

Delete a specific flavor

Positional arguments

<flavor> Name or ID of the flavor to delete

nova flavor-key command

```
usage: nova flavor-key <flavor> <action> <key=value> [<key=value> ...]
```

Set or unset extra_spec for a flavor.

Positional arguments

<flavor> Name or ID of flavor

<action> Actions: 'set' or 'unset'

<key=value> Extra_specs to set/unset (only key is necessary on unset)

nova flavor-list command

```
usage: nova flavor-list [--extra-specs] [--all]
```

Print a list of available 'flavors' (sizes of servers).

Optional arguments

–extra-specs Get extra-specs of each flavor.

–all Display all flavors (Admin only).

nova flavor-show command

```
usage: nova flavor-show <flavor>
```

Show details about the given flavor.

Positional arguments

<flavor> Name or ID of flavor

nova floating-ip-associate command

```
usage: nova floating-ip-associate [--fixed-address <fixed_address>]
                                  <server> <address>
```

Associate a floating IP address to a server.

Positional arguments

<server> Name or ID of server.

<address> IP Address.

Optional arguments

–fixed-address <fixed_address> Fixed IP Address to associate with.

nova floating-ip-bulk-create command

```
usage: nova floating-ip-bulk-create [--pool <pool>] [--interface <interface>]
                                    <range>
```

Bulk create floating ips by range.

Positional arguments

<range> Address range to create

Optional arguments

–pool <pool> Pool for new Floating IPs

–interface <interface> Interface for new Floating IPs

nova floating-ip-bulk-delete command

```
usage: nova floating-ip-bulk-delete <range>
```

Bulk delete floating ips by range.

Positional arguments

<range> Address range to delete

nova floating-ip-bulk-list command

```
usage: nova floating-ip-bulk-list [--host <host>]
```

List all floating ips.

Optional arguments

–host <host> Filter by host

nova floating-ip-create command

```
usage: nova floating-ip-create [<floating-ip-pool>]
```

Allocate a floating IP for the current tenant.

Positional arguments

<floating-ip-pool> Name of Floating IP Pool. (Optional)

nova floating-ip-delete command

```
usage: nova floating-ip-delete <address>
```

De-allocate a floating IP.

Positional arguments

<address> IP of Floating Ip.

nova floating-ip-disassociate command

```
usage: nova floating-ip-disassociate <server> <address>
```

Disassociate a floating IP address from a server.

Positional arguments

<server> Name or ID of server.

<address> IP Address.

nova floating-ip-list command

```
usage: nova floating-ip-list
```

List floating ips for this tenant.

nova floating-ip-pool-list command

```
usage: nova floating-ip-pool-list
```

List all floating ip pools.

nova force-delete command

```
usage: nova force-delete <server>
```

Force delete a server.

Positional arguments

<server> Name or ID of server.

nova get-password command

```
usage: nova get-password <server> [<private-key>]
```

Get password for a server.

Positional arguments

<server> Name or ID of server.

<private-key> Private key (used locally to decrypt password) (Optional). When specified, the command displays the clear (decrypted) VM password. When not specified, the ciphered VM password is displayed.

nova get-rdp-console command

```
usage: nova get-rdp-console <server> <console-type>
```

Get a rdp console to a server.

Positional arguments

<server> Name or ID of server.

<console-type> Type of rdp console ("rdp-html5").

nova get-spice-console command

```
usage: nova get-spice-console <server> <console-type>
```

Get a spice console to a server.

Positional arguments

<server> Name or ID of server.

<console-type> Type of spice console ("spice-html5").

nova get-vnc-console command

```
usage: nova get-vnc-console <server> <console-type>
```

Get a vnc console to a server.

Positional arguments

<server> Name or ID of server.

<console-type> Type of vnc console ("novnc" or "xvpvnc").

nova host-action command

```
usage: nova host-action [--action <action>] <hostname>
```

Perform a power action on a host.

Positional arguments

<hostname> Name of host.

Optional arguments

–action <action> A power action: startup, reboot, or shutdown.

nova host-describe command

```
usage: nova host-describe <hostname>
```

Describe a specific host.

Positional arguments

<hostname> Name of host.

nova host-evacuate command

```
usage: nova host-evacuate [--target_host <target_host>] [--on-shared-storage]
                          <host>
```

Evacuate all instances from failed host to specified one.

Positional arguments

<host> Name of host.

Optional arguments

–target_host <target_host> Name of target host.

–on-shared-storage Specifies whether all instances files are on shared storage

nova host-list command

```
usage: nova host-list [--zone <zone>]
```

List all hosts by service.

Optional arguments

–zone <zone> Filters the list, returning only those hosts in the availability zone <zone>.

nova host-meta command

```
usage: nova host-meta <host> <action> <key=value> [<key=value> ...]
```

Set or Delete metadata on all instances of a host.

Positional arguments

<host> Name of host.

<action> Actions: 'set' or 'delete'

<key=value> Metadata to set or delete (only key is necessary on delete)

nova host-servers-migrate command

```
usage: nova host-servers-migrate <host>
```

Migrate all instances of the specified host to other available hosts.

Positional arguments

<host> Name of host.

nova host-update command

```
usage: nova host-update [--status <enable|disable>]
                        [--maintenance <enable|disable>]
                        <hostname>
```

Update host settings.

Positional arguments

<hostname> Name of host.

Optional arguments

–status <enable|disable> Either enable or disable a host.

–maintenance <enable|disable> Either put or resume host to/from maintenance.

nova hypervisor-list command

```
usage: nova hypervisor-list [--matching <hostname>]
```

List hypervisors.

Optional arguments

–matching <hostname> List hypervisors matching the given <hostname>.

nova hypervisor-servers command

```
usage: nova hypervisor-servers <hostname>
```

List servers belonging to specific hypervisors.

Positional arguments

<hostname> The hypervisor hostname (or pattern) to search for.

nova hypervisor-show command

```
usage: nova hypervisor-show <hypervisor>
```

Display the details of the specified hypervisor.

Positional arguments

<hypervisor> Name or ID of the hypervisor to show the details of.

nova hypervisor-stats command

```
usage: nova hypervisor-stats
```

Get hypervisor statistics over all compute nodes.

nova hypervisor-uptime command

```
usage: nova hypervisor-uptime <hypervisor>
```

Display the uptime of the specified hypervisor.

Positional arguments

<hypervisor> Name or ID of the hypervisor to show the uptime of.

nova image-create command

```
usage: nova image-create [--show] [--poll] <server> <name>
```

Create a new image by taking a snapshot of a running server.

Positional arguments

<server> Name or ID of server.

<name> Name of snapshot.

Optional arguments

–show Print image info.

–poll Blocks while server snapshots so progress can be reported.

nova image-delete command

```
usage: nova image-delete <image> [<image> ...]
```

Delete specified image(s).

Positional arguments

<image> Name or ID of image(s).

nova image-list command

```
usage: nova image-list [--limit <limit>]
```

Print a list of available images to boot from.

Optional arguments

–limit <limit> number of images to return per request

nova image-meta command

```
usage: nova image-meta <image> <action> <key=value> [<key=value> ...]
```

Set or Delete metadata on an image.

Positional arguments

<image> Name or ID of image

<action> Actions: 'set' or 'delete'

<key=value> Metadata to add/update or delete (only key is necessary on delete)

nova image-show command

```
usage: nova image-show <image>
```

Show details about the given image.

Positional arguments

<image> Name or ID of image

nova instance-action command

```
usage: nova instance-action <server> <request_id>
```

Show an action.

Positional arguments

<server> Name or UUID of the server to show an action for.

<request_id> Request ID of the action to get.

nova instance-action-list command

```
usage: nova instance-action-list <server>
```

```
```

List actions on a server.

Positional arguments

<server> Name or UUID of the server to list actions for.

nova interface-attach command

```
usage: nova interface-attach [--port-id <port_id>] [--net-id <net_id>]
                             [--fixed-ip <fixed_ip>]
                             <server>
```

Attach a network interface to a server.

Positional arguments

<server> Name or ID of server.

Optional arguments

–port-id <port_id> Port ID.

–net-id <net_id> Network ID

–fixed-ip <fixed_ip> Requested fixed IP.

nova interface-detach command

```
usage: nova interface-detach <server> <port_id>
```

Detach a network interface from a server.

Positional arguments

<server> Name or ID of server.

<port_id> Port ID.

nova interface-list command

```
usage: nova interface-list <server>
```

List interfaces attached to a server.

Positional arguments

<server> Name or ID of server.

nova keypair-add command

```
usage: nova keypair-add [--pub-key <pub-key>] <name>
```

Create a new key pair for use with servers.

Positional arguments

<name> Name of key.

Optional arguments

–pub-key <pub-key> Path to a public ssh key.

nova keypair-delete command

```
usage: nova keypair-delete <name>
```

Delete keypair given by its name.

Positional arguments

<name> Keypair name to delete.

nova keypair-list command

```
usage: nova keypair-list
```

Print a list of keypairs for a user

nova keypair-show command

```
usage: nova keypair-show <keypair>
```

Show details about the given keypair.

Positional arguments

<keypair> Name or ID of keypair

nova list command

```
usage: nova list [--reservation-id <reservation-id>] [--ip <ip-regexp>]
                 [--ip6 <ip6-regexp>] [--name <name-regexp>]
                 [--instance-name <name-regexp>] [--status <status>]
                 [--flavor <flavor>] [--image <image>] [--host <hostname>]
                 [--all-tenants [<0|1>]] [--tenant [<tenant>]] [--deleted]
                 [--fields <fields>] [--minimal]
```

List active servers.

Optional arguments

–reservation-id <reservation-id> Only return servers that match reservation-id.

–ip <ip-regexp> Search with regular expression match by IP address (Admin only).

–ip6 <ip6-regexp> Search with regular expression match by IPv6 address (Admin only).

–name <name-regexp> Search with regular expression match by name

–instance-name <name-regexp> Search with regular expression match by server name (Admin only).

–status <status> Search by server status

–flavor <flavor> Search by flavor name or ID

–image <image> Search by image name or ID

–host <hostname> Search servers by hostname to which they are assigned (Admin only).

–all-tenants [<0|1>] Display information from all tenants (Admin only).

–tenant [<tenant>] Display information from single tenant (Admin only).

–deleted Only display deleted servers (Admin only).

–fields <fields> Comma-separated list of fields to display. Use the show command to see which fields are available.

–minimal Get only uuid and name.

nova list-extensions command

```
usage: nova list-extensions
```

List all the os-api extensions that are available.

nova list-secgroup command

```
usage: nova list-secgroup <server>
```

List Security Group(s) of a server.

Positional arguments

<server> Name or ID of server.

nova live-migration command

```
usage: nova live-migration [--block-migrate] [--disk-over-commit]
                           <server> [<host>]
```

Migrate running server to a new machine.

Positional arguments

<server> Name or ID of server.

<host> destination host name.

Optional arguments

–block-migrate True in case of block_migration. (Default=False:live_migration)

–disk-over-commit Allow overcommit.(Default=False)

nova lock command

```
usage: nova lock <server>
```

Lock a server.

Positional arguments

<server> Name or ID of server.

nova meta command

```
usage: nova meta <server> <action> <key=value> [<key=value> ...]
```

Set or Delete metadata on a server.

Positional arguments

<server> Name or ID of server

<action> Actions: 'set' or 'delete'

<key=value> Metadata to set or delete (only key is necessary on delete)

nova migrate command

```
usage: nova migrate [--poll] <server>
```

Migrate a server. The new host will be selected by the scheduler.

Positional arguments

<server> Name or ID of server.

Optional arguments

–poll Blocks while server migrates so progress can be reported.

nova migration-list command

```
usage: nova migration-list [--host <host>] [--status <status>]
                           [--cell_name <cell_name>]
```

Print a list of migrations.

Optional arguments

–host <host> Fetch migrations for the given host.

–status <status> Fetch migrations for the given status.

–cell_name <cell_name> Fetch migrations for the given cell_name.

nova net command

```
usage: nova net <network_id>
```

Show a network

Positional arguments

<network_id> ID of network

nova net-create command

```
usage: nova net-create <network_label> <cidr>
```

Create a network

Positional arguments

<network_label> Network label (ex. my_new_network)

<cidr> IP block to allocate from (ex. 172.16.0.0/24 or 2001:DB8::/64)

nova net-delete command

```
usage: nova net-delete <network_id>
```

```
```

Delete a network

Positional arguments

<network_id> ID of network

nova net-list command

```
usage: nova net-list
```

List networks

nova network-associate-host command

```
usage: nova network-associate-host <network> <host>
```

Associate host with network.

Positional arguments

<network> uuid of network

<host> Name of host

nova network-associate-project command

```
usage: nova network-associate-project <network>
```

Associate project with network.

Positional arguments

<network> uuid of network

nova network-create command

```
usage: nova network-create [--fixed-range-v4 <x.x.x.x/yy>]
                           [--fixed-range-v6 CIDR_V6] [--vlan <vlan id>]
                           [--vpn <vpn start>] [--gateway GATEWAY]
                           [--gateway-v6 GATEWAY_V6] [--bridge <bridge>]
                           [--bridge-interface <bridge interface>]
                           [--multi-host <'T'|'F'>] [--dns1 <DNS Address>]
                           [--dns2 <DNS Address>] [--uuid <network uuid>]
                           [--fixed-cidr <x.x.x.x/yy>]
                           [--project-id <project id>] [--priority <number>]
                           <network_label>
```

Create a network.

Positional arguments

<network_label> Label for network

Optional arguments

–fixed-range-v4 <x.x.x.x/yy> IPv4 subnet (ex: 10.0.0.0/8)

–fixed-range-v6 CIDR_V6 IPv6 subnet (ex: fe80::/64

–vlan <vlan id> vlan id

–vpn <vpn start> vpn start

–gateway GATEWAY gateway

–gateway-v6 GATEWAY_V6 ipv6 gateway

–bridge <bridge> VIFs on this network are connected to this bridge

–bridge-interface <bridge interface> the bridge is connected to this interface

–multi-host <'T'|'F'> Multi host

–dns1 <DNS Address> First DNS

–dns2 <DNS Address> Second DNS

–uuid <network uuid> Network UUID

–fixed-cidr <x.x.x.x/yy> IPv4 subnet for fixed IPS (ex: 10.20.0.0/16)

–project-id <project id> Project id

–priority <number> Network interface priority

nova network-disassociate command

```
usage: nova network-disassociate [--host-only [<0|1>]]
                                 [--project-only [<0|1>]]
                                 <network>
```

Disassociate host and/or project from the given network.

Positional arguments

<network> uuid of network

Optional arguments

–host-only [<0|1>]

–project-only [<0|1>]

nova network-list command

```
usage: nova network-list
```

Print a list of available networks.

nova network-show command

```
usage: nova network-show <network>
```

Show details about the given network.

Positional arguments

<network> uuid or label of network

nova pause command

```
usage: nova pause <server>
```

Pause a server.

Positional arguments

<server> Name or ID of server.

nova quota-class-show command

```
usage: nova quota-class-show <class>
```

List the quotas for a quota class.

Positional arguments

<class> Name of quota class to list the quotas for.

nova quota-class-update command

```
usage: nova quota-class-update [--instances <instances>] [--cores <cores>]
                               [--ram <ram>] [--floating-ips <floating-ips>]
                               [--metadata-items <metadata-items>]
                               [--injected-files <injected-files>]
                               [--injected-file-content-bytes <injected-file-
content-bytes>]
                               [--injected-file-path-bytes <injected-file-
path-bytes>]
                               [--key-pairs <key-pairs>]
                               [--security-groups <security-groups>]
```

```
                                [--security-group-rules <security-group-rules>]
                                <class>
```

Update the quotas for a quota class.

Positional arguments

<class> Name of quota class to set the quotas for.

Optional arguments

–instances <instances> New value for the "instances" quota.

–cores <cores> New value for the "cores" quota.

–ram <ram> New value for the "ram" quota.

–floating-ips <floating-ips> New value for the "floating-ips" quota.

–metadata-items <metadata-items> New value for the "metadata-items" quota.

–injected-files <injected-files> New value for the "injected-files" quota.

–injected-file-content-bytes <injected-file-content-bytes> New value for the "injected-file-content-bytes" quota.

–injected-file-path-bytes <injected-file-path-bytes> New value for the "injected-file-path-bytes" quota.

–key-pairs <key-pairs> New value for the "key-pairs" quota.

–security-groups <security-groups> New value for the "security-groups" quota.

–security-group-rules <security-group-rules> New value for the "security-group-rules" quota.

nova quota-defaults command

```
usage: nova quota-defaults [--tenant <tenant-id>]
```

List the default quotas for a tenant.

Optional arguments

–tenant <tenant-id> ID of tenant to list the default quotas for.

nova quota-delete command

```
usage: nova quota-delete [--tenant <tenant-id>] [--user <user-id>]
```

Delete quota for a tenant/user so their quota will Revert back to default.

Optional arguments

–tenant <tenant-id> ID of tenant to delete quota for.

–user <user-id> ID of user to delete quota for.

nova quota-show command

```
usage: nova quota-show [--tenant <tenant-id>] [--user <user-id>]
```

List the quotas for a tenant/user.

Optional arguments

–tenant <tenant-id> ID of tenant to list the quotas for.

–user <user-id> ID of user to list the quotas for.

nova quota-update command

```
usage: nova quota-update [--user <user-id>] [--instances <instances>]
                         [--cores <cores>] [--ram <ram>]
                         [--floating-ips <floating-ips>]
                         [--fixed-ips <fixed-ips>]
                         [--metadata-items <metadata-items>]
                         [--injected-files <injected-files>]
                         [--injected-file-content-bytes <injected-file-
content-bytes>]
                         [--injected-file-path-bytes <injected-file-path-
bytes>]
                         [--key-pairs <key-pairs>]
                         [--security-groups <security-groups>]
                         [--security-group-rules <security-group-rules>]
                         [--force]
                         <tenant-id>
```

Update the quotas for a tenant/user.

Positional arguments

<tenant-id> ID of tenant to set the quotas for.

Optional arguments

–user <user-id> ID of user to set the quotas for.

–instances <instances> New value for the "instances" quota.

–cores <cores> New value for the "cores" quota.

–ram <ram> New value for the "ram" quota.

–floating-ips <floating-ips>	New value for the "floating-ips" quota.
–fixed-ips <fixed-ips>	New value for the "fixed-ips" quota.
–metadata-items <metadata-items>	New value for the "metadata-items" quota.
–injected-files <injected-files>	New value for the "injected-files" quota.
–injected-file-content-bytes <injected-file-content-bytes>	New value for the "injected-file-content-bytes" quota.
–injected-file-path-bytes <injected-file-path-bytes>	New value for the "injected-file-path-bytes" quota.
–key-pairs <key-pairs>	New value for the "key-pairs" quota.
–security-groups <security-groups>	New value for the "security-groups" quota.
–security-group-rules <security-group-rules>	New value for the "security-group-rules" quota.
–force	Whether force update the quota even if the already used and reserved exceeds the new quota

nova rate-limits command

```
usage: nova rate-limits
```

Print a list of rate limits for a user

nova reboot command

```
usage: nova reboot [--hard] [--poll] <server>
```

Reboot a server.

Positional arguments

<server> Name or ID of server.

Optional arguments

–hard Perform a hard reboot (instead of a soft one).

–poll Blocks while server is rebooting.

nova rebuild command

```
usage: nova rebuild [--rebuild-password <rebuild-password>] [--poll]
```

```
                    [--minimal] [--preserve-ephemeral]
                    <server> <image>
```

Shutdown, re-image, and re-boot a server.

Positional arguments

<server> Name or ID of server.

<image> Name or ID of new image.

Optional arguments

–rebuild-password <rebuild-password>	Set the provided password on the rebuild server.
–poll	Blocks while server rebuilds so progress can be reported.
–minimal	Skips flavor/image lookups when showing servers
–preserve-ephemeral	Preserve the default ephemeral storage partition on rebuild.

nova refresh-network command

```
usage: nova refresh-network <server>
```

Refresh server network information.

Positional arguments

<server> Name or ID of a server for which the network cache should be refreshed from neutron (Admin only).

nova remove-fixed-ip command

```
usage: nova remove-fixed-ip <server> <address>
```

Remove an IP address from a server.

Positional arguments

<server> Name or ID of server.

<address> IP Address.

nova remove-secgroup command

```
usage: nova remove-secgroup <server> <secgroup>
```

```
```

Remove a Security Group from a server.

Positional arguments

<server> Name or ID of server.

<secgroup> Name of Security Group.

nova rename command

```
usage: nova rename <server> <name>
```

Rename a server.

Positional arguments

<server> Name (old name) or ID of server.

<name> New name for the server.

nova rescue command

```
usage: nova rescue <server>
```

Rescue a server.

Positional arguments

<server> Name or ID of server.

nova reset-network command

```
usage: nova reset-network <server>
```

Reset network of a server.

Positional arguments

<server> Name or ID of server.

nova reset-state command

```
usage: nova reset-state [--active] <server>
```

Reset the state of a server.

Positional arguments

<server> Name or ID of server.

Optional arguments

–active Request the server be reset to "active" state instead of "error" state (the default).

nova resize command

```
usage: nova resize [--poll] <server> <flavor>
```

Resize a server.

Positional arguments

<server> Name or ID of server.

<flavor> Name or ID of new flavor.

Optional arguments

–poll Blocks while servers resizes so progress can be reported.

nova resize-confirm command

```
usage: nova resize-confirm <server>
```

Confirm a previous resize.

Positional arguments

<server> Name or ID of server.

nova resize-revert command

```
usage: nova resize-revert <server>
```

Revert a previous resize (and return to the previous VM).

Positional arguments

<server> Name or ID of server.

nova restore command

```
usage: nova restore <server>
```

Restore a soft-deleted server.

Positional arguments

<server> Name or ID of server.

nova resume command

```
usage: nova resume <server>
```

Resume a server.

Positional arguments

<server> Name or ID of server.

nova root-password command

```
usage: nova root-password <server>
```

Change the root password for a server.

Positional arguments

<server> Name or ID of server.

nova scrub command

```
usage: nova scrub <project_id>
```

Delete data associated with the project.

Positional arguments

<project_id> The ID of the project.

nova secgroup-add-group-rule command

```
usage: nova secgroup-add-group-rule <secgroup> <source-group> <ip-proto>
                                    <from-port> <to-port>
```

Add a source group rule to a security group.

Positional arguments

<secgroup> ID or name of security group.

<source-group> ID or name of source group.

<ip-proto> IP protocol (icmp, tcp, udp).

<from-port> Port at start of range.

<to-port> Port at end of range.

nova secgroup-add-rule command

```
usage: nova secgroup-add-rule <secgroup> <ip-proto> <from-port> <to-port>
                              <cidr>
```

Add a rule to a security group.

Positional arguments

<secgroup> ID or name of security group.

<ip-proto> IP protocol (icmp, tcp, udp).

<from-port> Port at start of range.

<to-port> Port at end of range.

<cidr> CIDR for address range.

nova secgroup-create command

```
usage: nova secgroup-create <name> <description>
```

Create a security group.

Positional arguments

<name> Name of security group.

<description> Description of security group.

nova secgroup-delete command

```
usage: nova secgroup-delete <secgroup>
```

Delete a security group.

Positional arguments

<secgroup> ID or name of security group.

nova secgroup-delete-group-rule command

```
usage: nova secgroup-delete-group-rule <secgroup> <source-group> <ip-proto>
```

```
                                        <from-port> <to-port>
```

Delete a source group rule from a security group.

Positional arguments

<secgroup> ID or name of security group.

<source-group> ID or name of source group.

<ip-proto> IP protocol (icmp, tcp, udp).

<from-port> Port at start of range.

<to-port> Port at end of range.

nova secgroup-delete-rule command

```
usage: nova secgroup-delete-rule <secgroup> <ip-proto> <from-port> <to-port>
                                 <cidr>
```

Delete a rule from a security group.

Positional arguments

<secgroup> ID or name of security group.

<ip-proto> IP protocol (icmp, tcp, udp).

<from-port> Port at start of range.

<to-port> Port at end of range.

<cidr> CIDR for address range.

nova secgroup-list command

```
usage: nova secgroup-list [--all-tenants [<0|1>]]
```

List security groups for the current tenant.

Optional arguments

–all-tenants [<0|1>] Display information from all tenants (Admin only).

nova secgroup-list-rules command

```
usage: nova secgroup-list-rules <secgroup>
```

List rules for a security group.

Positional arguments

<secgroup> ID or name of security group.

nova secgroup-update command

```
usage: nova secgroup-update <secgroup> <name> <description>
```

Update a security group.

Positional arguments

<secgroup> ID or name of security group.

<name> Name of security group.

<description> Description of security group.

nova service-disable command

```
usage: nova service-disable [--reason <reason>] <hostname> <binary>
```

Disable the service.

Positional arguments

<hostname> Name of host.

<binary> Service binary.

Optional arguments

–reason <reason> Reason for disabling service.

nova service-enable command

```
usage: nova service-enable <hostname> <binary>
```

Enable the service.

Positional arguments

<hostname> Name of host.

<binary> Service binary.

nova service-list command

```
usage: nova service-list [--host <hostname>] [--binary <binary>]
```

```
```

Show a list of all running services. Filter by host & binary.

Optional arguments

–host <hostname> Name of host.

–binary <binary> Service binary.

nova shelve command

```
usage: nova shelve <server>
```

Shelve a server.

Positional arguments

<server> Name or ID of server.

nova shelve-offload command

```
usage: nova shelve-offload <server>
```

Remove a shelved server from the compute node.

Positional arguments

<server> Name or ID of server.

nova show command

```
usage: nova show [--minimal] <server>
```

Show details about the given server.

Positional arguments

<server> Name or ID of server.

Optional arguments

–minimal Skips flavor/image lookups when showing servers

nova ssh command

```
usage: nova ssh [--port PORT] [--private] [--ipv6] [--login <login>]
                [-i IDENTITY] [--extra-opts EXTRA]
                <server>
```

```
```

SSH into a server.

Positional arguments

<server> Name or ID of server.

Optional arguments

–port PORT	Optional flag to indicate which port to use for ssh. (Default=22)
–private	Optional flag to indicate whether to only use private address attached to an instance. (Default=False). If no public address is found try private address
–ipv6	Optional flag to indicate whether to use an IPv6 address attached to a server. (Defaults to IPv4 address)
–login <login>	Login to use.
-i IDENTITY, –identity IDENTITY	Private key file, same as the -i option to the ssh command.
–extra-opts EXTRA	Extra options to pass to ssh. see: man ssh

nova start command

```
usage: nova start <server>
```

Start a server.

Positional arguments

<server> Name or ID of server.

nova stop command

```
usage: nova stop <server>
```

Stop a server.

Positional arguments

<server> Name or ID of server.

nova suspend command

```
usage: nova suspend <server>
```

```
```

Suspend a server.

Positional arguments

<server> Name or ID of server.

nova unlock command

```
usage: nova unlock <server>
```

Unlock a server.

Positional arguments

<server> Name or ID of server.

nova unpause command

```
usage: nova unpause <server>
```

Unpause a server.

Positional arguments

<server> Name or ID of server.

nova unrescue command

```
usage: nova unrescue <server>
```

Unrescue a server.

Positional arguments

<server> Name or ID of server.

nova unshelve command

```
usage: nova unshelve <server>
```

Unshelve a server.

Positional arguments

<server> Name or ID of server.

nova usage command

```
usage: nova usage [--start <start>] [--end <end>] [--tenant <tenant-id>]
```

Show usage data for a single tenant.

Optional arguments

–start <start> Usage range start date ex 2012-01-20 (default: 4 weeks ago)

–end <end> Usage range end date, ex 2012-01-20 (default: tomorrow)

–tenant <tenant-id> UUID or name of tenant to get usage for.

nova usage-list command

```
usage: nova usage-list [--start <start>] [--end <end>]
```

List usage data for all tenants.

Optional arguments

–start <start> Usage range start date ex 2012-01-20 (default: 4 weeks ago)

–end <end> Usage range end date, ex 2012-01-20 (default: tomorrow)

nova volume-attach command

```
usage: nova volume-attach <server> <volume> [<device>]
```

Attach a volume to a server.

Positional arguments

<server> Name or ID of server.

<volume> ID of the volume to attach.

<device> Name of the device e.g. /dev/vdb. Use "auto" for autoassign (if supported)

nova volume-create command

```
usage: nova volume-create [--snapshot-id <snapshot-id>]
                          [--image-id <image-id>]
                          [--display-name <display-name>]
                          [--display-description <display-description>]
                          [--volume-type <volume-type>]
                          [--availability-zone <availability-zone>]
```

```
                              <size>
```

Add a new volume.

Positional arguments

<size> Size of volume in GB

Optional arguments

–snapshot-id <snapshot-id>	Optional snapshot id to create the volume from. (Default=None)
–image-id <image-id>	Optional image id to create the volume from. (Default=None)
–display-name <display-name>	Optional volume name. (Default=None)
–display-description <display-description>	Optional volume description. (Default=None)
–volume-type <volume-type>	Optional volume type. (Default=None)
–availability-zone <availability-zone>	Optional Availability Zone for volume. (Default=None)

nova volume-delete command

```
usage: nova volume-delete <volume> [<volume> ...]
```

Remove volume(s).

Positional arguments

<volume> Name or ID of the volume(s) to delete.

nova volume-detach command

```
usage: nova volume-detach <server> <volume>
```

Detach a volume from a server.

Positional arguments

<server> Name or ID of server.

<volume> Attachment ID of the volume.

nova volume-list command

```
usage: nova volume-list [--all-tenants [<0|1>]]
```

```
```

List all the volumes.

Optional arguments

–all-tenants [<0|1>] Display information from all tenants (Admin only).

nova volume-show command

```
usage: nova volume-show <volume>
```

Show details about a volume.

Positional arguments

<volume> Name or ID of the volume.

nova volume-snapshot-create command

```
usage: nova volume-snapshot-create [--force <True|False>]
                                   [--display-name <display-name>]
                                   [--display-description <display-
description>]
                                   <volume-id>
```

Add a new snapshot.

Positional arguments

<volume-id> ID of the volume to snapshot

Optional arguments

–force <True|False> Optional flag to indicate whether to snapshot a volume even if its attached to a server. (Default=False)

–display-name <display-name> Optional snapshot name. (Default=None)

–display-description <display-description> Optional snapshot description. (Default=None)

nova volume-snapshot-delete command

```
usage: nova volume-snapshot-delete <snapshot>
```

Remove a snapshot.

Positional arguments

<snapshot> Name or ID of the snapshot to delete.

nova volume-snapshot-list command

```
usage: nova volume-snapshot-list
```

List all the snapshots.

nova volume-snapshot-show command

```
usage: nova volume-snapshot-show <snapshot>
```

Show details about a snapshot.

Positional arguments

<snapshot> Name or ID of the snapshot.

nova volume-type-create command

```
usage: nova volume-type-create <name>
```

Create a new volume type.

Positional arguments

<name> Name of the new flavor

nova volume-type-delete command

```
usage: nova volume-type-delete <id>
```

Delete a specific flavor

Positional arguments

<id> Unique ID of the volume type to delete

nova volume-type-list command

```
usage: nova volume-type-list
```

Print a list of available 'volume types'.

nova volume-update command

```
usage: nova volume-update <server> <volume> <volume>
```

Update volume attachment.

Positional arguments

<server> Name or ID of server.

<volume> Attachment ID of the volume.

<volume> ID of the volume to attach.

nova x509-create-cert command

```
usage: nova x509-create-cert [<private-key-filename>] [<x509-cert-filename>]
```

Create x509 cert for a user in tenant.

Positional arguments

<private-key-filename> Filename for the private key [Default: pk.pem]

<x509-cert-filename> Filename for the X.509 certificate [Default: cert.pem]

nova x509-get-root-cert command

```
usage: nova x509-get-root-cert [<filename>]
```

Fetch the x509 root cert.

Positional arguments

<filename> Filename to write the x509 root cert.

4. Identity service command-line client

The **keystone** client is the command-line interface (CLI) for the OpenStack Identity API and its extensions. This chapter documents **keystone** version 0.7.1.

For help on a specific **keystone** command, enter:

```
$ keystone help COMMAND
```

keystone usage

```
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> ...
```

Subcommands

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.

keystone 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>	Specify the region to use. Defaults to `env[OS_REGION_NAME]`.
–os-identity-api-version <identity-api-version>	Specify Identity API version to use. 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.

keystone bootstrap command

```
usage: keystone bootstrap [--user-name <user-name>] --pass <password>
                          [--role-name <role-name>]
                          [--tenant-name <tenant-name>]
```

Grants a new role to a new user on a new tenant, after creating each.

Arguments

–user-name <user-name>	The name of the user to be created (default="admin").
–pass <password>	The password for the new user.

–role-name <role-name> The name of the role to be created and granted to the user (default="admin").

–tenant-name <tenant-name> The name of the tenant to be created (default="admin").

keystone catalog command

```
usage: keystone catalog [--service <service-type>]
```

List service catalog, possibly filtered by service.

Arguments

–service <service-type> Service type to return.

keystone discover command

```
usage: keystone discover
```

Discover Keystone servers, supported API versions and extensions.

keystone ec2-credentials-create command

```
usage: keystone ec2-credentials-create [--user-id <user-id>]
                                       [--tenant-id <tenant-id>]
```

Create EC2-compatible credentials for user per tenant.

Arguments

–user-id <user-id> User ID for which to create credentials. If not specified, the authenticated user will be used.

–tenant-id <tenant-id> Tenant ID for which to to create credentials. If not specified, the authenticated tenant ID will be used.

keystone ec2-credentials-delete command

```
usage: keystone ec2-credentials-delete [--user-id <user-id>] --access
                                       <access-key>
```

Delete EC2-compatible credentials.

Arguments

–user-id <user-id> User ID.

–access <access-key> Access Key.

keystone ec2-credentials-get command

```
usage: keystone ec2-credentials-get [--user-id <user-id>] --access
                                    <access-key>
```

Display EC2-compatible credentials.

Arguments

–user-id <user-id> User ID.

–access <access-key> Access Key.

keystone ec2-credentials-list command

```
usage: keystone ec2-credentials-list [--user-id <user-id>]
```

List EC2-compatible credentials for a user.

Arguments

–user-id <user-id> User ID.

keystone endpoint-create command

```
usage: keystone endpoint-create [--region <endpoint-region>] --service
                                <service> --publicurl <public-url>
                                [--adminurl <admin-url>]
                                [--internalurl <internal-url>]
```

Create a new endpoint associated with a service.

Arguments

–region <endpoint-region> Endpoint region.

–service <service>, –service-id <service>, –service_id <service> Name or ID of service associated with endpoint.

–publicurl <public-url> Public URL endpoint.

–adminurl <admin-url> Admin URL endpoint.

–internalurl <internal-url> Internal URL endpoint.

keystone endpoint-delete command

```
usage: keystone endpoint-delete <endpoint-id>
```

Delete a service endpoint.

Arguments

<endpoint-id> ID of endpoint to delete.

keystone endpoint-get command

```
usage: keystone endpoint-get --service <service-type>
                             [--endpoint-type <endpoint-type>]
                             [--attr <service-attribute>] [--value <value>]
```

Find endpoint filtered by a specific attribute or service type.

Arguments

–service <service-type>	Service type to select.
–endpoint-type <endpoint-type>	Endpoint type to select.
–attr <service-attribute>	Service attribute to match for selection.
–value <value>	Value of attribute to match.

keystone endpoint-list command

```
usage: keystone endpoint-list
```

List configured service endpoints.

keystone password-update command

```
usage: keystone password-update [--current-password <current-password>]
                                [--new-password  <new-password>]
```

Update own password.

Arguments

–current-password <current-password>	Current password, Defaults to the password as set by –os-password or `env[OS_PASSWORD]`.
–new-password <new-password>	Desired new password.

keystone role-create command

```
usage: keystone role-create --name <role-name>
```

```
```

Create new role.

Arguments

–name <role-name> Name of new role.

keystone role-delete command

```
usage: keystone role-delete <role>
```

Delete role.

Arguments

<role> Name or ID of role to delete.

keystone role-get command

```
usage: keystone role-get <role>
```

Display role details.

Arguments

<role> Name or ID of role to display.

keystone role-list command

```
usage: keystone role-list
```

List all roles.

keystone service-create command

```
usage: keystone service-create --name <name> --type <type>
                               [--description <service-description>]
```

Add service to Service Catalog.

Arguments

–name <name> Name of new service (must be unique).

–type <type> Service type (one of: identity, compute, network, image, object-store, or other service identifier string).

–description <service-description> Description of service.

keystone service-delete command

```
usage: keystone service-delete <service>
```

Delete service from Service Catalog.

Arguments

<service> Name or ID of service to delete.

keystone service-get command

```
usage: keystone service-get <service>
```

Display service from Service Catalog.

Arguments

<service> Name or ID of service to display.

keystone service-list command

```
usage: keystone service-list
```

List all services in Service Catalog.

keystone tenant-create command

```
usage: keystone tenant-create --name <tenant-name>
                              [--description <tenant-description>]
                              [--enabled <true|false>]
```

Create new tenant.

Arguments

–name <tenant-name> New tenant name (must be unique).

–description <tenant-description> Description of new tenant. Default is none.

–enabled <true|false> Initial tenant enabled status. Default is true.

keystone tenant-delete command

```
usage: keystone tenant-delete <tenant>
```

```
```

Delete tenant.

Arguments

<tenant> Name or ID of tenant to delete.

keystone tenant-get command

```
usage: keystone tenant-get <tenant>
```

Display tenant details.

Arguments

<tenant> Name or ID of tenant to display.

keystone tenant-list command

```
usage: keystone tenant-list
```

List all tenants.

keystone tenant-update command

```
usage: keystone tenant-update [--name <tenant_name>]
                              [--description <tenant-description>]
                              [--enabled <true|false>]
                              <tenant>
```

Update tenant name, description, enabled status.

Arguments

–name <tenant_name>	Desired new name of tenant.
–description <tenant-description>	Desired new description of tenant.
–enabled <true|false>	Enable or disable tenant.
<tenant>	Name or ID of tenant to update.

keystone token-get command

```
usage: keystone token-get [--wrap <integer>]
```

Display the current user token.

Arguments

–wrap <integer> Wrap PKI tokens to a specified length, or 0 to disable.

keystone user-create command

```
usage: keystone user-create --name <user-name> [--tenant <tenant>]
                            [--pass [<pass>]] [--email <email>]
                            [--enabled <true|false>]
```

Create new user

Arguments

–name <user-name> New user name (must be unique).

–tenant <tenant>, –tenant-id <tenant> New user default tenant.

–pass [<pass>] New user password; required for some auth backends.

–email <email> New user email address.

–enabled <true|false> Initial user enabled status. Default is true.

keystone user-delete command

```
usage: keystone user-delete <user>
```

Delete user.

Arguments

<user> Name or ID of user to delete.

keystone user-get command

```
usage: keystone user-get <user>
```

Display user details.

Arguments

<user> Name or ID of user to display.

keystone user-list command

```
usage: keystone user-list [--tenant <tenant>]
```

List users.

Arguments

–tenant <tenant>, –tenant-id <tenant> Tenant; lists all users if not specified.

keystone user-password-update command

```
usage: keystone user-password-update [--pass <password>] <user>
```

Update user password.

Arguments

–pass <password> Desired new password.

<user> Name or ID of user to update password.

keystone user-role-add command

```
usage: keystone user-role-add --user <user> --role <role> [--tenant <tenant>]
```

Add role to user.

Arguments

–user <user>, –user-id <user>, –user_id <user> Name or ID of user.

–role <role>, –role-id <role>, –role_id <role> Name or ID of role.

–tenant <tenant>, –tenant-id <tenant> Name or ID of tenant.

keystone user-role-list command

```
usage: keystone user-role-list [--user <user>] [--tenant <tenant>]
```

List roles granted to a user.

Arguments

–user <user>, –user-id <user> List roles granted to specified user.

–tenant <tenant>, –tenant-id <tenant> List only roles granted on specified tenant.

keystone user-role-remove command

```
usage: keystone user-role-remove --user <user> --role <role>
                                 [--tenant <tenant>]
```

Remove role from user.

Arguments

–user <user>, –user-id <user>, –user_id <user> Name or ID of user.

–role <role>, –role-id <role>, –role_id <role> Name or ID of role.

–tenant <tenant>, –tenant-id <tenant> Name or ID of tenant.

keystone user-update command

```
usage: keystone user-update [--name <user-name>] [--email <email>]
                            [--enabled <true|false>]
                            <user>
```

Update user's name, email, and enabled status.

Arguments

–name <user-name> Desired new user name.

–email <email> Desired new email address.

–enabled <true|false> Enable or disable user.

<user> Name or ID of user to update.

5. Image Service command-line client

The **glance** client is the command-line interface (CLI) for the OpenStack Image Service API and its extensions. This chapter documents **glance** version 0.12.0.

For help on a specific **glance** command, enter:

```
$ glance help COMMAND
```

glance usage

```
usage: glance [--version] [-d] [-v] [--get-schema] [-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> ...
```

Subcommands

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.

glance optional arguments

–version	show program's version number and exit
-d, –debug	Defaults to `env[GLANCECLIENT_DEBUG]`
-v, –verbose	Print more verbose output
–get-schema	Force retrieving the schema used to generate portions of the help text rather than using a cached copy. Ignored with api version 1
-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.

glance image-create command

```
usage: glance image-create [--id <IMAGE_ID>] [--name <NAME>] [--store <STORE>]
                           [--disk-format <DISK_FORMAT>]
                           [--container-format <CONTAINER_FORMAT>]
                           [--owner <TENANT_ID>] [--size <SIZE>]
                           [--min-disk <DISK_GB>] [--min-ram <DISK_RAM>]
                           [--location <IMAGE_URL>] [--file <FILE>]
                           [--checksum <CHECKSUM>] [--copy-from <IMAGE_URL>]
                           [--is-public {True,False}]
                           [--is-protected {True,False}]
                           [--property <key=value>] [--human-readable]
                           [--progress]
```

Create a new image.

Optional arguments

–id <IMAGE_ID> ID	of image to reserve.
–name <NAME>	Name of image.
–store <STORE>	Store to upload image to.
–disk-format <DISK_FORMAT>	Disk format of image. Acceptable formats: ami, ari, aki, vhd, vmdk, raw, qcow2, vdi, and iso.
–container-format <CONTAINER_FORMAT>	Container format of image. Acceptable formats: ami, ari, aki, bare, and ovf.
–owner <TENANT_ID>	Tenant who should own image.
–size <SIZE>	Size of image data (in bytes). Only used with '– location' and '–copy_from'.
–min-disk <DISK_GB>	Minimum size of disk needed to boot image (in gigabytes).
–min-ram <DISK_RAM>	Minimum amount of ram needed to boot image (in megabytes).
–location <IMAGE_URL>	URL where the data for this image already 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 creation. Alternatively, images can be passed to the client via stdin.
–checksum <CHECKSUM>	Hash of image data used Glance can use for verification. Provide a md5 checksum here.
–copy-from <IMAGE_URL>	Similar to '–location' in usage, but this indicates that the Glance server should immediately copy the data and store it in its configured image store.
–is-public {True,False}	Make image accessible to the public.
–is-protected {True,False}	Prevent image from being deleted.
–property <key=value>	Arbitrary property to associate with image. May be used multiple times.
–human-readable	Print image size in a human-friendly format.
–progress	Show upload progress bar.

glance image-delete command

```
usage: glance image-delete <IMAGE> [<IMAGE> ...]
```

Delete specified image(s).

Positional arguments

<IMAGE> Name or ID of image(s) to delete.

glance image-list command

```
usage: glance image-list [--name <NAME>] [--status <STATUS>]
                         [--container-format <CONTAINER_FORMAT>]
                         [--disk-format <DISK_FORMAT>] [--size-min <SIZE>]
                         [--size-max <SIZE>] [--property-filter <KEY=VALUE>]
                         [--page-size <SIZE>] [--human-readable]
                         [--sort-key {name,status,container_format,
disk_format,size,id,created_at,updated_at}]
                         [--sort-dir {asc,desc}] [--is-public {True,False}]
                         [--owner <TENANT_ID>] [--all-tenants]
```

List images you can access.

Optional arguments

–name <NAME>	Filter images to those that have this name.

–status <STATUS>	Filter images to those that have this status.
–container-format <CONTAINER_FORMAT>	Filter images to those that have this container format. Acceptable formats: ami, ari, aki, bare, and ovf.
–disk-format <DISK_FORMAT>	Filter images to those that have this disk format. Acceptable formats: ami, ari, aki, vhd, vmdk, raw, qcow2, vdi, and iso.
–size-min <SIZE>	Filter images to those with a size greater than this.
–size-max <SIZE>	Filter images to those with a size less than this.
–property-filter <KEY=VALUE>	Filter images by a user-defined image property.
–page-size <SIZE>	Number of images to request in each paginated request.
–human-readable	Print image size in a human-friendly format.
–sort-key {name,status,container_format,disk_format,size,id,created_at,updated_at}	Sort image list by specified field.
–sort-dir {asc,desc}	Sort image list in specified direction.
–is-public {True,False}	Allows the user to select a listing of public or non public images.
–owner <TENANT_ID>	Display only images owned by this tenant id. Filtering occurs on the client side so may be inefficient. This option is mainly intended for admin use. Use an empty string ('') to list images with no owner. Note: This option overrides the –is-public argument if present. Note: the v2 API supports more efficient server-side owner based filtering.
–all-tenants	Allows the admin user to list all images irrespective of the image's owner or is_public value.

glance image-show command

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

Describe a specific image.

Positional arguments

<IMAGE> Name or ID of image to describe.

Optional arguments

–human-readable Print image size in a human-friendly format.

glance image-update command

```
usage: glance image-update [--name <NAME>] [--disk-format <DISK_FORMAT>]
                           [--container-format <CONTAINER_FORMAT>]
                           [--owner <TENANT_ID>] [--size <SIZE>]
                           [--min-disk <DISK_GB>] [--min-ram <DISK_RAM>]
                           [--location <IMAGE_URL>] [--file <FILE>]
                           [--checksum <CHECKSUM>] [--copy-from <IMAGE_URL>]
                           [--is-public {True,False}]
                           [--is-protected {True,False}]
                           [--property <key=value>] [--purge-props]
                           [--human-readable] [--progress]
                           <IMAGE>
```

Update a specific image.

Positional arguments

<IMAGE> Name or ID of image to modify.

Optional arguments

–name <NAME>	Name of image.
–disk-format <DISK_FORMAT>	Disk format of image. Acceptable formats: ami, ari, aki, vhd, vmdk, raw, qcow2, vdi, and iso.
–container-format <CONTAINER_FORMAT>	Container format of image. Acceptable formats: ami, ari, aki, bare, and ovf.
–owner <TENANT_ID>	Tenant who should own image.
–size <SIZE>	Size of image data (in bytes).
–min-disk <DISK_GB>	Minimum size of disk needed to boot image (in gigabytes).
–min-ram <DISK_RAM>	Minimum amount of ram needed to boot image (in megabytes).
–location <IMAGE_URL>	URL where the data for this image already 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, images can be passed to the client via stdin.
–checksum <CHECKSUM>	Hash of image data used Glance can use for verification.
–copy-from <IMAGE_URL>	Similar to '–location' in usage, but this indicates that the Glance server should immediately copy the data and store it in its configured image store.

–is-public {True,False}	Make image accessible to the public.
–is-protected {True,False}	Prevent image from being deleted.
–property <key=value>	Arbitrary property to associate with image. May be used multiple times.
–purge-props	If this flag is present, delete all image properties not explicitly set in the update request. Otherwise, those properties not referenced are preserved.
–human-readable	Print image size in a human-friendly format.
–progress	Show upload progress bar.

glance member-create command

```
usage: glance member-create [--can-share] <IMAGE> <TENANT_ID>
```

Share a specific image with a tenant.

Positional arguments

<IMAGE>	Image to add member to.
<TENANT_ID>	Tenant to add as member

Optional arguments

–can-share	Allow the specified tenant to share this image.

glance member-delete command

```
usage: glance member-delete <IMAGE> <TENANT_ID>
```

Remove a shared image from a tenant.

Positional arguments

<IMAGE>	Image from which to remove member
<TENANT_ID>	Tenant to remove as member

glance member-list command

```
usage: glance member-list [--image-id <IMAGE_ID>] [--tenant-id <TENANT_ID>]
```

Describe sharing permissions by image or tenant.

Optional arguments

–image-id <IMAGE_ID>	Filter results by an image ID.
–tenant-id <TENANT_ID>	Filter results by a tenant ID.

6. Networking command-line client

The **neutron** client is the command-line interface (CLI) for the OpenStack Networking API and its extensions. This chapter documents **neutron** version 2.3.4.

For help on a specific **neutron** command, enter:

```
$ neutron help COMMAND
```

neutron usage

```
usage: neutron [--version] [-v] [-q] [-h] [--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]
```

neutron optional arguments

–version	show program's version number and exit
-v, –verbose, –debug	Increase verbosity of output and show tracebacks on errors. Can be repeated.
-q, –quiet	Suppress output except warnings and errors
-h, –help	Show this help message and exit
–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]`
–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 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.

neutron API v2.0 commands

agent-delete	Delete a given agent.
agent-list	List agents.
agent-show	Show information of a given agent.
agent-update	Update a given agent.
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.
complete	print bash completion command
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.
meter-label-create	Create a metering label for a given tenant.
meter-label-delete	Delete a given metering label.
meter-label-list	List metering labels that belong to a given tenant.
meter-label-rule-create	Create a metering label rule for a given label.
meter-label-rule-delete	Delete a given metering label.
meter-label-rule-list	List metering labels that belong to a given label.
meter-label-rule-show	Show information of a given metering label rule.
meter-label-show	Show information of a given metering label.
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 quotas of all tenants who have non-default quota values.
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 subnets 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.

neutron agent-delete command

```
usage: neutron agent-delete [-h] [--request-format {json,xml}] AGENT
```

Delete a given agent.

Positional arguments

AGENT ID of agent to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron agent-list command

```
usage: neutron agent-list [-h] [-f {csv,table}] [-c COLUMN]
                          [--quote {all,minimal,none,nonnumeric}]
                          [--request-format {json,xml}] [-D] [-F FIELD]
```

List agents.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron agent-show command

```
usage: neutron agent-show [-h] [-f {shell,table}] [-c COLUMN]
                          [--variable VARIABLE] [--prefix PREFIX]
                          [--request-format {json,xml}] [-D] [-F FIELD]
                          AGENT
```

Show information of a given agent.

Positional arguments

AGENT ID of agent to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron agent-update command

```
usage: neutron agent-update [-h] [--request-format {json,xml}] AGENT
```

Update a given agent.

Positional arguments

AGENT ID or name of agent to update

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron cisco-credential-create command

```
usage: neutron cisco-credential-create [-h] [-f {shell,table}] [-c COLUMN]
                                       [--variable VARIABLE] [--prefix PREFIX]
                                       [--request-format {json,xml}]
                                       [--tenant-id TENANT_ID]
                                       [--username USERNAME]
                                       [--password PASSWORD]
                                       credential_name credential_type
```

Creates a credential.

Positional arguments

credential_name Name/Ip address for Credential

credential_type Type of the Credential

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–tenant-id TENANT_ID The owner tenant ID

–username USERNAME Username for the credential

–password PASSWORD Password for the credential

neutron cisco-credential-delete command

```
usage: neutron cisco-credential-delete [-h] [--request-format {json,xml}]
```

```
                                        CREDENTIAL
```

Delete a given credential.

Positional arguments

CREDENTIAL ID of credential to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron cisco-credential-list command

```
usage: neutron cisco-credential-list [-h] [-f {csv,table}] [-c COLUMN]
                                     [--quote {all,minimal,none,nonnumeric}]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD]
```

List credentials that belong to a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron cisco-credential-show command

```
usage: neutron cisco-credential-show [-h] [-f {shell,table}] [-c COLUMN]
                                     [--variable VARIABLE] [--prefix PREFIX]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD]
                                     CREDENTIAL
```

Show information of a given credential.

Positional arguments

CREDENTIAL ID of credential to look up

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron cisco-network-profile-create command

```
usage: neutron cisco-network-profile-create [-h] [-f {shell,table}]
                                            [-c COLUMN] [--variable VARIABLE]
                                            [--prefix PREFIX]
                                            [--request-format {json,xml}]
                                            [--tenant-id TENANT_ID]
                                            [--sub_type SUB_TYPE]
                                            [--segment_range SEGMENT_RANGE]
                                            [--physical_network
 PHYSICAL_NETWORK]
                                            [--multicast_ip_range
 MULTICAST_IP_RANGE]
                                            [--add-tenant ADD_TENANT]
                                            name
                                            {vlan,overlay,multi-segment,trunk}
```

Creates a network profile.

Positional arguments

name	Name for Network Profile
{vlan,overlay,multi-segment,trunk}	Segment type

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–sub_type SUB_TYPE	Sub-type for the segment. Available sub-types for overlay segments: native, enhanced; For trunk segments: vlan, overlay.
–segment_range SEGMENT_RANGE	Range for the Segment
–physical_network PHYSICAL_NETWORK	Name for the Physical Network
–multicast_ip_range MULTICAST_IP_RANGE	Multicast IPv4 Range

–add-tenant ADD_TENANT Add tenant to the network profile

neutron cisco-network-profile-delete command

```
usage: neutron cisco-network-profile-delete [-h] [--request-format {json,xml}]
                                            NETWORK_PROFILE
```

Delete a given network profile.

Positional arguments

NETWORK_PROFILE ID or name of network_profile to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron cisco-network-profile-list command

```
usage: neutron cisco-network-profile-list [-h] [-f {csv,table}] [-c COLUMN]
                                          [--quote {all,minimal,none,
nonnumeric}]
                                          [--request-format {json,xml}] [-D]
                                          [-F FIELD]
```

List network profiles that belong to a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron cisco-network-profile-show command

```
usage: neutron cisco-network-profile-show [-h] [-f {shell,table}] [-c COLUMN]
                                          [--variable VARIABLE]
                                          [--prefix PREFIX]
                                          [--request-format {json,xml}] [-D]
                                          [-F FIELD]
                                          NETWORK_PROFILE
```

Show information of a given network profile.

Positional arguments

NETWORK_PROFILE ID or name of network_profile to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron cisco-network-profile-update command

```
usage: neutron cisco-network-profile-update [-h] [--request-format {json,xml}]
                                            NETWORK_PROFILE
```

Update network profile's information.

Positional arguments

NETWORK_PROFILE ID or name of network_profile to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron cisco-policy-profile-list command

```
usage: neutron cisco-policy-profile-list [-h] [-f {csv,table}] [-c COLUMN]
                                         [--quote {all,minimal,none,
nonnumeric}]
                                         [--request-format {json,xml}] [-D]
                                         [-F FIELD]
```

List policy profiles that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron cisco-policy-profile-show command

```
usage: neutron cisco-policy-profile-show [-h] [-f {shell,table}] [-c COLUMN]
                                         [--variable VARIABLE]
                                         [--prefix PREFIX]
                                         [--request-format {json,xml}] [-D]
                                         [-F FIELD]
                                         POLICY_PROFILE
```

Show information of a given policy profile.

Positional arguments

POLICY_PROFILE ID or name of policy_profile to look up

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron cisco-policy-profile-update command

```
usage: neutron cisco-policy-profile-update [-h] [--request-format {json,xml}]
                                           POLICY_PROFILE
```

Update policy profile's information.

Positional arguments

POLICY_PROFILE ID or name of policy_profile to update

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron dhcp-agent-list-hosting-net command

```
usage: neutron dhcp-agent-list-hosting-net [-h] [-f {csv,table}] [-c COLUMN]
                                           [--quote {all,minimal,none,
nonnumeric}]
                                           [--request-format {json,xml}] [-D]
                                           [-F FIELD]
                                           network
```

List DHCP agents hosting a network.

Positional arguments

network Network to query

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron dhcp-agent-network-add command

```
usage: neutron dhcp-agent-network-add [-h] [--request-format {json,xml}]
                                      dhcp_agent network
```

Add a network to a DHCP agent.

Positional arguments

dhcp_agent ID of the DHCP agent

network Network to add

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron dhcp-agent-network-remove command

```
usage: neutron dhcp-agent-network-remove [-h] [--request-format {json,xml}]
                                         dhcp_agent network
```

Remove a network from a DHCP agent.

Positional arguments

dhcp_agent ID of the DHCP agent

network Network to remove

Optional arguments

-h, –help	show this help message and exit

–request-format {json,xml} The xml or json request format

neutron ext-list command

```
usage: neutron ext-list [-h] [-f {csv,table}] [-c COLUMN]
                        [--quote {all,minimal,none,nonnumeric}]
                        [--request-format {json,xml}] [-D] [-F FIELD]
```

List all extensions.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron ext-show command

```
usage: neutron ext-show [-h] [-f {shell,table}] [-c COLUMN]
                        [--variable VARIABLE] [--prefix PREFIX]
                        [--request-format {json,xml}] [-D] [-F FIELD]
                        EXT-ALIAS
```

Show information of a given resource.

Positional arguments

EXT-ALIAS The extension alias

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron firewall-create command

```
usage: neutron firewall-create [-h] [-f {shell,table}] [-c COLUMN]
                               [--variable VARIABLE] [--prefix PREFIX]
                               [--request-format {json,xml}]
                               [--tenant-id TENANT_ID] [--name NAME]
```

```
                                [--description DESCRIPTION] [--shared]
                                [--admin-state-down]
                                POLICY
```

Create a firewall.

Positional arguments

POLICY Firewall policy id

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–name NAME	Name for the firewall
–description DESCRIPTION	Description for the firewall rule
–shared	Set shared to True (default False)
–admin-state-down	Set admin state up to false

neutron firewall-delete command

```
usage: neutron firewall-delete [-h] [--request-format {json,xml}] FIREWALL
```

Delete a given firewall.

Positional arguments

FIREWALL ID or name of firewall to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron firewall-list command

```
usage: neutron firewall-list [-h] [-f {csv,table}] [-c COLUMN]
                             [--quote {all,minimal,none,nonnumeric}]
                             [--request-format {json,xml}] [-D] [-F FIELD]
                             [-P SIZE] [--sort-key FIELD]
                             [--sort-dir {asc,desc}]
```

List firewalls that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron firewall-policy-create command

```
usage: neutron firewall-policy-create [-h] [-f {shell,table}] [-c COLUMN]
                                      [--variable VARIABLE] [--prefix PREFIX]
                                      [--request-format {json,xml}]
                                      [--tenant-id TENANT_ID]
                                      [--description DESCRIPTION] [--shared]
                                      [--firewall-rules FIREWALL_RULES]
                                      [--audited]
                                      NAME
```

Create a firewall policy.

Positional arguments

NAME Name for the firewall policy

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–description DESCRIPTION	Description for the firewall policy
–shared	To create a shared policy
–firewall-rules FIREWALL_RULES	Ordered list of whitespace-delimited firewall rule names or IDs; e.g., –firewall-rules "rule1 rule2"

–audited To set audited to True

neutron firewall-policy-delete command

```
usage: neutron firewall-policy-delete [-h] [--request-format {json,xml}]
                                      FIREWALL_POLICY
```

Delete a given firewall policy.

Positional arguments

FIREWALL_POLICY ID or name of firewall_policy to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron firewall-policy-insert-rule command

```
usage: neutron firewall-policy-insert-rule [-h] [--request-format {json,xml}]
                                           [--insert-before FIREWALL_RULE]
                                           [--insert-after FIREWALL_RULE]
                                           FIREWALL_POLICY FIREWALL_RULE
```

Insert a rule into a given firewall policy.

Positional arguments

FIREWALL_POLICY ID or name of firewall_policy to update

FIREWALL_RULE New rule to insert

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–insert-before FIREWALL_RULE Insert before this rule

–insert-after FIREWALL_RULE Insert after this rule

neutron firewall-policy-list command

```
usage: neutron firewall-policy-list [-h] [-f {csv,table}] [-c COLUMN]
                                    [--quote {all,minimal,none,nonnumeric}]
                                    [--request-format {json,xml}] [-D]
                                    [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                    [--sort-dir {asc,desc}]
```

List firewall policies that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron firewall-policy-remove-rule command

```
usage: neutron firewall-policy-remove-rule [-h] [--request-format {json,xml}]
                                           FIREWALL_POLICY FIREWALL_RULE
```

Remove a rule from a given firewall policy.

Positional arguments

FIREWALL_POLICY	ID or name of firewall_policy to update
FIREWALL_RULE	Firewall rule to remove from policy

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron firewall-policy-show command

```
usage: neutron firewall-policy-show [-h] [-f {shell,table}] [-c COLUMN]
                                    [--variable VARIABLE] [--prefix PREFIX]
                                    [--request-format {json,xml}] [-D]
                                    [-F FIELD]
                                    FIREWALL_POLICY
```

Show information of a given firewall policy.

Positional arguments

FIREWALL_POLICY ID or name of firewall_policy to look up

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron firewall-policy-update command

```
usage: neutron firewall-policy-update [-h] [--request-format {json,xml}]
                                      FIREWALL_POLICY
```

Update a given firewall policy.

Positional arguments

FIREWALL_POLICY ID or name of firewall_policy to update

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron firewall-rule-create command

```
usage: neutron firewall-rule-create [-h] [-f {shell,table}] [-c COLUMN]
                                    [--variable VARIABLE] [--prefix PREFIX]
                                    [--request-format {json,xml}]
                                    [--tenant-id TENANT_ID] [--name NAME]
                                    [--description DESCRIPTION] [--shared]
                                    [--source-ip-address SOURCE_IP_ADDRESS]
                                    [--destination-ip-address
 DESTINATION_IP_ADDRESS]
                                    [--source-port SOURCE_PORT]
                                    [--destination-port DESTINATION_PORT]
                                    [--disabled] --protocol {tcp,udp,icmp,any}
                                    --action {allow,deny}
```

Create a firewall rule.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–name NAME	Name for the firewall rule
–description DESCRIPTION	Description for the firewall rule
–shared	Set shared to True (default False)
–source-ip-address SOURCE_IP_ADDRESS	Source ip address or subnet
–destination-ip-address DESTINATION_IP_ADDRESS	Destination ip address or subnet
–source-port SOURCE_PORT	Source port (integer in [1, 65535] or range in a:b)
–destination-port DESTINATION_PORT	Destination port (integer in [1, 65535] or range in a:b)
–disabled	To disable this rule
–protocol {tcp,udp,icmp,any}	Protocol for the firewall rule
–action {allow,deny}	Action for the firewall rule

neutron firewall-rule-delete command

```
usage: neutron firewall-rule-delete [-h] [--request-format {json,xml}]
                                    FIREWALL_RULE
```

Delete a given firewall rule.

Positional arguments

FIREWALL_RULE	ID or name of firewall_rule to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron firewall-rule-list command

```
usage: neutron firewall-rule-list [-h] [-f {csv,table}] [-c COLUMN]
                                  [--quote {all,minimal,none,nonnumeric}]
                                  [--request-format {json,xml}] [-D]
                                  [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                  [--sort-dir {asc,desc}]
```

List firewall rules that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron firewall-rule-show command

```
usage: neutron firewall-rule-show [-h] [-f {shell,table}] [-c COLUMN]
                                  [--variable VARIABLE] [--prefix PREFIX]
                                  [--request-format {json,xml}] [-D]
                                  [-F FIELD]
                                  FIREWALL_RULE
```

Show information of a given firewall rule.

Positional arguments

FIREWALL_RULE	ID or name of firewall_rule to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron firewall-rule-update command

```
usage: neutron firewall-rule-update [-h] [--request-format {json,xml}]
                                    [--protocol {tcp,udp,icmp,any}]
                                    FIREWALL_RULE
```

Update a given firewall rule.

Positional arguments

FIREWALL_RULE ID or name of firewall_rule to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–protocol {tcp,udp,icmp,any}	Protocol for the firewall rule

neutron firewall-show command

```
usage: neutron firewall-show [-h] [-f {shell,table}] [-c COLUMN]
                             [--variable VARIABLE] [--prefix PREFIX]
                             [--request-format {json,xml}] [-D] [-F FIELD]
                             FIREWALL
```

Show information of a given firewall.

Positional arguments

FIREWALL ID or name of firewall to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron firewall-update command

```
usage: neutron firewall-update [-h] [--request-format {json,xml}] FIREWALL
```

Update a given firewall.

Positional arguments

FIREWALL ID or name of firewall to update

Optional arguments

-h, –help	show this help message and exit

–request-format {json,xml} The xml or json request format

neutron floatingip-associate command

```
usage: neutron floatingip-associate [-h] [--request-format {json,xml}]
                                    [--fixed-ip-address FIXED_IP_ADDRESS]
                                    FLOATINGIP_ID PORT
```

Create a mapping between a floating ip and a fixed ip.

Positional arguments

FLOATINGIP_ID ID of the floating IP to associate

PORT ID or name of the port to be associated with the floatingip

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–fixed-ip-address FIXED_IP_ADDRESS IP address on the port (only required if port has multipleIPs)

neutron floatingip-create command

```
usage: neutron floatingip-create [-h] [-f {shell,table}] [-c COLUMN]
                                 [--variable VARIABLE] [--prefix PREFIX]
                                 [--request-format {json,xml}]
                                 [--tenant-id TENANT_ID] [--port-id PORT_ID]
                                 [--fixed-ip-address FIXED_IP_ADDRESS]
                                 FLOATING_NETWORK
```

Create a floating ip for a given tenant.

Positional arguments

FLOATING_NETWORK Network name or id to allocate floating IP from

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–tenant-id TENANT_ID The owner tenant ID

–port-id PORT_ID ID of the port to be associated with the floatingip

–fixed-ip-address FIXED_IP_ADDRESS IP address on the port (only required if port has multipleIPs)

neutron floatingip-delete command

```
usage: neutron floatingip-delete [-h] [--request-format {json,xml}] FLOATINGIP
```

Delete a given floating ip.

Positional arguments

FLOATINGIP ID of floatingip to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron floatingip-disassociate command

```
usage: neutron floatingip-disassociate [-h] [--request-format {json,xml}]
                                       FLOATINGIP_ID
```

Remove a mapping from a floating ip to a fixed ip.

Positional arguments

FLOATINGIP_ID ID of the floating IP to associate

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron floatingip-list command

```
usage: neutron floatingip-list [-h] [-f {csv,table}] [-c COLUMN]
                               [--quote {all,minimal,none,nonnumeric}]
                               [--request-format {json,xml}] [-D] [-F FIELD]
                               [-P SIZE] [--sort-key FIELD]
                               [--sort-dir {asc,desc}]
```

List floating ips that belong to a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron floatingip-show command

```
usage: neutron floatingip-show [-h] [-f {shell,table}] [-c COLUMN]
                               [--variable VARIABLE] [--prefix PREFIX]
                               [--request-format {json,xml}] [-D] [-F FIELD]
                               FLOATINGIP
```

Show information of a given floating ip.

Positional arguments

FLOATINGIP ID of floatingip to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron ipsec-site-connection-create command

```
usage: neutron ipsec-site-connection-create [-h] [-f {shell,table}]
                                            [-c COLUMN] [--variable VARIABLE]
                                            [--prefix PREFIX]
                                            [--request-format {json,xml}]
                                            [--tenant-id TENANT_ID]
                                            [--admin-state-down] [--name NAME]
                                            [--description DESCRIPTION]
                                            [--mtu MTU]
                                            [--initiator {bi-directional,
response-only}]
                                            [--dpd action=ACTION,interval=
INTERVAL,timeout=TIMEOUT]
                                            --vpnservice-id VPNSERVICE
                                            --ikepolicy-id IKEPOLICY
```

```
                                        --ipsecpolicy-id IPSECPOLICY
                                        --peer-address PEER_ADDRESS
                                        --peer-id PEER_ID --peer-cidr
                                        PEER_CIDRS --psk PSK
```

Create an IPsecSiteConnection.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set admin state up to false
–name NAME	Set friendly name for the connection
–description DESCRIPTION	Set a description for the connection
–mtu MTU MTU	size for the connection, default:1500
–initiator {bi-directional,response-only}	Initiator state in lowercase, default:bi-directional
–dpd	action=ACTION,interval=INTERVAL,timeout=TIMEOUT Ipsec connection Dead Peer Detection Attributes. 'action'-hold,clear,disabled,restart,restart-by-peer. 'interval' and 'timeout' are non negative integers. 'interval' should be less than 'timeout' value. 'action', default:hold 'interval', default:30, 'timeout', default:120.
–vpnservice-id VPNSERVICE	VPNService instance id associated with this connection
–ikepolicy-id IKEPOLICY	IKEPolicy id associated with this connection
–ipsecpolicy-id IPSECPOLICY	IPsecPolicy id associated with this connection
–peer-address PEER_ADDRESS	Peer gateway public IPv4/IPv6 address or FQDN.
–peer-id PEER_ID	Peer router identity for authentication. Can be IPv4/ IPv6 address, e-mail address, key id, or FQDN.
–peer-cidr PEER_CIDRS	Remote subnet(s) in CIDR format
–psk PSK	Pre-Shared Key string

neutron ipsec-site-connection-delete command

```
usage: neutron ipsec-site-connection-delete [-h] [--request-format {json,xml}]
                                            IPSEC_SITE_CONNECTION
```

Delete a given IPsecSiteConnection.

Positional arguments

IPSEC_SITE_CONNECTION ID or name of ipsec_site_connection to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron ipsec-site-connection-list command

```
usage: neutron ipsec-site-connection-list [-h] [-f {csv,table}] [-c COLUMN]
                                          [--quote {all,minimal,none,
nonnumeric}]
                                          [--request-format {json,xml}] [-D]
                                          [-F FIELD] [-P SIZE]
                                          [--sort-key FIELD]
                                          [--sort-dir {asc,desc}]
```

List IPsecSiteConnections that belong to a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

-P SIZE, –page-size SIZE Specify retrieve unit of each request, then split one request to several requests

–sort-key FIELD Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction

–sort-dir {asc,desc} Sort list in specified directions (This option can be repeated)

neutron ipsec-site-connection-show command

```
usage: neutron ipsec-site-connection-show [-h] [-f {shell,table}] [-c COLUMN]
                                          [--variable VARIABLE]
                                          [--prefix PREFIX]
                                          [--request-format {json,xml}] [-D]
                                          [-F FIELD]
                                          IPSEC_SITE_CONNECTION
```

Show information of a given IPsecSiteConnection.

Positional arguments

IPSEC_SITE_CONNECTION ID or name of ipsec_site_connection to look up

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron ipsec-site-connection-update command

```
usage: neutron ipsec-site-connection-update [-h] [--request-format {json,xml}]
                                            [--dpd action=ACTION,interval=
INTERVAL,timeout=TIMEOUT]
                                            IPSEC_SITE_CONNECTION
```

Update a given IPsecSiteConnection.

Positional arguments

IPSEC_SITE_CONNECTION ID or name of ipsec_site_connection to update

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–dpd action=ACTION,interval=INTERVAL,timeout=TIMEOUT Ipsec connection Dead Peer Detection Attributes. 'action'-hold,clear,disabled,restart,restart-by-peer. 'interval' and 'timeout' are non negative integers. 'interval' should be less than 'timeout' value. 'action', default:hold 'interval', default:30, 'timeout', default:120.

neutron l3-agent-list-hosting-router command

```
usage: neutron l3-agent-list-hosting-router [-h] [-f {csv,table}] [-c COLUMN]
                                            [--quote {all,minimal,none,
nonnumeric}]
                                            [--request-format {json,xml}] [-D]
                                            [-F FIELD]
                                            router
```

List L3 agents hosting a router.

Positional arguments

router Router to query

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron l3-agent-router-add command

```
usage: neutron l3-agent-router-add [-h] [--request-format {json,xml}]
                                   l3_agent router
```

Add a router to a L3 agent.

Positional arguments

l3_agent ID of the L3 agent

router Router to add

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron l3-agent-router-remove command

```
usage: neutron l3-agent-router-remove [-h] [--request-format {json,xml}]
                                      l3_agent router
```

Remove a router from a L3 agent.

Positional arguments

l3_agent ID of the L3 agent

router Router to remove

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-agent-hosting-pool command

```
usage: neutron lb-agent-hosting-pool [-h] [-f {csv,table}] [-c COLUMN]
                                     [--quote {all,minimal,none,nonnumeric}]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD]
                                     pool
```

Get loadbalancer agent hosting a pool. Deriving from ListCommand though server will return only one agent to keep common output format for all agent schedulers

Positional arguments

pool Pool to query

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-healthmonitor-associate command

```
usage: neutron lb-healthmonitor-associate [-h] [--request-format {json,xml}]
                                          HEALTH_MONITOR_ID POOL
```

Create a mapping between a health monitor and a pool.

Positional arguments

HEALTH_MONITOR_ID	Health monitor to associate
POOL	ID of the pool to be associated with the health monitor

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-healthmonitor-create command

```
usage: neutron lb-healthmonitor-create [-h] [-f {shell,table}] [-c COLUMN]
                                       [--variable VARIABLE] [--prefix PREFIX]
                                       [--request-format {json,xml}]
                                       [--tenant-id TENANT_ID]
                                       [--admin-state-down]
```

```
                                        [--expected-codes EXPECTED_CODES]
                                        [--http-method HTTP_METHOD]
                                        [--url-path URL_PATH] --delay DELAY
                                        --max-retries MAX_RETRIES --timeout
                                        TIMEOUT --type {PING,TCP,HTTP,HTTPS}
```

Create a healthmonitor.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set admin state up to false
–expected-codes EXPECTED_CODES	The list of HTTP status codes expected in response from the member to declare it healthy. This attribute can contain one value, or a list of values separated by comma, or a range of values (e.g. "200-299"). If this attribute is not specified, it defaults to "200".
–http-method HTTP_METHOD	The HTTP method used for requests by the monitor of type HTTP.
–url-path URL_PATH	The HTTP path used in the HTTP request used by the monitor to test a member health. This must be a string beginning with a / (forward slash)
–delay DELAY	The time in seconds between sending probes to members.
–max-retries MAX_RETRIES	Number of permissible connection failures before changing the member status to INACTIVE. [1..10]
–timeout TIMEOUT	Maximum number of seconds for a monitor to wait for a connection to be established before it times out. The value must be less than the delay value.
–type {PING,TCP,HTTP,HTTPS}	One of predefined health monitor types

neutron lb-healthmonitor-delete command

```
usage: neutron lb-healthmonitor-delete [-h] [--request-format {json,xml}]
                                       HEALTH_MONITOR
```

Delete a given healthmonitor.

Positional arguments

HEALTH_MONITOR ID or name of health_monitor to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron lb-healthmonitor-disassociate command

```
usage: neutron lb-healthmonitor-disassociate [-h]
                                             [--request-format {json,xml}]
                                             HEALTH_MONITOR_ID POOL
```

Remove a mapping from a health monitor to a pool.

Positional arguments

HEALTH_MONITOR_ID Health monitor to associate

POOL ID of the pool to be associated with the health monitor

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron lb-healthmonitor-list command

```
usage: neutron lb-healthmonitor-list [-h] [-f {csv,table}] [-c COLUMN]
                                     [--quote {all,minimal,none,nonnumeric}]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                     [--sort-dir {asc,desc}]
```

List healthmonitors that belong to a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

-P SIZE, –page-size SIZE Specify retrieve unit of each request, then split one request to several requests

–sort-key FIELD Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each

	other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron lb-healthmonitor-show command

```
usage: neutron lb-healthmonitor-show [-h] [-f {shell,table}] [-c COLUMN]
                                     [--variable VARIABLE] [--prefix PREFIX]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD]
                                     HEALTH_MONITOR
```

Show information of a given healthmonitor.

Positional arguments

HEALTH_MONITOR ID or name of health_monitor to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-healthmonitor-update command

```
usage: neutron lb-healthmonitor-update [-h] [--request-format {json,xml}]
                                       HEALTH_MONITOR
```

Update a given healthmonitor.

Positional arguments

HEALTH_MONITOR ID or name of health_monitor to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-member-create command

```
usage: neutron lb-member-create [-h] [-f {shell,table}] [-c COLUMN]
                                [--variable VARIABLE] [--prefix PREFIX]
```

```
                                [--request-format {json,xml}]
                                [--tenant-id TENANT_ID] [--admin-state-down]
                                [--weight WEIGHT] --address ADDRESS
                                --protocol-port PROTOCOL_PORT
                                POOL
```

Create a member.

Positional arguments

POOL Pool id or name this vip belongs to

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set admin state up to false
–weight WEIGHT	Weight of pool member in the pool (default:1, [0..256])
–address ADDRESS IP	address of the pool member on the pool network.
–protocol-port PROTOCOL_PORT	Port on which the pool member listens for requests or connections.

neutron lb-member-delete command

```
usage: neutron lb-member-delete [-h] [--request-format {json,xml}] MEMBER
```

Delete a given member.

Positional arguments

MEMBER ID or name of member to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-member-list command

```
usage: neutron lb-member-list [-h] [-f {csv,table}] [-c COLUMN]
                              [--quote {all,minimal,none,nonnumeric}]
                              [--request-format {json,xml}] [-D] [-F FIELD]
                              [-P SIZE] [--sort-key FIELD]
                              [--sort-dir {asc,desc}]
```

List members that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron lb-member-show command

```
usage: neutron lb-member-show [-h] [-f {shell,table}] [-c COLUMN]
                              [--variable VARIABLE] [--prefix PREFIX]
                              [--request-format {json,xml}] [-D] [-F FIELD]
                              MEMBER
```

Show information of a given member.

Positional arguments

MEMBER ID or name of member to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-member-update command

```
usage: neutron lb-member-update [-h] [--request-format {json,xml}] MEMBER
```

Update a given member.

Positional arguments

MEMBER ID or name of member to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-pool-create command

```
usage: neutron lb-pool-create [-h] [-f {shell,table}] [-c COLUMN]
                              [--variable VARIABLE] [--prefix PREFIX]
                              [--request-format {json,xml}]
                              [--tenant-id TENANT_ID] [--admin-state-down]
                              [--description DESCRIPTION] --lb-method
                              {ROUND_ROBIN,LEAST_CONNECTIONS,SOURCE_IP} --name
                              NAME --protocol {HTTP,HTTPS,TCP} --subnet-id
                              SUBNET [--provider PROVIDER]
```

Create a pool.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set admin state up to false
–description DESCRIPTION	Description of the pool
–lb-method {ROUND_ROBIN,LEAST_CONNECTIONS,SOURCE_IP}	The algorithm used to distribute load between the members of the pool
–name NAME	The name of the pool
–protocol {HTTP,HTTPS,TCP}	Protocol for balancing
–subnet-id SUBNET	The subnet on which the members of the pool will be located
–provider PROVIDER	Provider name of loadbalancer service

neutron lb-pool-delete command

```
usage: neutron lb-pool-delete [-h] [--request-format {json,xml}] POOL
```

Delete a given pool.

Positional arguments

POOL ID or name of pool to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-pool-list command

```
usage: neutron lb-pool-list [-h] [-f {csv,table}] [-c COLUMN]
                            [--quote {all,minimal,none,nonnumeric}]
                            [--request-format {json,xml}] [-D] [-F FIELD]
                            [-P SIZE] [--sort-key FIELD]
                            [--sort-dir {asc,desc}]
```

List pools that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron lb-pool-list-on-agent command

```
usage: neutron lb-pool-list-on-agent [-h] [-f {csv,table}] [-c COLUMN]
                                     [--quote {all,minimal,none,nonnumeric}]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD]
                                     lbaas_agent
```

List the pools on a loadbalancer agent.

Positional arguments

lbaas_agent ID of the loadbalancer agent to query

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-pool-show command

```
usage: neutron lb-pool-show [-h] [-f {shell,table}] [-c COLUMN]
                            [--variable VARIABLE] [--prefix PREFIX]
                            [--request-format {json,xml}] [-D] [-F FIELD]
                            POOL
```

Show information of a given pool.

Positional arguments

POOL ID or name of pool to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-pool-stats command

```
usage: neutron lb-pool-stats [-h] [-f {shell,table}] [-c COLUMN]
                             [--variable VARIABLE] [--prefix PREFIX]
                             [--request-format {json,xml}] [-D] [-F FIELD]
                             POOL
```

Retrieve stats for a given pool.

Positional arguments

POOL ID or name of pool to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-pool-update command

```
usage: neutron lb-pool-update [-h] [--request-format {json,xml}] POOL
```

Update a given pool.

Positional arguments

POOL ID or name of pool to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-vip-create command

```
usage: neutron lb-vip-create [-h] [-f {shell,table}] [-c COLUMN]
                             [--variable VARIABLE] [--prefix PREFIX]
                             [--request-format {json,xml}]
                             [--tenant-id TENANT_ID] [--address ADDRESS]
                             [--admin-state-down]
                             [--connection-limit CONNECTION_LIMIT]
                             [--description DESCRIPTION] --name NAME
                             --protocol-port PROTOCOL_PORT --protocol
                             {TCP,HTTP,HTTPS} --subnet-id SUBNET
                             POOL
```

Create a vip.

Positional arguments

POOL Pool id or name this vip belongs to

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID

–address ADDRESS IP	address of the vip
–admin-state-down	Set admin state up to false
–connection-limit CONNECTION_LIMIT	The maximum number of connections per second allowed for the vip. Positive integer or -1 for unlimited (default)
–description DESCRIPTION	Description of the vip
–name NAME	Name of the vip
–protocol-port PROTOCOL_PORT	TCP port on which to listen for client traffic that is associated with the vip address
–protocol {TCP,HTTP,HTTPS}	Protocol for balancing
–subnet-id SUBNET	The subnet on which to allocate the vip address

neutron lb-vip-delete command

```
usage: neutron lb-vip-delete [-h] [--request-format {json,xml}] VIP
```

Delete a given vip.

Positional arguments

VIP ID or name of vip to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron lb-vip-list command

```
usage: neutron lb-vip-list [-h] [-f {csv,table}] [-c COLUMN]
                           [--quote {all,minimal,none,nonnumeric}]
                           [--request-format {json,xml}] [-D] [-F FIELD]
                           [-P SIZE] [--sort-key FIELD]
                           [--sort-dir {asc,desc}]
```

List vips that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info

-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron lb-vip-show command

```
usage: neutron lb-vip-show [-h] [-f {shell,table}] [-c COLUMN]
                           [--variable VARIABLE] [--prefix PREFIX]
                           [--request-format {json,xml}] [-D] [-F FIELD]
                           VIP
```

Show information of a given vip.

Positional arguments

VIP ID or name of vip to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron lb-vip-update command

```
usage: neutron lb-vip-update [-h] [--request-format {json,xml}] VIP
```

Update a given vip.

Positional arguments

VIP ID or name of vip to update

Optional arguments

-h, –help	show this help message and exit

–request-format {json,xml} The xml or json request format

neutron meter-label-create command

```
usage: neutron meter-label-create [-h] [-f {shell,table}] [-c COLUMN]
                                  [--variable VARIABLE] [--prefix PREFIX]
                                  [--request-format {json,xml}]
                                  [--tenant-id TENANT_ID]
                                  [--description DESCRIPTION]
                                  NAME
```

Create a metering label for a given tenant.

Positional arguments

NAME Name of metering label to create

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–tenant-id TENANT_ID The owner tenant ID

–description DESCRIPTION Description of metering label to create

neutron meter-label-delete command

```
usage: neutron meter-label-delete [-h] [--request-format {json,xml}]
                                  METERING_LABEL
```

Delete a given metering label.

Positional arguments

METERING_LABEL ID or name of metering_label to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron meter-label-list command

```
usage: neutron meter-label-list [-h] [-f {csv,table}] [-c COLUMN]
                                [--quote {all,minimal,none,nonnumeric}]
                                [--request-format {json,xml}] [-D] [-F FIELD]
                                [-P SIZE] [--sort-key FIELD]
                                [--sort-dir {asc,desc}]
```

List metering labels that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron meter-label-rule-create command

```
usage: neutron meter-label-rule-create [-h] [-f {shell,table}] [-c COLUMN]
                                       [--variable VARIABLE] [--prefix PREFIX]
                                       [--request-format {json,xml}]
                                       [--tenant-id TENANT_ID]
                                       [--direction {ingress,egress}]
                                       [--excluded]
                                       LABEL REMOTE_IP_PREFIX
```

Create a metering label rule for a given label.

Positional arguments

LABEL	Id or Name of the label
REMOTE_IP_PREFIX	CIDR to match on

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–direction {ingress,egress}	Direction of traffic, default:ingress
–excluded	Exclude this cidr from the label, default:not excluded

neutron meter-label-rule-delete command

```
usage: neutron meter-label-rule-delete [-h] [--request-format {json,xml}]
                                       METERING_LABEL_RULE
```

Delete a given metering label.

Positional arguments

METERING_LABEL_RULE ID or name of metering_label_rule to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron meter-label-rule-list command

```
usage: neutron meter-label-rule-list [-h] [-f {csv,table}] [-c COLUMN]
                                     [--quote {all,minimal,none,nonnumeric}]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                     [--sort-dir {asc,desc}]
```

List metering labels that belong to a given label.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

-P SIZE, –page-size SIZE Specify retrieve unit of each request, then split one request to several requests

–sort-key FIELD Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction

–sort-dir {asc,desc} Sort list in specified directions (This option can be repeated)

neutron meter-label-rule-show command

```
usage: neutron meter-label-rule-show [-h] [-f {shell,table}] [-c COLUMN]
```

```
                                    [--variable VARIABLE] [--prefix PREFIX]
                                    [--request-format {json,xml}] [-D]
                                    [-F FIELD]
                                    METERING_LABEL_RULE
```

Show information of a given metering label rule.

Positional arguments

METERING_LABEL_RULE ID or name of metering_label_rule to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron meter-label-show command

```
usage: neutron meter-label-show [-h] [-f {shell,table}] [-c COLUMN]
                                [--variable VARIABLE] [--prefix PREFIX]
                                [--request-format {json,xml}] [-D] [-F FIELD]
                                METERING_LABEL
```

Show information of a given metering label.

Positional arguments

METERING_LABEL ID or name of metering_label to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron net-create command

```
usage: neutron net-create [-h] [-f {shell,table}] [-c COLUMN]
                          [--variable VARIABLE] [--prefix PREFIX]
                          [--request-format {json,xml}]
                          [--tenant-id TENANT_ID] [--admin-state-down]
                          [--shared]
```

```
                                NAME
```

Create a network for a given tenant.

Positional arguments

NAME Name of network to create

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set Admin State Up to false
–shared	Set the network as shared

neutron net-delete command

```
usage: neutron net-delete [-h] [--request-format {json,xml}] NETWORK
```

Delete a given network.

Positional arguments

NETWORK ID or name of network to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron net-external-list command

```
usage: neutron net-external-list [-h] [-f {csv,table}] [-c COLUMN]
                                 [--quote {all,minimal,none,nonnumeric}]
                                 [--request-format {json,xml}] [-D] [-F FIELD]
                                 [-P SIZE] [--sort-key FIELD]
                                 [--sort-dir {asc,desc}]
```

List external networks that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron net-gateway-connect command

```
usage: neutron net-gateway-connect [-h] [--request-format {json,xml}]
                                   [--segmentation-type SEGMENTATION_TYPE]
                                   [--segmentation-id SEGMENTATION_ID]
                                   NET-GATEWAY-ID NETWORK-ID
```

Add an internal network interface to a router.

Positional arguments

NET-GATEWAY-ID	ID of the network gateway
NETWORK-ID	ID of the internal network to connect on the gateway

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–segmentation-type SEGMENTATION_TYPE	L2 segmentation strategy on the external side of the gateway (e.g.: VLAN, FLAT)
–segmentation-id SEGMENTATION_ID	Identifier for the L2 segment on the external side of the gateway

neutron net-gateway-create command

```
usage: neutron net-gateway-create [-h] [-f {shell,table}] [-c COLUMN]
                                  [--variable VARIABLE] [--prefix PREFIX]
                                  [--request-format {json,xml}]
                                  [--tenant-id TENANT_ID] [--device DEVICE]
                                  NAME
```

Create a network gateway.

Positional arguments

NAME Name of network gateway to create

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–device DEVICE	Device info for this gateway device_id=<device identifier>,interface_name=<name_or_identifier> It can be repeated for multiple devices for HA gateways

neutron net-gateway-delete command

```
usage: neutron net-gateway-delete [-h] [--request-format {json,xml}]
                                  NETWORK_GATEWAY
```

Delete a given network gateway.

Positional arguments

NETWORK_GATEWAYID or name of network_gateway to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron net-gateway-disconnect command

```
usage: neutron net-gateway-disconnect [-h] [--request-format {json,xml}]
                                      [--segmentation-type SEGMENTATION_TYPE]
                                      [--segmentation-id SEGMENTATION_ID]
                                      NET-GATEWAY-ID NETWORK-ID
```

Remove a network from a network gateway.

Positional arguments

NET-GATEWAY-ID	ID of the network gateway
NETWORK-ID	ID of the internal network to connect on the gateway

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

–segmentation-type SEGMENTATION_TYPE	L2 segmentation strategy on the external side of the gateway (e.g.: VLAN, FLAT)
–segmentation-id SEGMENTATION_ID	Identifier for the L2 segment on the external side of the gateway

neutron net-gateway-list command

```
usage: neutron net-gateway-list [-h] [-f {csv,table}] [-c COLUMN]
                                [--quote {all,minimal,none,nonnumeric}]
                                [--request-format {json,xml}] [-D] [-F FIELD]
```

List network gateways for a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron net-gateway-show command

```
usage: neutron net-gateway-show [-h] [-f {shell,table}] [-c COLUMN]
                                [--variable VARIABLE] [--prefix PREFIX]
                                [--request-format {json,xml}] [-D] [-F FIELD]
                                NETWORK_GATEWAY
```

Show information of a given network gateway.

Positional arguments

NETWORK_GATEWAYID or name of network_gateway to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron net-gateway-update command

```
usage: neutron net-gateway-update [-h] [--request-format {json,xml}]
```

```
                                    NETWORK_GATEWAY
```

Update the name for a network gateway.

Positional arguments

NETWORK_GATEWAYID or name of network_gateway to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron net-list command

```
usage: neutron net-list [-h] [-f {csv,table}] [-c COLUMN]
                        [--quote {all,minimal,none,nonnumeric}]
                        [--request-format {json,xml}] [-D] [-F FIELD]
                        [-P SIZE] [--sort-key FIELD] [--sort-dir {asc,desc}]
```

List networks that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron net-list-on-dhcp-agent command

```
usage: neutron net-list-on-dhcp-agent [-h] [-f {csv,table}] [-c COLUMN]
                                      [--quote {all,minimal,none,nonnumeric}]
                                      [--request-format {json,xml}] [-D]
                                      [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                      [--sort-dir {asc,desc}]
```

```
                                        dhcp_agent
```

List the networks on a DHCP agent.

Positional arguments

dhcp_agent ID of the DHCP agent

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron net-show command

```
usage: neutron net-show [-h] [-f {shell,table}] [-c COLUMN]
                        [--variable VARIABLE] [--prefix PREFIX]
                        [--request-format {json,xml}] [-D] [-F FIELD]
                        NETWORK
```

Show information of a given network.

Positional arguments

NETWORK ID or name of network to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron net-update command

```
usage: neutron net-update [-h] [--request-format {json,xml}] NETWORK
```

Update network's information.

Positional arguments

NETWORK ID or name of network to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron port-create command

```
usage: neutron port-create [-h] [-f {shell,table}] [-c COLUMN]
                           [--variable VARIABLE] [--prefix PREFIX]
                           [--request-format {json,xml}]
                           [--tenant-id TENANT_ID] [--name NAME]
                           [--admin-state-down] [--mac-address MAC_ADDRESS]
                           [--device-id DEVICE_ID]
                           [--fixed-ip subnet_id=SUBNET,ip_address=IP_ADDR]
                           [--security-group SECURITY_GROUP | --no-security-
groups]
                           [--extra-dhcp-opt EXTRA_DHCP_OPTS]
                           NETWORK
```

Create a port for a given tenant.

Positional arguments

NETWORK Network id or name this port belongs to

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–name NAME	Name of this port
–admin-state-down	Set admin state up to false
–mac-address MAC_ADDRESS	MAC address of this port
–device-id DEVICE_ID	Device id of this port

–fixed-ip	subnet_id=SUBNET,ip_address=IP_ADDR Desired IP and/or subnet for this port: subnet_id=<name_or_id>,ip_address=<ip>, (This option can be repeated.)
–security-group SECURITY_GROUP	Security group associated with the port (This option can be repeated)
–no-security-groups	Associate no security groups with the port
–extra-dhcp-opt EXTRA_DHCP_OPTS	Extra dhcp options to be assigned to this port: opt_name=<dhcp_option_name>,opt_value=<value>, (This option can be repeated.)

neutron port-delete command

```
usage: neutron port-delete [-h] [--request-format {json,xml}] PORT
```

Delete a given port.

Positional arguments

PORT ID or name of port to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron port-list command

```
usage: neutron port-list [-h] [-f {csv,table}] [-c COLUMN]
                         [--quote {all,minimal,none,nonnumeric}]
                         [--request-format {json,xml}] [-D] [-F FIELD]
                         [-P SIZE] [--sort-key FIELD] [--sort-dir {asc,desc}]
```

List ports that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests

–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron port-show command

```
usage: neutron port-show [-h] [-f {shell,table}] [-c COLUMN]
                         [--variable VARIABLE] [--prefix PREFIX]
                         [--request-format {json,xml}] [-D] [-F FIELD]
                         PORT
```

Show information of a given port.

Positional arguments

PORT ID or name of port to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron port-update command

```
usage: neutron port-update [-h] [--request-format {json,xml}]
                           [--security-group SECURITY_GROUP | --no-security-
groups]
                           [--extra-dhcp-opt EXTRA_DHCP_OPTS]
                           PORT
```

Update port's information.

Positional arguments

PORT ID or name of port to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

–security-group SECURITY_GROUP	Security group associated with the port (This option can be repeated)
–no-security-groups	Associate no security groups with the port
–extra-dhcp-opt EXTRA_DHCP_OPTS	Extra dhcp options to be assigned to this port: opt_name=<dhcp_option_name>,opt_value=<value>, (This option can be repeated.)

neutron queue-create command

```
usage: neutron queue-create [-h] [-f {shell,table}] [-c COLUMN]
                            [--variable VARIABLE] [--prefix PREFIX]
                            [--request-format {json,xml}]
                            [--tenant-id TENANT_ID] [--min MIN] [--max MAX]
                            [--qos-marking QOS_MARKING] [--default DEFAULT]
                            [--dscp DSCP]
                            NAME
```

Create a queue.

Positional arguments

NAME Name of queue

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–min MIN	min-rate
–max MAX	max-rate
–qos-marking QOS_MARKING	QOS marking untrusted/trusted
–default DEFAULT	If true all ports created with be the size of this queue if queue is not specified
–dscp DSCP	Differentiated Services Code Point

neutron queue-delete command

```
usage: neutron queue-delete [-h] [--request-format {json,xml}] QOS_QUEUE
```

Delete a given queue.

Positional arguments

QOS_QUEUE ID or name of qos_queue to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron queue-list command

```
usage: neutron queue-list [-h] [-f {csv,table}] [-c COLUMN]
                          [--quote {all,minimal,none,nonnumeric}]
                          [--request-format {json,xml}] [-D] [-F FIELD]
```

List queues that belong to a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron queue-show command

```
usage: neutron queue-show [-h] [-f {shell,table}] [-c COLUMN]
                          [--variable VARIABLE] [--prefix PREFIX]
                          [--request-format {json,xml}] [-D] [-F FIELD]
                          QOS_QUEUE
```

Show information of a given queue.

Positional arguments

QOS_QUEUE ID or name of qos_queue to look up

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron quota-delete command

```
usage: neutron quota-delete [-h] [--request-format {json,xml}]
```

```
                              [--tenant-id tenant-id]
```

Delete defined quotas of a given tenant.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–tenant-id tenant-id The owner tenant ID

neutron quota-list command

```
usage: neutron quota-list [-h] [-f {csv,table}] [-c COLUMN]
                          [--quote {all,minimal,none,nonnumeric}]
                          [--request-format {json,xml}]
```

List quotas of all tenants who have non-default quota values.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron quota-show command

```
usage: neutron quota-show [-h] [-f {shell,table}] [-c COLUMN]
                          [--variable VARIABLE] [--prefix PREFIX]
                          [--request-format {json,xml}]
                          [--tenant-id tenant-id]
```

Show quotas of a given tenant

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–tenant-id tenant-id The owner tenant ID

neutron quota-update command

```
usage: neutron quota-update [-h] [-f {shell,table}] [-c COLUMN]
                            [--variable VARIABLE] [--prefix PREFIX]
                            [--request-format {json,xml}]
                            [--tenant-id tenant-id] [--network networks]
                            [--subnet subnets] [--port ports]
```

```
                              [--router routers] [--floatingip floatingips]
                              [--security-group security_groups]
                              [--security-group-rule security_group_rules]
```

Define tenant's quotas not to use defaults.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id	tenant-id The owner tenant ID
–network	networks The limit of networks
–subnet	subnets The limit of subnets
–port	ports The limit of ports
–router	routers The limit of routers
–floatingip	floatingips The limit of floating IPs
–security-group	security_groups The limit of security groups
–security-group-rule	security_group_rules The limit of security groups rules

neutron router-create command

```
usage: neutron router-create [-h] [-f {shell,table}] [-c COLUMN]
                             [--variable VARIABLE] [--prefix PREFIX]
                             [--request-format {json,xml}]
                             [--tenant-id TENANT_ID] [--admin-state-down]
                             NAME
```

Create a router for a given tenant.

Positional arguments

NAME	Name of router to create
distributed	Create a distributed router (VMware NSX plugin only)

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set Admin State Up to false

neutron router-delete command

```
usage: neutron router-delete [-h] [--request-format {json,xml}] ROUTER
```

Delete a given router.

Positional arguments

ROUTER ID or name of router to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron router-gateway-clear command

```
usage: neutron router-gateway-clear [-h] [--request-format {json,xml}]
                                    router-id
```

Remove an external network gateway from a router.

Positional arguments

router-id ID of the router

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron router-gateway-set command

```
usage: neutron router-gateway-set [-h] [--request-format {json,xml}]
                                  [--disable-snat]
                                  router-id external-network-id
```

Set the external network gateway for a router.

Positional arguments

router-id ID of the router

external-network-id ID of the external network for the gateway

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–disable-snat Disable Source NAT on the router gateway

neutron router-interface-add command

```
usage: neutron router-interface-add [-h] [--request-format {json,xml}]
                                    router-id INTERFACE
```

Add an internal network interface to a router.

Positional arguments

router-id ID of the router

INTERFACE The format is "SUBNET|subnet=SUBNET|port=PORT". Either a subnet or port must be specified. Both ID and name are accepted as SUBNET or PORT. Note that "subnet=" can be omitted when specifying subnet.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron router-interface-delete command

```
usage: neutron router-interface-delete [-h] [--request-format {json,xml}]
                                       router-id INTERFACE
```

Remove an internal network interface from a router.

Positional arguments

router-id ID of the router

INTERFACE The format is "SUBNET|subnet=SUBNET|port=PORT". Either a subnet or port must be specified. Both ID and name are accepted as SUBNET or PORT. Note that "subnet=" can be omitted when specifying subnet.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron router-list command

```
usage: neutron router-list [-h] [-f {csv,table}] [-c COLUMN]
                           [--quote {all,minimal,none,nonnumeric}]
```

```
                              [--request-format {json,xml}] [-D] [-F FIELD]
                              [-P SIZE] [--sort-key FIELD]
                              [--sort-dir {asc,desc}]
```

List routers that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron router-list-on-l3-agent command

```
usage: neutron router-list-on-l3-agent [-h] [-f {csv,table}] [-c COLUMN]
                                       [--quote {all,minimal,none,nonnumeric}]
                                       [--request-format {json,xml}] [-D]
                                       [-F FIELD]
                                       l3_agent
```

List the routers on a L3 agent.

Positional arguments

l3_agent	ID of the L3 agent to query

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron router-port-list command

```
usage: neutron router-port-list [-h] [-f {csv,table}] [-c COLUMN]
                                [--quote {all,minimal,none,nonnumeric}]
                                [--request-format {json,xml}] [-D] [-F FIELD]
                                [-P SIZE] [--sort-key FIELD]
                                [--sort-dir {asc,desc}]
                                router
```

List ports that belong to a given tenant, with specified router.

Positional arguments

router ID or name of router to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron router-show command

```
usage: neutron router-show [-h] [-f {shell,table}] [-c COLUMN]
                           [--variable VARIABLE] [--prefix PREFIX]
                           [--request-format {json,xml}] [-D] [-F FIELD]
                           ROUTER
```

Show information of a given router.

Positional arguments

ROUTER ID or name of router to look up

Optional arguments

-h, –help	show this help message and exit

–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron router-update command

```
usage: neutron router-update [-h] [--request-format {json,xml}] ROUTER
```

Update router's information.

Positional arguments

ROUTER ID or name of router to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron security-group-create command

```
usage: neutron security-group-create [-h] [-f {shell,table}] [-c COLUMN]
                                     [--variable VARIABLE] [--prefix PREFIX]
                                     [--request-format {json,xml}]
                                     [--tenant-id TENANT_ID]
                                     [--description DESCRIPTION]
                                     NAME
```

Create a security group.

Positional arguments

NAME Name of security group

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–description DESCRIPTION	Description of security group

neutron security-group-delete command

```
usage: neutron security-group-delete [-h] [--request-format {json,xml}]
```

```
                                        SECURITY_GROUP
```

Delete a given security group.

Positional arguments

SECURITY_GROUP ID or name of security_group to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron security-group-list command

```
usage: neutron security-group-list [-h] [-f {csv,table}] [-c COLUMN]
                                   [--quote {all,minimal,none,nonnumeric}]
                                   [--request-format {json,xml}] [-D]
                                   [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                   [--sort-dir {asc,desc}]
```

List security groups that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron security-group-rule-create command

```
usage: neutron security-group-rule-create [-h] [-f {shell,table}] [-c COLUMN]
                                          [--variable VARIABLE]
                                          [--prefix PREFIX]
                                          [--request-format {json,xml}]
```

```
                                        [--tenant-id TENANT_ID]
                                        [--direction {ingress,egress}]
                                        [--ethertype ETHERTYPE]
                                        [--protocol PROTOCOL]
                                        [--port-range-min PORT_RANGE_MIN]
                                        [--port-range-max PORT_RANGE_MAX]
                                        [--remote-ip-prefix
REMOTE_IP_PREFIX]
                                        [--remote-group-id REMOTE_GROUP]
                                        SECURITY_GROUP
```

Create a security group rule.

Positional arguments

SECURITY_GROUP Security group name or id to add rule.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–direction {ingress,egress}	Direction of traffic: ingress/egress
–ethertype ETHERTYPE	IPv4/IPv6
–protocol PROTOCOL	Protocol of packet
–port-range-min PORT_RANGE_MIN	Starting port range
–port-range-max PORT_RANGE_MAX	Ending port range
–remote-ip-prefix REMOTE_IP_PREFIX	CIDR to match on
–remote-group-id REMOTE_GROUP	Remote security group name or id to apply rule

neutron security-group-rule-delete command

```
usage: neutron security-group-rule-delete [-h] [--request-format {json,xml}]
                                          SECURITY_GROUP_RULE
```

Delete a given security group rule.

Positional arguments

SECURITY_GROUP_RULE ID of security_group_rule to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron security-group-rule-list command

```
usage: neutron security-group-rule-list [-h] [-f {csv,table}] [-c COLUMN]
                                        [--quote {all,minimal,none,
nonnumeric}]
                                        [--request-format {json,xml}] [-D]
                                        [-F FIELD] [-P SIZE]
                                        [--sort-key FIELD]
                                        [--sort-dir {asc,desc}]
                                        [--no-nameconv]
```

List security group rules that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)
–no-nameconv	Do not convert security group ID to its name

neutron security-group-rule-show command

```
usage: neutron security-group-rule-show [-h] [-f {shell,table}] [-c COLUMN]
                                        [--variable VARIABLE]
                                        [--prefix PREFIX]
                                        [--request-format {json,xml}] [-D]
                                        [-F FIELD]
                                        SECURITY_GROUP_RULE
```

Show information of a given security group rule.

Positional arguments

SECURITY_GROUP_RULE ID of security_group_rule to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron security-group-show command

```
usage: neutron security-group-show [-h] [-f {shell,table}] [-c COLUMN]
                                   [--variable VARIABLE] [--prefix PREFIX]
                                   [--request-format {json,xml}] [-D]
                                   [-F FIELD]
                                   SECURITY_GROUP
```

Show information of a given security group.

Positional arguments

SECURITY_GROUP ID or name of security_group to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron security-group-update command

```
usage: neutron security-group-update [-h] [--request-format {json,xml}]
                                     [--name NAME] [--description DESCRIPTION]
                                     SECURITY_GROUP
```

Update a given security group.

Positional arguments

SECURITY_GROUP ID or name of security_group to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–name NAME	Name of security group
–description DESCRIPTION	Description of security group

neutron service-provider-list command

```
usage: neutron service-provider-list [-h] [-f {csv,table}] [-c COLUMN]
                                     [--quote {all,minimal,none,nonnumeric}]
                                     [--request-format {json,xml}] [-D]
                                     [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                     [--sort-dir {asc,desc}]
```

List service providers.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron subnet-create command

```
usage: neutron subnet-create [-h] [-f {shell,table}] [-c COLUMN]
                             [--variable VARIABLE] [--prefix PREFIX]
                             [--request-format {json,xml}]
                             [--tenant-id TENANT_ID] [--name NAME]
                             [--ip-version {4,6}] [--gateway GATEWAY_IP]
                             [--no-gateway]
                             [--allocation-pool start=IP_ADDR,end=IP_ADDR]
                             [--host-route destination=CIDR,nexthop=IP_ADDR]
                             [--dns-nameserver DNS_NAMESERVER]
                             [--disable-dhcp]
```

```
                                NETWORK CIDR
```

Create a subnet for a given tenant.

Positional arguments

NETWORK Network id or name this subnet belongs to

CIDR CIDR of subnet to create

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–name NAME	Name of this subnet
–ip-version {4,6} IP	version with default 4
–gateway GATEWAY_IP	Gateway ip of this subnet
–no-gateway	No distribution of gateway
–allocation-pool	start=IP_ADDR,end=IP_ADDR Allocation pool IP addresses for this subnet (This option can be repeated)
–host-route	destination=CIDR,nexthop=IP_ADDR Additional route (This option can be repeated)
–dns-nameserver DNS_NAMESERVER	DNS name server for this subnet (This option can be repeated)
–disable-dhcp	Disable DHCP for this subnet

neutron subnet-delete command

```
usage: neutron subnet-delete [-h] [--request-format {json,xml}] SUBNET
```

Delete a given subnet.

Positional arguments

SUBNET ID or name of subnet to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron subnet-list command

```
usage: neutron subnet-list [-h] [-f {csv,table}] [-c COLUMN]
                           [--quote {all,minimal,none,nonnumeric}]
                           [--request-format {json,xml}] [-D] [-F FIELD]
                           [-P SIZE] [--sort-key FIELD]
                           [--sort-dir {asc,desc}]
```

List subnets that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron subnet-show command

```
usage: neutron subnet-show [-h] [-f {shell,table}] [-c COLUMN]
                           [--variable VARIABLE] [--prefix PREFIX]
                           [--request-format {json,xml}] [-D] [-F FIELD]
                           SUBNET
```

Show information of a given subnet.

Positional arguments

SUBNET ID or name of subnet to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron subnet-update command

```
usage: neutron subnet-update [-h] [--request-format {json,xml}] SUBNET
```

Update subnet's information.

Positional arguments

SUBNET ID or name of subnet to update

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron vpn-ikepolicy-create command

```
usage: neutron vpn-ikepolicy-create [-h] [-f {shell,table}] [-c COLUMN]
                                    [--variable VARIABLE] [--prefix PREFIX]
                                    [--request-format {json,xml}]
                                    [--tenant-id TENANT_ID]
                                    [--description DESCRIPTION]
                                    [--auth-algorithm {sha1}]
                                    [--encryption-algorithm {3des,aes-128,
aes-192,aes-256}]
                                    [--phase1-negotiation-mode {main}]
                                    [--ike-version {v1,v2}]
                                    [--pfs {group2,group5,group14}]
                                    [--lifetime units=UNITS,value=VALUE]
                                    NAME
```

Create an IKEPolicy.

Positional arguments

NAME Name of the IKE Policy

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

–tenant-id TENANT_ID The owner tenant ID

–description DESCRIPTION Description of the IKE policy

–auth-algorithm {sha1} Authentication algorithm in lowercase. default:sha1

–encryption-algorithm {3des,aes-128,aes-192,aes-256}	Encryption Algorithm in lowercase, default:aes-128
–phase1-negotiation-mode {main}	IKE Phase1 negotiation mode in lowercase, default:main
–ike-version {v1,v2}	IKE version in lowercase, default:v1
–pfs {group2,group5,group14}	Perfect Forward Secrecy in lowercase, default:group5
–lifetime	units=UNITS,value=VALUE IKE Lifetime Attributes.'units'-seconds,default:seconds. 'value'-non negative integer, default:3600.

neutron vpn-ikepolicy-delete command

```
usage: neutron vpn-ikepolicy-delete [-h] [--request-format {json,xml}]
                                    IKEPOLICY
```

Delete a given IKE Policy.

Positional arguments

IKEPOLICY ID or name of ikepolicy to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron vpn-ikepolicy-list command

```
usage: neutron vpn-ikepolicy-list [-h] [-f {csv,table}] [-c COLUMN]
                                  [--quote {all,minimal,none,nonnumeric}]
                                  [--request-format {json,xml}] [-D]
                                  [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                  [--sort-dir {asc,desc}]
```

List IKEPolicies that belong to a tenant.

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

-P SIZE, –page-size SIZE Specify retrieve unit of each request, then split one request to several requests

–sort-key FIELD Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction

–sort-dir {asc,desc} Sort list in specified directions (This option can be repeated)

neutron vpn-ikepolicy-show command

```
usage: neutron vpn-ikepolicy-show [-h] [-f {shell,table}] [-c COLUMN]
                                  [--variable VARIABLE] [--prefix PREFIX]
                                  [--request-format {json,xml}] [-D]
                                  [-F FIELD]
                                  IKEPOLICY
```

Show information of a given IKEPolicy.

Positional arguments

IKEPOLICY ID or name of ikepolicy to look up

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

neutron vpn-ikepolicy-update command

```
usage: neutron vpn-ikepolicy-update [-h] [--request-format {json,xml}]
                                    [--lifetime units=UNITS,value=VALUE]
                                    IKEPOLICY
```

Update a given IKE Policy.

Positional arguments

IKEPOLICY ID or name of ikepolicy to update

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml}	The xml or json request format
–lifetime	units=UNITS,value=VALUE IKE Lifetime Attributes.'units'-seconds,default:seconds. 'value'-non negative integer, default:3600.

neutron vpn-ipsecpolicy-create command

```
usage: neutron vpn-ipsecpolicy-create [-h] [-f {shell,table}] [-c COLUMN]
                                      [--variable VARIABLE] [--prefix PREFIX]
                                      [--request-format {json,xml}]
                                      [--tenant-id TENANT_ID]
                                      [--description DESCRIPTION]
                                      [--transform-protocol {esp,ah,ah-esp}]
                                      [--auth-algorithm {sha1}]
                                      [--encryption-algorithm {3des,aes-128,
aes-192,aes-256}]
                                      [--encapsulation-mode {tunnel,
transport}]
                                      [--pfs {group2,group5,group14}]
                                      [--lifetime units=UNITS,value=VALUE]
                                      NAME
```

Create an ipsecpolicy.

Positional arguments

NAME Name of the IPsecPolicy

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–description DESCRIPTION	Description of the IPsecPolicy
–transform-protocol {esp,ah,ah-esp}	Transform Protocol in lowercase, default:esp
–auth-algorithm {sha1}	Authentication algorithm in lowercase, default:sha1
–encryption-algorithm {3des,aes-128,aes-192,aes-256}	Encryption Algorithm in lowercase, default:aes-128
–encapsulation-mode {tunnel,transport}	Encapsulation Mode in lowercase, default:tunnel
–pfs {group2,group5,group14}	Perfect Forward Secrecy in lowercase, default:group5
–lifetime	units=UNITS,value=VALUE IPsec Lifetime Attributes.'units'-seconds,default:seconds. 'value'-non negative integer, default:3600.

neutron vpn-ipsecpolicy-delete command

```
usage: neutron vpn-ipsecpolicy-delete [-h] [--request-format {json,xml}]
                                      IPSECPOLICY
```

Delete a given ipsecpolicy.

Positional arguments

IPSECPOLICY ID or name of ipsecpolicy to delete

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

neutron vpn-ipsecpolicy-list command

```
usage: neutron vpn-ipsecpolicy-list [-h] [-f {csv,table}] [-c COLUMN]
                                    [--quote {all,minimal,none,nonnumeric}]
                                    [--request-format {json,xml}] [-D]
                                    [-F FIELD] [-P SIZE] [--sort-key FIELD]
                                    [--sort-dir {asc,desc}]
```

List ipsecpolicies that belongs to a given tenant connection.

Optional arguments

-h, –help show this help message and exit

–request-format {json,xml} The xml or json request format

-D, –show-details Show detailed info

-F FIELD, –field FIELD Specify the field(s) to be returned by server, can be repeated

-P SIZE, –page-size SIZE Specify retrieve unit of each request, then split one request to several requests

–sort-key FIELD Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction

–sort-dir {asc,desc} Sort list in specified directions (This option can be repeated)

neutron vpn-ipsecpolicy-show command

```
usage: neutron vpn-ipsecpolicy-show [-h] [-f {shell,table}] [-c COLUMN]
```

```
                                    [--variable VARIABLE] [--prefix PREFIX]
                                    [--request-format {json,xml}] [-D]
                                    [-F FIELD]
                                    IPSECPOLICY
```

Show information of a given ipsecpolicy.

Positional arguments

IPSECPOLICY ID or name of ipsecpolicy to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron vpn-ipsecpolicy-update command

```
usage: neutron vpn-ipsecpolicy-update [-h] [--request-format {json,xml}]
                                      [--lifetime units=UNITS,value=VALUE]
                                      IPSECPOLICY
```

Update a given ipsec policy.

Positional arguments

IPSECPOLICY ID or name of ipsecpolicy to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–lifetime	units=UNITS,value=VALUE IPsec Lifetime Attributes.'units'-seconds,default:seconds. 'value'-non negative integer, default:3600.

neutron vpn-service-create command

```
usage: neutron vpn-service-create [-h] [-f {shell,table}] [-c COLUMN]
                                  [--variable VARIABLE] [--prefix PREFIX]
                                  [--request-format {json,xml}]
                                  [--tenant-id TENANT_ID] [--admin-state-down]
                                  [--name NAME] [--description DESCRIPTION]
                                  ROUTER SUBNET
```

Create a VPNService.

Positional arguments

ROUTER Router unique identifier for the vpnservice

SUBNET Subnet unique identifier for the vpnservice deployment

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
–tenant-id TENANT_ID	The owner tenant ID
–admin-state-down	Set admin state up to false
–name NAME	Set a name for the vpnservice
–description DESCRIPTION	Set a description for the vpnservice

neutron vpn-service-delete command

```
usage: neutron vpn-service-delete [-h] [--request-format {json,xml}]
                                  VPNSERVICE
```

Delete a given VPNService.

Positional arguments

VPNSERVICE ID or name of vpnservice to delete

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

neutron vpn-service-list command

```
usage: neutron vpn-service-list [-h] [-f {csv,table}] [-c COLUMN]
                                [--quote {all,minimal,none,nonnumeric}]
                                [--request-format {json,xml}] [-D] [-F FIELD]
                                [-P SIZE] [--sort-key FIELD]
                                [--sort-dir {asc,desc}]
```

List VPNService configurations that belong to a given tenant.

Optional arguments

-h, –help	show this help message and exit

–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated
-P SIZE, –page-size SIZE	Specify retrieve unit of each request, then split one request to several requests
–sort-key FIELD	Sort list by specified fields (This option can be repeated), The number of sort_dir and sort_key should match each other, more sort_dir specified will be omitted, less will be filled with asc as default direction
–sort-dir {asc,desc}	Sort list in specified directions (This option can be repeated)

neutron vpn-service-show command

```
usage: neutron vpn-service-show [-h] [-f {shell,table}] [-c COLUMN]
                                [--variable VARIABLE] [--prefix PREFIX]
                                [--request-format {json,xml}] [-D] [-F FIELD]
                                VPNSERVICE
```

Show information of a given VPNService.

Positional arguments

VPNSERVICE ID or name of vpnservice to look up

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format
-D, –show-details	Show detailed info
-F FIELD, –field FIELD	Specify the field(s) to be returned by server, can be repeated

neutron vpn-service-update command

```
usage: neutron vpn-service-update [-h] [--request-format {json,xml}]
                                  VPNSERVICE
```

Update a given VPNService.

Positional arguments

VPNSERVICE ID or name of vpnservice to update

Optional arguments

-h, –help	show this help message and exit
–request-format {json,xml}	The xml or json request format

7. neutron-debug command-line client

The **neutron-debug** client is an extension to the **neutron** command-line interface (CLI) for the OpenStack neutron-debug tool. This chapter documents **neutron-debug** version 2.3.0.

For help on a specific **neutron-debug** command, enter:

```
$ neutron-debug help COMMAND
```

neutron-debug usage

```
            [--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 <volume-api-ver>]
            [--os-cacert <ca-certificate>] [--retries <retries>]
            <subcommand> ...
```

Subcommands

probe-create Create probe port - create port and interface within a network namespace.

probe-list List all probes.

probe-clear Clear all probes.

probe-delete Delete probe - delete port then delete the namespace.

probe-exec Execute commands in the namespace of the probe.

ping-all ping-all is all-in-one command to ping all fixed IP's in a specified network.

neutron-debug optional arguments

–version Show version number and exit.

-v, –verbose, –debug Increase verbosity of output and show tracebacks on errors. Can be repeated.

-q, –quiet	Suppress output except warnings and errors
-h, –help	Show this help message and exit
–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 public URL.
–os-url <url>	Defaults to `env[OS_URL]`
–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 neutron-debug 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.
–config-file CONFIG_FILE	Config file for interface driver (You may also use l3_agent.ini)

neutron-debug probe-create command

```
usage: neutron-debug probe-create NET
```

Create probe port - create port and interface, then place it into the created network namespace.

Positional arguments

NET ID ID of the network in which the probe will be created.

neutron-debug probe-list command

```
usage: neutron-debug probe-list
```

```
```

List probes.

neutron-debug probe-clear command

```
usage: neutron-debug probe-clear
```

Clear all probes.

neutron-debug probe-delete command

```
usage: neutron-debug probe-delete <port-id>
```

Remove a probe.

Positional arguments

<port-id> ID of the probe to delete.

neutron-debug probe-exec command

```
usage: neutron-debug probe-exec <port-id> <command>
```

Execute commands in the namespace of the probe

neutron-debug ping-all command

```
usage: neutron-debug ping-all <port-id> --timeout <number
```

All-in-one command to ping all fixed IP's in a specified network.

Positional arguments

<port-id> ID of the port to use.

Optional arguments

–timeout <timeout in seconds> Optional ping timeout.

neutron-debug example

```
usage: neutron-debug create-probe < NET_ID>
```

Create a probe namespace within the network identified by NET_ID. The namespace will have the name of qprobe-<UUID of the probe port>

Note

For the following examples to function, the security group rules may need to be modified to allow the SSH (TCP port 22) or ping (ICMP) traffic into network.

```
usage: neutron-debug probe-exec <probe ID> "ssh <IP of instance>"
```

SSH to an instance within the network.

```
usage: neutron-debug ping-all <network ID>"
```

Ping all instances on this network to verify they are responding.

```
usage: neutron-debug probe-exec <probe_ID> dhcping <VM_MAC address> -s <IP of
 DHCP server>"
```

Ping the DHCP server for this network using dhcping to verify it is working.

8. Object Storage command-line client

The **swift** client is the command-line interface (CLI) for the OpenStack Object Storage API and its extensions. This chapter documents **swift** version 2.0.3.

For help on a specific **swift** command, enter:

```
$ swift help COMMAND
```

swift usage

```
            [--debug] [--info] [--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-id <auth-tenant-id>]
            [--os-tenant-name <auth-tenant-name>]
            [--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> ...
```

Subcommands

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; creates containers if not present
stat	Displays information for the account, container, or object
upload	Uploads files or directories to the given container
capabilities	List cluster capabilities

swift examples

```
swift -A https://auth.api.rackspacecloud.com/v1.0 -U user -K api_key stat -v
```

```
swift --os-auth-url https://api.example.com/v2.0 --os-tenant-name tenant \
    --os-username user --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
```

swift optional arguments

–version	show program's version number and exit
-h, –help	show this help message and exit
-s, –snet	Use SERVICENET internal network
-v, –verbose	Print more info
–debug	Show the curl commands and results of all http queries regardless of result status.
–info	Show the curl commands and results of all http queries which return an error.
-q, –quiet	Suppress status output
-A AUTH, –auth=AUTH URL	for obtaining an auth token
-V AUTH_VERSION, –auth-version=AUTH_VERSION	Specify a version for authentication. Defaults to 1.0.
-U USER, –user=USER	User name for obtaining an auth token.
-K KEY, –key=KEY	Key for obtaining an auth token.
-R RETRIES, –retries=RETRIES	The number of times to retry a failed connection.
–os-username=<auth-user-name>	OpenStack username. Defaults to `env[OS_USERNAME]`.
–os-password=<auth-password>	OpenStack password. Defaults to `env[OS_PASSWORD]`.
–os-tenant-id=<auth-tenant-id>	OpenStack tenant ID. Defaults to `env[OS_TENANT_ID]`
–os-tenant-name=<auth-tenant-name>	OpenStack tenant name. Defaults to `env[OS_TENANT_NAME]`.
–os-auth-url=<auth-url>	OpenStack auth URL. Defaults to `env[OS_AUTH_URL]`.
–os-auth-token=<auth-token>	OpenStack token. Defaults to `env[OS_AUTH_TOKEN]`. Used with –os-storage-url to bypass the usual username/password authentication.
–os-storage-url=<storage-url>	OpenStack storage URL. Defaults to `env[OS_STORAGE_URL]`. Overrides the storage url

	returned during auth. Will bypass authentication when used with –os-auth-token.
–os-region-name=<region-name>	OpenStack region name. Defaults to `env[OS_REGION_NAME]`
–os-service-type=<service-type>	OpenStack Service type. Defaults to `env[OS_SERVICE_TYPE]`
–os-endpoint-type=<endpoint-type>	OpenStack Endpoint type. Defaults to `env[OS_ENDPOINT_TYPE]`
–os-cacert=<ca-certificate>	Specify a CA bundle file to use in verifying a TLS (https) server certificate. Defaults to `env[OS_CACERT]`
–insecure	Allow swiftclient to access servers without having to verify the SSL certificate. Defaults to `env[SWIFTCLIENT_INSECURE]` (set to 'true' to enable).
–no-ssl-compression	This option is deprecated and not used anymore. SSL compression should be disabled by default by the system SSL library

swift delete command

```
Usage: Delete a container or objects within a container
```

Positional arguments

<container>	Name of container to delete from
[object]	Name of object to delete. Specify multiple times for multiple objects

Optional arguments

–all	Delete all containers and objects
–leave-segments	Do not delete segments of manifest objects
–object-threads <threads>	Number of threads to use for deleting objects. Default is 10
–container-threads <threads>	Number of threads to use for deleting containers. Default is 10

swift download command

```
Usage: Download objects from containers
```

Positional arguments

<container> Name of container to download from. To download a whole account, omit this and specify –all.

[object] Name of object to download. Specify multiple times for multiple objects. Omit this to download all objects from the container.

Optional arguments

–all Indicates that you really want to download everything in the account

–marker Marker to use when starting a container or account download

–prefix <prefix> Only download items beginning with <prefix>

–output <out_file> For a single file download, stream the output to <out_file>. Specifying "-" as <out_file> will redirect to stdout

–object-threads <threads> Number of threads to use for downloading objects. Default is 10

–container-threads <threads> Number of threads to use for downloading containers. Default is 10

–no-download Perform download(s), but don't actually write anything to disk

–header <header_name:header_value> Adds a customized request header to the query, like "Range" or "If-Match". This argument is repeatable. Example –header "content-type:text/plain"

–skip-identical Skip downloading files that are identical on both sides

swift list command

```
Usage: Lists the containers for the account or the objects for a container
```

Positional arguments

[container] Name of container to list object in

Optional arguments

–long Long listing format, similar to ls -l

–lh Report sizes in human readable format similar to ls -lh

–totals Used with -l or –lh, only report totals

–prefix Only list items beginning with the prefix

–delimiter Roll up items with the given delimiter. For containers only. See OpenStack Swift API documentation for what this means.

swift post command

```
Usage: Updates meta information for the account, container, or object.
If the container is not found, it will be created automatically.
```

Positional arguments

[container] Name of container to post to

[object] Name of object to post. Specify multiple times for multiple objects

Optional arguments

–read-acl <acl> Read ACL for containers. Quick summary of ACL syntax: .r:*, .r:-.example.com, .r:www.example.com, account1, account2:user2

–write-acl <acl> Write ACL for containers. Quick summary of ACL syntax: account1 account2:user2

–sync-to <sync-to> Sync To for containers, for multi-cluster replication

–sync-key <sync-key> Sync Key for containers, for multi-cluster replication

–meta <name:value> Sets a meta data item. This option may be repeated. Example: -m Color:Blue -m Size:Large

–header <header> Set request headers. This option may be repeated. Example -H "content-type:text/plain"

swift stat command

```
Usage: Displays information for the account, container, or object
```

Positional arguments

[container] Name of container to stat from

[object] Name of object to stat. Specify multiple times for multiple objects

Optional arguments

–lh Report sizes in human readable format similar to ls -lh

swift upload command

```
Usage: Uploads specified files and directories to the given container
```

Positional arguments

<container>	Name of container to upload to
<file_or_directory>	Name of file or directory to upload. Specify multiple times for multiple uploads

Optional arguments

–changed	Only upload files that have changed since the last upload
–skip-identical	Skip uploading files that are identical on both sides
–segment-size <size>	Upload files in segments no larger than <size> and then create a "manifest" file that will download all the segments as if it were the original file
–segment-container <container>	Upload the segments into the specified container. If not specified, the segments will be uploaded to a <container>_segments container so as to not pollute the main <container> listings.
–leave-segments	Indicates that you want the older segments of manifest objects left alone (in the case of overwrites)
–object-threads <threads>	Number of threads to use for uploading full objects. Default is 10.
–segment-threads <threads>	Number of threads to use for uploading object segments. Default is 10.
–header <header>	Set request headers with the syntax header:value. This option may be repeated. Example -H "content-type:text/plain".
–use-slo	When used in conjunction with –segment-size will create a Static Large Object instead of the default Dynamic Large Object.
–object-name <object-name>	Upload file and name object to <object-name> or upload dir and use <object-name> as object prefix instead of folder name

9. Orchestration command-line client

The **heat** client is the command-line interface (CLI) for the Orchestration API and its extensions. This chapter documents **heat** version 0.2.8.

For help on a specific **heat** command, enter:

```
$ heat help COMMAND
```

heat usage

```
usage: heat [--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] [--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] [--include-password]
            <subcommand> ...
```

Subcommands

action-resume Resume the stack.

action-suspend	Suspend the stack.
build-info	Retrieve build information.
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.
output-list	Show available outputs.
output-show	Show a specific stack output.
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.
resource-signal	Send a signal to a resource.
resource-template	Generate a template based on a resource.
resource-type-list	List the available resource types.
resource-type-show	Show the resource type.
stack-abandon	Abandon the stack.
stack-adopt	Adopt a stack.
stack-create	Create the stack.
stack-delete	Delete the stack(s).
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.
bash-completion	Prints all of the commands and options to stdout.
help	Display help about this program or one of its subcommands.

heat optional arguments

–version	Shows the client version and exits.
-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 keystone 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]`.
–include-password	Send os-username and os-password to heat.

heat action-resume command

```
usage: heat action-resume <NAME or ID>
```

Resume the stack.

Positional arguments

<NAME or ID> Name or ID of stack to resume.

heat action-suspend command

```
usage: heat action-suspend <NAME or ID>
```

Suspend the stack.

Positional arguments

<NAME or ID> Name or ID of stack to suspend.

heat build-info command

```
usage: heat build-info
```

Retrieve build information.

heat event-list command

```
usage: heat event-list [-r <RESOURCE>] <NAME or ID>
```

List events for a stack.

Positional arguments

<NAME or ID> Name or ID of stack to show the events for.

Optional arguments

-r <RESOURCE>, –resource <RESOURCE> Name of the resource to filter events by.

heat event-show command

```
usage: heat event-show <NAME or ID> <RESOURCE> <EVENT>
```

Describe the event.

Positional arguments

<NAME or ID> Name or ID of stack to show the events for.

<RESOURCE> Name of the resource the event belongs to.

<EVENT> ID of event to display details for.

heat output-list command

```
usage: heat output-list <NAME or ID>
```

Show available outputs.

Positional arguments

<NAME or ID> Name or ID of stack to query.

heat output-show command

```
usage: heat output-show <NAME or ID> <OUTPUT NAME>
```

Show a specific stack output.

Positional arguments

<NAME or ID> Name or ID of stack to query.

<OUTPUT NAME> Name of an output to display.

heat resource-list command

```
usage: heat resource-list <NAME or ID>
```

Show list of resources belonging to a stack.

Positional arguments

<NAME or ID> Name or ID of stack to show the resources for.

heat resource-metadata command

```
usage: heat resource-metadata <NAME or ID> <RESOURCE>
```

List resource metadata.

Positional arguments

<NAME or ID> Name or ID of stack to show the resource metadata for.

<RESOURCE> Name of the resource to show the metadata for.

heat resource-show command

```
usage: heat resource-show <NAME or ID> <RESOURCE>
```

Describe the resource.

Positional arguments

<NAME or ID> Name or ID of stack to show the resource for.

<RESOURCE> Name of the resource to show the details for.

heat resource-signal command

```
usage: heat resource-signal [-D <DATA>] [-f <FILE>] <NAME or ID> <RESOURCE>
```

Send a signal to a resource.

Positional arguments

<NAME or ID> Name or ID of stack the resource belongs to.

<RESOURCE> Name of the resource to signal.

Optional arguments

-D <DATA>, –data <DATA> JSON Data to send to the signal handler.

-f <FILE>, –data-file <FILE> File containing JSON data to send to the signal handler.

heat resource-template command

```
usage: heat resource-template [-F <FORMAT>] <RESOURCE>
```

Generate a template based on a resource.

Positional arguments

<RESOURCE> Name of the resource to generate a template for.

Optional arguments

-F <FORMAT>, –format <FORMAT> The template output format, one of: yaml, json.

heat resource-type-list command

```
usage: heat resource-type-list
```

List the available resource types.

heat resource-type-show command

```
usage: heat resource-type-show <RESOURCE_TYPE>
```

Show the resource type.

Positional arguments

<RESOURCE_TYPE> Resource type to get the details for.

heat stack-abandon command

```
usage: heat stack-abandon <NAME or ID>
```

Abandon the stack.

Positional arguments

<NAME or ID> Name or ID of stack to abandon.

heat stack-adopt command

```
usage: heat stack-adopt [-f <FILE>] [-e <FILE or URL>] [-u <URL>] [-o <URL>]
                        [-c <TIMEOUT>] [-a <FILE or URL>] [-r]
                        [-P <KEY1=VALUE1;KEY2=VALUE2...>]
                        <STACK_NAME>
```

Adopt a stack.

Positional arguments

<STACK_NAME> Name of the stack to adopt.

Optional arguments

-f <FILE>, –template-file <FILE>	Path to the template.
-e <FILE or URL>, –environment-file <FILE or URL>	Path to the environment.
-u <URL>, –template-url <URL>	URL of template.
-o <URL>, –template-object <URL>	URL to retrieve template object (e.g from swift).
-c <TIMEOUT>, –create-timeout <TIMEOUT>	Stack creation timeout in minutes. Default: 60.
-a <FILE or URL>, –adopt-file <FILE or URL>	Path to adopt stack data file.
-r, –enable-rollback	Enable rollback on create/update failure.
-P <KEY1=VALUE1;KEY2=VALUE2...>, –parameters <KEY1=VALUE1;KEY2=VALUE2...>	Parameter values used to create the stack. This can be specified multiple times, or once with parameters separated by a semicolon.

heat stack-create command

```
usage: heat stack-create [-f <FILE>] [-e <FILE or URL>] [-u <URL>] [-o <URL>]
                         [-c <TIMEOUT>] [-r] [-P <KEY1=VALUE1;KEY2=VALUE2...>]
                         <STACK_NAME>
```

Create the stack.

Positional arguments

<STACK_NAME> Name of the stack to create.

Optional arguments

-f <FILE>, –template-file <FILE>	Path to the template.
-e <FILE or URL>, –environment-file <FILE or URL>	Path to the environment.
-u <URL>, –template-url <URL>	URL of template.
-o <URL>, –template-object <URL>	URL to retrieve template object (e.g. from swift).
-c <TIMEOUT>, –create-timeout <TIMEOUT>	Stack creation timeout in minutes. Default: 60.
-r, –enable-rollback	Enable rollback on create/update failure.

-P <KEY1=VALUE1;KEY2=VALUE2...>, –parameters <KEY1=VALUE1;KEY2=VALUE2...> Parameter values used to create the stack. This can be specified multiple times, or once with parameters separated by a semicolon.

heat stack-delete command

```
usage: heat stack-delete <NAME or ID> [<NAME or ID> ...]
```

Delete the stack(s).

Positional arguments

<NAME or ID> Name or ID of stack(s) to delete.

heat stack-list command

```
usage: heat stack-list [-f <KEY1=VALUE1;KEY2=VALUE2...>] [-l <LIMIT>]
                       [-m <ID>]
```

List the user's stacks.

Optional arguments

-f <KEY1=VALUE1;KEY2=VALUE2...>, –filters <KEY1=VALUE1;KEY2=VALUE2...> Filter parameters to apply on returned stacks. This can be specified multiple times, or once with parameters separated by a semicolon.

-l <LIMIT>, –limit <LIMIT> Limit the number of stacks returned.

-m <ID>, –marker <ID> Only return stacks that appear after the given stack ID.

heat stack-show command

```
usage: heat stack-show <NAME or ID>
```

Describe the stack.

Positional arguments

<NAME or ID> Name or ID of stack to describe.

heat stack-update command

```
usage: heat stack-update [-f <FILE>] [-e <FILE or URL>] [-u <URL>] [-o <URL>]
                         [-P <KEY1=VALUE1;KEY2=VALUE2...>]
                         <NAME or ID>
```

Update the stack.

Positional arguments

<NAME or ID> Name or ID of stack to update.

Optional arguments

-f <FILE>, –template-file <FILE> Path to the template.

-e <FILE or URL>, –environment-file <FILE or URL> Path to the environment.

-u <URL>, –template-url <URL> URL of template.

-o <URL>, –template-object <URL> URL to retrieve template object (e.g. from swift).

-P <KEY1=VALUE1;KEY2=VALUE2...>, –parameters <KEY1=VALUE1;KEY2=VALUE2...> Parameter values used to create the stack. This can be specified multiple times, or once with parameters separated by a semicolon.

heat template-show command

```
usage: heat template-show <NAME or ID>
```

Get the template for the specified stack.

Positional arguments

<NAME or ID> Name or ID of stack to get the template for.

heat template-validate command

```
usage: heat template-validate [-u <URL>] [-f <FILE>] [-e <FILE or URL>]
                              [-o <URL>] [-P <KEY1=VALUE1;KEY2=VALUE2...>]
```

Validate a template with parameters.

Optional arguments

-u <URL>, –template-url <URL> URL of template.

-f <FILE>, –template-file <FILE> Path to the template.

-e <FILE or URL>, –environment-file <FILE or URL> Path to the environment.

-o <URL>, –template-object <URL> URL to retrieve template object (e.g. from swift).

-P <KEY1=VALUE1;KEY2=VALUE2...>, –parameters <KEY1=VALUE1;KEY2=VALUE2...>	Parameter values to validate. This can be specified multiple times, or once with parameters separated by a semicolon.

10. Telemetry command-line client

The **ceilometer** client is the command-line interface (CLI) for the Telemetry API and its extensions. This chapter documents **ceilometer** version 1.0.9.

For help on a specific **ceilometer** command, enter:

```
$ ceilometer help COMMAND
```

ceilometer usage

```
usage: ceilometer [--version] [-d] [-v] [-k] [--cert-file CERT_FILE]
                  [--key-file KEY_FILE] [--os-cacert <ca-certificate-file>]
                  [--ca-file OS_CACERT] [--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> ...
```

Subcommands

alarm-combination-create Create a new alarm based on state of other alarms.

alarm-combination-update	Update an existing alarm based on state of other alarms.
alarm-create	Create a new alarm (Deprecated).
alarm-delete	Delete an alarm.
alarm-history	Display the change history of an alarm.
alarm-list	List the user's alarms.
alarm-show	Show an alarm.
alarm-state-get	Get the state of an alarm.
alarm-state-set	Set the state of an alarm.
alarm-threshold-create	Create a new alarm based on computed statistics.
alarm-threshold-update	Update an existing alarm based on computed statistics.
alarm-update	Update an existing alarm.
event-list	List events.
event-show	Show a particular event.
event-type-list	List event types.
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 a meter.
statistics	List the statistics for a meter.
trait-description-list	List trait info for an event type.
trait-list	List trait all traits with name <trait_name> for Event Type
bash-completion	Prints all of the commands and options to stdout.
help	Display help about this program or one of its subcommands.

ceilometer optional arguments

–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.
–os-cacert <ca-certificate-file>	Path of CA TLS certificate(s) used to verifythe remote server's certificate. Without this option ceilometer 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
–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]`

ceilometer alarm-combination-create command

```
usage: ceilometer alarm-combination-create --name <NAME>
```

```
                                          [--project-id <PROJECT_ID>]
                                          [--user-id <USER_ID>]
                                          [--description <DESCRIPTION>]
                                          [--state <STATE>]
                                          [--enabled {True|False}]
                                          [--alarm-action <Webhook URL>]
                                          [--ok-action <Webhook URL>]
                                          [--insufficient-data-action
<Webhook URL>]
                                          --alarm_ids <ALARM IDS>
                                          [--operator <OPERATOR>]
                                          [--repeat-actions {True|False}]
```

Create a new alarm based on state of other alarms.

Optional arguments

–name <NAME> Name of the alarm (must be unique per tenant) Required.

–project-id <PROJECT_ID> Tenant to associate with alarm (only settable by admin users)

–user-id <USER_ID> User to associate with alarm (only settable by admin users)

–description <DESCRIPTION> Free text description of the alarm

–state <STATE> State of the alarm, one of: ['ok', 'alarm', 'insufficient_data']

–enabled {True|False} True if alarm evaluation/actioning is enabled

–alarm-action <Webhook URL> URL to invoke when state transitions to alarm. May be used multiple times. Defaults to None.

–ok-action <Webhook URL> URL to invoke when state transitions to OK. May be used multiple times. Defaults to None.

–insufficient-data-action <Webhook URL> URL to invoke when state transitions to insufficient_data. May be used multiple times. Defaults to None.

–alarm_ids <ALARM IDS> List of alarm id Required.

–operator <OPERATOR> Operator to compare with, one of: ['and', 'or']

–repeat-actions {True|False} True if actions should be repeatedly notified while alarm remains in target state Defaults to False.

ceilometer alarm-combination-update command

```
usage: ceilometer alarm-combination-update -a <ALARM_ID> [--name <NAME>]
                                           [--project-id <PROJECT_ID>]
```

```
                                        [--user-id <USER_ID>]
                                        [--description <DESCRIPTION>]
                                        [--state <STATE>]
                                        [--enabled {True|False}]
                                        [--alarm-action <Webhook URL>]
                                        [--ok-action <Webhook URL>]
                                        [--insufficient-data-action
<Webhook URL>]
                                        [--alarm_ids <ALARM IDS>]
                                        [--operator <OPERATOR>]
                                        [--repeat-actions {True|False}]
```

Update an existing alarm based on state of other alarms.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID>	ID of the alarm to update. Required.
–name <NAME>	Name of the alarm (must be unique per tenant)
–project-id <PROJECT_ID>	Tenant to associate with alarm (only settable by admin users)
–user-id <USER_ID>	User to associate with alarm (only settable by admin users)
–description <DESCRIPTION>	Free text description of the alarm
–state <STATE>	State of the alarm, one of: ['ok', 'alarm', 'insufficient_data']
–enabled {True|False}	True if alarm evaluation/actioning is enabled
–alarm-action <Webhook URL>	URL to invoke when state transitions to alarm. May be used multiple times. Defaults to None.
–ok-action <Webhook URL>	URL to invoke when state transitions to OK. May be used multiple times. Defaults to None.
–insufficient-data-action <Webhook URL>	URL to invoke when state transitions to insufficient_data. May be used multiple times. Defaults to None.
–alarm_ids <ALARM IDS>	List of alarm id
–operator <OPERATOR>	Operator to compare with, one of: ['and', 'or']
–repeat-actions {True|False}	True if actions should be repeatedly notified while alarm remains in target state

ceilometer alarm-delete command

```
usage: ceilometer alarm-delete -a <ALARM_ID>
```

```
```

Delete an alarm.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID>	ID of the alarm to delete. Required.

ceilometer alarm-history command

```
usage: ceilometer alarm-history -a <ALARM_ID> [-q <QUERY>]
```

Display the change history of an alarm.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID>	ID of the alarm for which history is shown. Required.
-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean

ceilometer alarm-list command

```
usage: ceilometer alarm-list [-q <QUERY>]
```

List the user's alarms.

Optional arguments

-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean

ceilometer alarm-show command

```
usage: ceilometer alarm-show -a <ALARM_ID>
```

Show an alarm.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID>	ID of the alarm to show. Required.

ceilometer alarm-state-get command

```
usage: ceilometer alarm-state-get -a <ALARM_ID>
```

Get the state of an alarm.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID> ID of the alarm state to show. Required.

ceilometer alarm-state-set command

```
usage: ceilometer alarm-state-set -a <ALARM_ID> --state <STATE>
```

Set the state of an alarm.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID> ID of the alarm state to set. Required.

–state <STATE> State of the alarm, one of: ['ok', 'alarm', 'insufficient_data'] Required.

ceilometer alarm-threshold-create command

```
usage: ceilometer alarm-threshold-create --name <NAME>
                                         [--project-id <PROJECT_ID>]
                                         [--user-id <USER_ID>]
                                         [--description <DESCRIPTION>]
                                         [--state <STATE>]
                                         [--enabled {True|False}]
                                         [--alarm-action <Webhook URL>]
                                         [--ok-action <Webhook URL>]
                                         [--insufficient-data-action <Webhook
 URL>]
                                         -m <METRIC> [--period <PERIOD>]
                                         [--evaluation-periods <COUNT>]
                                         [--statistic <STATISTIC>]
                                         [--comparison-operator <OPERATOR>]
                                         --threshold <THRESHOLD> [-q <QUERY>]
                                         [--repeat-actions {True|False}]
```

Create a new alarm based on computed statistics.

Optional arguments

–name <NAME> Name of the alarm (must be unique per tenant) Required.

–project-id <PROJECT_ID> Tenant to associate with alarm (only settable by admin users)

–user-id <USER_ID> User to associate with alarm (only settable by admin users)

–description <DESCRIPTION>	Free text description of the alarm
–state <STATE>	State of the alarm, one of: ['ok', 'alarm', 'insufficient_data']
–enabled {True|False}	True if alarm evaluation/actioning is enabled
–alarm-action <Webhook URL>	URL to invoke when state transitions to alarm. May be used multiple times. Defaults to None.
–ok-action <Webhook URL>	URL to invoke when state transitions to OK. May be used multiple times. Defaults to None.
–insufficient-data-action <Webhook URL>	URL to invoke when state transitions to insufficient_data. May be used multiple times. Defaults to None.
-m <METRIC>, –meter-name <METRIC>	Metric to evaluate against Required.
–period <PERIOD>	Length of each period (seconds) to evaluate over
–evaluation-periods <COUNT>	Number of periods to evaluate over
–statistic <STATISTIC>	Statistic to evaluate, one of: ['max', 'min', 'avg', 'sum', 'count']
–comparison-operator <OPERATOR>	Operator to compare with, one of: ['lt', 'le', 'eq', 'ne', 'ge', 'gt']
–threshold <THRESHOLD>	Threshold to evaluate against Required.
-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean
–repeat-actions {True|False}	True if actions should be repeatedly notified while alarm remains in target state Defaults to False.

ceilometer alarm-threshold-update command

```
usage: ceilometer alarm-threshold-update -a <ALARM_ID> [--name <NAME>]
                                         [--project-id <PROJECT_ID>]
                                         [--user-id <USER_ID>]
                                         [--description <DESCRIPTION>]
                                         [--state <STATE>]
                                         [--enabled {True|False}]
                                         [--alarm-action <Webhook URL>]
                                         [--ok-action <Webhook URL>]
                                         [--insufficient-data-action <Webhook
 URL>]
                                         [-m <METRIC>] [--period <PERIOD>]
                                         [--evaluation-periods <COUNT>]
                                         [--statistic <STATISTIC>]
                                         [--comparison-operator <OPERATOR>]
                                         [--threshold <THRESHOLD>]
                                         [-q <QUERY>]
```

```
                                        [--repeat-actions {True|False}]
```

Update an existing alarm based on computed statistics.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID>	ID of the alarm to update. Required.
–name <NAME>	Name of the alarm (must be unique per tenant)
–project-id <PROJECT_ID>	Tenant to associate with alarm (only settable by admin users)
–user-id <USER_ID>	User to associate with alarm (only settable by admin users)
–description <DESCRIPTION>	Free text description of the alarm
–state <STATE>	State of the alarm, one of: ['ok', 'alarm', 'insufficient_data']
–enabled {True|False}	True if alarm evaluation/actioning is enabled
–alarm-action <Webhook URL>	URL to invoke when state transitions to alarm. May be used multiple times. Defaults to None.
–ok-action <Webhook URL>	URL to invoke when state transitions to OK. May be used multiple times. Defaults to None.
–insufficient-data-action <Webhook URL>	URL to invoke when state transitions to insufficient_data. May be used multiple times. Defaults to None.
-m <METRIC>, –meter-name <METRIC>	Metric to evaluate against
–period <PERIOD>	Length of each period (seconds) to evaluate over
–evaluation-periods <COUNT>	Number of periods to evaluate over
–statistic <STATISTIC>	Statistic to evaluate, one of: ['max', 'min', 'avg', 'sum', 'count']
–comparison-operator <OPERATOR>	Operator to compare with, one of: ['lt', 'le', 'eq', 'ne', 'ge', 'gt']
–threshold <THRESHOLD>	Threshold to evaluate against
-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean
–repeat-actions {True|False}	True if actions should be repeatedly notified while alarm remains in target state

ceilometer alarm-update command

```
usage: ceilometer alarm-update -a <ALARM_ID> [--name <NAME>]
                               [--project-id <PROJECT_ID>]
                               [--user-id <USER_ID>]
                               [--description <DESCRIPTION>] [--state <STATE>]
                               [--enabled {True|False}]
                               [--alarm-action <Webhook URL>]
                               [--ok-action <Webhook URL>]
                               [--insufficient-data-action <Webhook URL>]
                               [--period <PERIOD>]
                               [--evaluation-periods <COUNT>] [-m <METRIC>]
                               [--statistic <STATISTIC>]
                               [--comparison-operator <OPERATOR>]
                               [--threshold <THRESHOLD>]
                               [--matching-metadata <Matching Metadata>]
                               [--repeat-actions {True|False}]
```

Update an existing alarm.

Optional arguments

-a <ALARM_ID>, –alarm_id <ALARM_ID>	ID of the alarm to update. Required.
–name <NAME>	Name of the alarm (must be unique per tenant)
–project-id <PROJECT_ID>	Tenant to associate with alarm (only settable by admin users)
–user-id <USER_ID>	User to associate with alarm (only settable by admin users)
–description <DESCRIPTION>	Free text description of the alarm
–state <STATE>	State of the alarm, one of: ['ok', 'alarm', 'insufficient_data']
–enabled {True|False}	True if alarm evaluation/actioning is enabled
–alarm-action <Webhook URL>	URL to invoke when state transitions to alarm. May be used multiple times. Defaults to None.
–ok-action <Webhook URL>	URL to invoke when state transitions to OK. May be used multiple times. Defaults to None.
–insufficient-data-action <Webhook URL>	URL to invoke when state transitions to insufficient_data. May be used multiple times. Defaults to None.
–period <PERIOD>	Length of each period (seconds) to evaluate over
–evaluation-periods <COUNT>	Number of periods to evaluate over

-m <METRIC>, –meter-name <METRIC>	Metric to evaluate against
–statistic <STATISTIC>	Statistic to evaluate, one of: ['max', 'min', 'avg', 'sum', 'count']
–comparison-operator <OPERATOR>	Operator to compare with, one of: ['lt', 'le', 'eq', 'ne', 'ge', 'gt']
–threshold <THRESHOLD>	Threshold to evaluate against
–matching-metadata <Matching Metadata>	A meter should match this resource metadata (key=value) additionally to the meter_name Defaults to None.
–repeat-actions {True|False}	True if actions should be repeatedly notified while alarm remains in target state

ceilometer event-list command

```
usage: ceilometer event-list [-q <QUERY>]
```

List events.

Optional arguments

-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, floator datetime.

ceilometer event-show command

```
usage: ceilometer event-show -m <message_id>
```

Show a particular event.

Optional arguments

-m <message_id>, –message_id <message_id>	The id of the event. Should be a UUID Required.

ceilometer event-type-list command

```
usage: ceilometer event-type-list
```

List event types.

ceilometer meter-list command

```
usage: ceilometer meter-list [-q <QUERY>]
```

List the user's meters.

Optional arguments

-q <QUERY>, –query <QUERY> key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean

ceilometer resource-list command

```
usage: ceilometer resource-list [-q <QUERY>]
```

List the resources.

Optional arguments

-q <QUERY>, –query <QUERY> key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean.

ceilometer resource-show command

```
usage: ceilometer resource-show -r <RESOURCE_ID>
```

Show the resource.

Optional arguments

-r <RESOURCE_ID>, –resource_id <RESOURCE_ID> ID of the resource to show. Required.

ceilometer sample-create command

```
usage: ceilometer sample-create [--project-id <PROJECT_ID>]
                                [--user-id <USER_ID>] -r <RESOURCE_ID> -m
                                <METER_NAME> --meter-type <METER_TYPE>
                                --meter-unit <METER_UNIT> --sample-volume
                                <SAMPLE_VOLUME>
                                [--resource-metadata <RESOURCE_METADATA>]
                                [--timestamp <TIMESTAMP>]
```

Create a sample.

Optional arguments

–project-id <PROJECT_ID> Tenant to associate with sample (only settable by admin users)

–user-id <USER_ID> User to associate with sample (only settable by admin users)

-r <RESOURCE_ID>, –resource-id <RESOURCE_ID> ID of the resource. Required.

-m <METER_NAME>, –meter-name <METER_NAME>	the meter name Required.
–meter-type <METER_TYPE>	the meter type Required.
–meter-unit <METER_UNIT>	the meter unit Required.
–sample-volume <SAMPLE_VOLUME>	The sample volume Required.
–resource-metadata <RESOURCE_METADATA>	resource metadata
–timestamp <TIMESTAMP>	the sample timestamp

ceilometer sample-list command

```
usage: ceilometer sample-list [-q <QUERY>] -m <NAME> [-l <NUMBER>]
```

List the samples for a meter.

Optional arguments

-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean
-m <NAME>, –meter <NAME>	Name of meter to show samples for. Required.
-l <NUMBER>, –limit <NUMBER>	Maximum number of samples to return.

ceilometer statistics command

```
usage: ceilometer statistics [-q <QUERY>] -m <NAME> [-p <PERIOD>] [-g <FIELD>]
```

List the statistics for a meter.

Optional arguments

-q <QUERY>, –query <QUERY>	key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean
-m <NAME>, –meter <NAME>	Name of meter to show samples for. Required.
-p <PERIOD>, –period <PERIOD>	Period in seconds over which to group samples.
-g <FIELD>, –groupby <FIELD>	Field for group aggregation.

ceilometer trait-description-list command

```
usage: ceilometer trait-description-list -e <EVENT_TYPE>
```

List trait info for an event type.

Optional arguments

-e <EVENT_TYPE>, –event_type <EVENT_TYPE>	Type of the event for which traits will be shown Required.

ceilometer trait-list command

```
usage: ceilometer trait-list -e <EVENT_TYPE> -t <TRAIT_NAME>
```

List trait all traits with name <trait_name> for Event Type <event_type>.

Optional arguments

-e <EVENT_TYPE>, –event_type <EVENT_TYPE>	Type of the event for which traits will listed Required.
-t <TRAIT_NAME>, –trait_name <TRAIT_NAME>	The name of the trait to list Required.

11. Database Service command-line client

The **trove** client is the command-line interface (CLI) for the Database API and its extensions. This chapter documents **trove** version 1.0.3.

For help on a specific **trove** command, enter:

```
$ trove help COMMAND
```

trove usage

```
usage: trove [--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>] [--bypass-url <bypass-url>]
            [--database-service-name <database-service-name>]
            [--endpoint-type <endpoint-type>]
            [--os-database-api-version <database-api-ver>]
            [--os-cacert <ca-certificate>] [--retries <retries>]
            <subcommand> ...
```

Subcommands

backup-create	Creates a backup.
backup-delete	Deletes a backup.
backup-list	List available backups.
backup-list-instance	List available backups for an instance.
backup-show	Show details of a backup.
create	Creates a new instance.
database-create	Creates a database on an instance.
database-delete	Deletes a database.
database-list	Lists available databases on an instance.
delete	Deletes an instance.
flavor-list	Lists available flavors.
flavor-show	Show details of a flavor.
limit-list	Lists the limits for a tenant.
list	List all the instances.
resize-flavor	Resizes the flavor of an instance.
resize-volume	Resizes the volume size of an instance.
restart	Restarts the instance.
root-enable	Enables root for a instance.
root-show	Gets root enabled status for a instance.
secgroup-add-rule	Creates a security group rule.
secgroup-delete-rule	Deletes a security group rule.
secgroup-list	Lists all security groups.
secgroup-show	Shows details about a security group.
show	Show details of an instance.
user-create	Creates a user.

user-delete	Deletes a user from the instance.
user-grant-access	Grants access to a database(s) for a user.
user-list	Lists the users for a instance.
user-revoke-access	Revokes access to a database for a user.
user-show	Gets a user from the instance.
user-show-access	Gets a users access from the instance.
user-update-attributes	Updates a users attributes from the instance.
bash-completion	Print arguments for bash_completion.
help	Display help about this program or one of its subcommands.

trove 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 database for most actions
–service-name <service-name>	Defaults to `env[TROVE_SERVICE_NAME]`
–bypass-url <bypass-url>	Defaults to `env[TROVE_BYPASS_URL]`
–database-service-name <database-service-name>	Defaults to `env[TROVE_DATABASE_SERVICE_NAME]`
–endpoint-type <endpoint-type>	Defaults to `env[TROVE_ENDPOINT_TYPE]` or publicURL.
–os-database-api-version <database-api-ver>	Accepts 1,defaults to `env[OS_DATABASE_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.

trove backup-create command

```
usage: trove backup-create [--description <description>] <name> <instance>
```

Creates a backup.

Positional arguments

<name> Name of the backup.

<instance> UUID of the instance.

Optional arguments

–description <description> An optional description for the backup.

trove backup-delete command

```
usage: trove backup-delete <backup>
```

Deletes a backup.

Positional arguments

<backup> ID of the backup.

trove backup-list command

```
usage: trove backup-list
```

List available backups.

trove backup-list-instance command

```
usage: trove backup-list-instance <instance>
```

List available backups for an instance.

Positional arguments

<instance> ID of the instance.

trove backup-show command

```
usage: trove backup-show <backup>
```

```
```

Show details of a backup.

Positional arguments

<backup> ID of the backup.

trove create command

```
usage: trove create [--size <size>]
                    [--databases <databases> [<databases> ...]]
                    [--users <users> [<users> ...]] [--backup <backup>]
                    [--availability_zone <availability_zone>]
                    <name> <flavor_id>
```

Creates a new instance.

Positional arguments

<name> Name of the instance

<flavor_id> Flavor of the instance

Optional arguments

–size <size> Size of the instance disk in GB

–databases <databases> [<databases> ...] Optional list of databases.

–users <users> [<users> ...] Optional list of users in the form user:password.

–backup <backup> A backup UUID

–availability_zone <availability_zone> The Zone hint to give to nova

trove database-create command

```
usage: trove database-create [--character_set <character_set>]
                             [--collate <collate>]
                             <instance> <name>
```

Creates a database on an instance.

Positional arguments

<instance> UUID of the instance.

<name> Name of the backup.

Optional arguments

–character_set <character_set> Optional character set for database

–collate <collate> Optional collation type for database

trove database-delete command

```
usage: trove database-delete <instance> <database>
```

Deletes a database.

Positional arguments

<instance> UUID of the instance.

<database> Name of the database.

trove database-list command

```
usage: trove database-list <instance>
```

Lists available databases on an instance.

Positional arguments

<instance> UUID of the instance.

trove delete command

```
usage: trove delete <instance>
```

Deletes an instance.

Positional arguments

<instance> ID of the instance.

trove flavor-list command

```
usage: trove flavor-list
```

Lists available flavors.

trove flavor-show command

```
usage: trove flavor-show <flavor>
```

```
```

Show details of a flavor.

Positional arguments

<flavor> ID of the flavor.

trove limit-list command

```
usage: trove limit-list
```

Lists the limits for a tenant.

trove list command

```
usage: trove list
```

List all the instances.

trove resize-flavor command

```
usage: trove resize-flavor <instance> <flavor_id>
```

Resizes the flavor of an instance.

Positional arguments

<instance> UUID of the instance

<flavor_id> Flavor of the instance

trove resize-volume command

```
usage: trove resize-volume <instance> <size>
```

Resizes the volume size of an instance.

Positional arguments

<instance> UUID of the instance

<size> Size of the instance disk in GB

trove restart command

```
usage: trove restart <instance>
```

```
```

Restarts the instance.

Positional arguments

<instance> UUID of the instance

trove root-enable command

```
usage: trove root-enable <instance>
```

Enables root for a instance.

Positional arguments

<instance> UUID of the instance.

trove root-show command

```
usage: trove root-show <instance>
```

Gets root enabled status for a instance.

Positional arguments

<instance> UUID of the instance.

trove secgroup-add-rule command

```
usage: trove secgroup-add-rule <security_group> <protocol> <from_port>
                               <to_port> <cidr>
```

Creates a security group rule.

Positional arguments

<security_group> Security group name

<protocol> Protocol

<from_port> from port

<to_port> to port

<cidr> CIDR address

trove secgroup-delete-rule command

```
usage: trove secgroup-delete-rule <security_group_rule>
```

```
```

Deletes a security group rule.

Positional arguments

<security_group_rule> Security group rule

trove secgroup-list command

```
usage: trove secgroup-list
```

Lists all security groups.

trove secgroup-show command

```
usage: trove secgroup-show <security_group>
```

Shows details about a security group.

Positional arguments

<security_group> ID of the security group.

trove show command

```
usage: trove show <instance>
```

Show details of an instance.

Positional arguments

<instance> ID of the instance.

trove user-create command

```
usage: trove user-create [--host <host>]
                         [--databases <databases> [<databases> ...]]
                         <instance> <name> <password>
```

Creates a user.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

<password> Password of user

Optional arguments

–host <host> Optional host of user

–databases <databases> [<databases> ...] Optional list of databases.

trove user-delete command

```
usage: trove user-delete [--host <host>] <instance> <name>
```

Deletes a user from the instance.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

Optional arguments

–host <host> Optional host of user

trove user-grant-access command

```
usage: trove user-grant-access [--host <host>]
                               <instance> <name> <databases> [<databases> ...]
```

Grants access to a database(s) for a user.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

<databases> List of databases.

Optional arguments

–host <host> Optional host of user

trove user-list command

```
usage: trove user-list <instance>
```

Lists the users for a instance.

Positional arguments

<instance> UUID of the instance.

trove user-revoke-access command

```
usage: trove user-revoke-access [--host <host>] <instance> <name> <database>
```

Revokes access to a database for a user.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

<database> A single database.

Optional arguments

–host <host> Optional host of user

trove user-show command

```
usage: trove user-show [--host <host>] <instance> <name>
```

Gets a user from the instance.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

Optional arguments

–host <host> Optional host of user

trove user-show-access command

```
usage: trove user-show-access [--host <host>] <instance> <name>
```

Gets a users access from the instance.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

Optional arguments

–host <host> Optional host of user

trove user-update-attributes command

```
usage: trove user-update-attributes [--host <host>] [--new_name <new_name>]
                                    [--new_password <new_password>]
                                    [--new_host <new_host>]
                                    <instance> <name>
```

Updates a users attributes from the instance.

Positional arguments

<instance> UUID of the instance.

<name> Name of user

Optional arguments

–host <host> Optional host of user

–new_name <new_name> Optional new name of user

–new_password <new_password> Optional new password of user

–new_host <new_host> Optional new host of user

Appendix A. Community support

Table of Contents

Many resources are available to help you run and use OpenStack. Members of the OpenStack community 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 associated 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*
- *Command-Line Interface Reference*

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*

The *Training Guides* offer software training for cloud administration and management.

ask.openstack.org

During the set up or testing of OpenStack, you might have questions about how a specific task is completed 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 has already been 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, links to screen shots, and so on.

OpenStack mailing lists

A great way to get answers and insights is to post your question or problematic scenario to the OpenStack mailing list. You can learn from and help others who might have similar issues. 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 a broad range of topics but some of the information can be difficult to find or is a few pages deep. 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 relevant material. 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

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 has already been 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, links to screen shots, and so on.
- Be sure to include the software and package versions 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 Block Storage (cinder)
- Bugs: OpenStack Compute (nova)
- Bugs: OpenStack Dashboard (horizon)
- Bugs: OpenStack Identity (keystone)
- Bugs: OpenStack Image Service (glance)
- Bugs: OpenStack Networking (neutron)
- Bugs: OpenStack Object Storage (swift)
- Bugs: Bare Metal (ironic)
- Bugs: Data Processing Service (sahara)
- Bugs: Database Service (trove)
- Bugs: Orchestration (heat)
- Bugs: Telemetry (ceilometer)
- Bugs: Queue Service (marconi)

- Bugs: OpenStack API Documentation (api.openstack.org)
- Bugs: OpenStack Documentation (docs.openstack.org)

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